How to overcome/avoid High Frequency Effects on Debug Interfaces – Trace Port Design Guidelines

An On-Chip Debugger/Analyzer (OCD) like iSYSTEM’s iC5000 (Figure 1) acts as a link to the target hardware by standard debug interfaces such as JTAG, OnCE, DAP, NEXUS, ETM, etc. Some of these interfaces - especially the latter two - offer so called trace ports through which information of an embedded application can be retrieved.

Depending on the amount and quality of these on-chip trace data the OCD offers debugging functionality for software development as well as extensive software tests. One example is tracing and analyzing the execution of the running software. Program states can be inspected and program execution can be halted based on complex trigger conditions. In addition the OCD also supports performance optimization by function as well as data profiling and code coverage analysis.

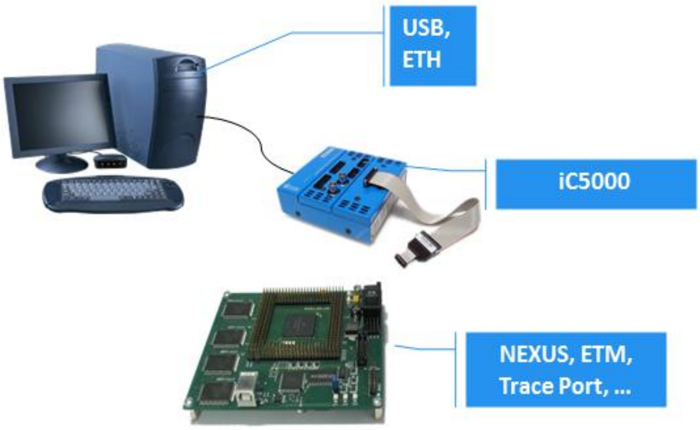


Figure 1: iC5000 connection to embedded hardware

Due to the continuously increasing demands of today’s embedded applications, MCUs get more and more complex by higher integration, performance acceleration and power saving requirements. Upcoming multi-core processors and new debug interfaces impose even bigger challenges in terms of more bandwidth at higher frequencies on trace ports.
However physically all trace ports consist of several data lines and clock. With higher integration and frequencies the on-chip trace signals are subject to (increased) signal distortion. And high signal distortion can prevent a trace tool (e.g. OCD) from capturing the trace stream correctly.

The quality of on-chip trace signals depends mostly on the way how the debug interface is realized by the semi-conductor vendor but also on the PCB design itself.

In this document we will discuss design considerations for the Printed Circuit Board (PCB) itself and will give some guidelines on how to design PCBs to minimize high frequency effects.

High Frequency Trace Signals

Before we talk about provisions to avoid signal distortion let’s talk about high frequencies trace signals.

On-chip debug functionality on most of today’s controllers is realized by compressing the CPU activity on-chip and delivering it to a trace port to enable an external tool (e.g. OCD) to record, analyze and display this information. The trace port consists of data lines and clock line which is used to capture the data in a moment, when it is valid and stable.
The data can be captured on rising edge of the clock (“Full Rate”) or on both clock edges (“Half Rate”). At lower frequencies and good signal integrity we can consider the clock and data lines as pure digital signals, which are correctly phase aligned. The OCD can capture the data accurately in this scenario.

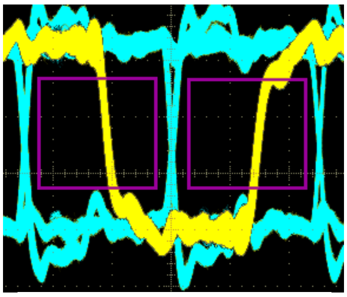


Figure 2: Good signal integrity at low frequency with large "Data Eyes"

Beside figures show the data and clock signals measured with an oscilloscope.

The *data* signal is shown in *blue*, the *clock* signal is *yellow*. The different blue lines show one signal at different time periods - the oscilloscope ran in “persistence” mode, so the curves drawn on the oscilloscope’s screen are accumulated. The dark area within the signal is called the *data eye* – sampling within this region yields a reliable trace recording (Figure 2: *pink* squares).

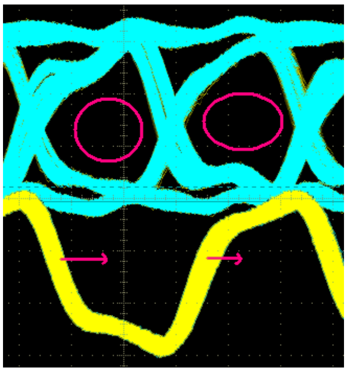


Figure 3: Higher frequency: Valid "Data Eyes require clock shift (pink arrows)

However when the frequency rises, the capturing of data becomes challenging due to increased signal integrity issues (noise, skew, crosstalk, reflections, ground bounce…). As you see in Figure 3 with higher frequencies the data eye shrinks (pink circle) and the clock has to get shifted (pink arrows).

Examples of High Frequency Effects

The following two examples show how the length of trace lines is reflected in signal integrity and consequently in functionality.
A typical evaluation board was used (figure 4 and 5), where the CPU sits on the upper piggyback board, which is optionally fitted to a lower, larger measurement board using high quality inter-board connectors. The debug interface is located on the CPU board.

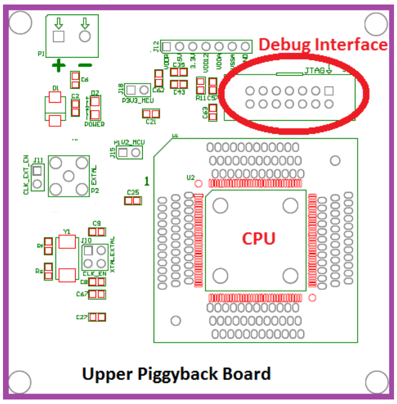


Figure 4: Upper Piggyback Board with CPU and Debug Interface

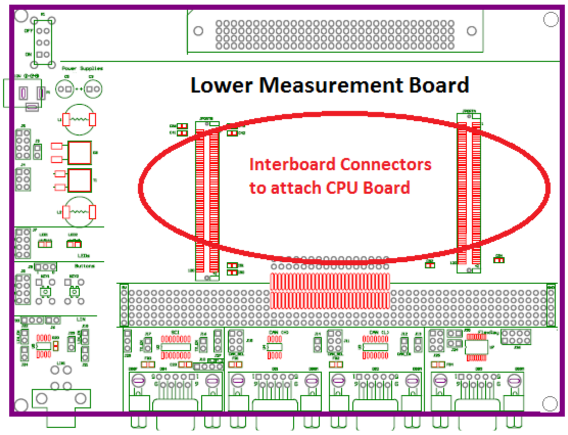


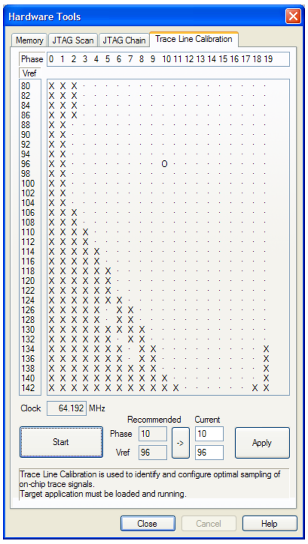
Figure 5: Lower Measurement Board with Interboard Connectors

The ***Line Calibration Result*** (figure 6 and 8) shows a view at the signal from the OCD perspective. The dotted sampling points indicate a stable signal, and the X positions indicate a changing (unstable) signal. A more detailed description how the signal is sampled and interpreted by the OCD you find in the chapter “Auto calibration”.

The ***Oscilloscope*** (figure 7 and 9) shows the same signal measured by an oscilloscope. The *yellow* line is the trace *clock* and the *blue* lines are the trace *data*, the dark area within the signal is the *data eye*. The oscilloscope ran again in “persistence” mode.

Trace lines with short stubs

This trace was made directly on the CPU board where the debug port is located. The CPU board is *not* attached to the measurement board this time. Due to short stubs, the size of the data eye is large and the tool has no problems in recording the trace.



Phase	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Vref																				
80	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
82	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
84	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
86	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
88	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
90	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
92	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
94	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
96	X	X	·	·	·	·	·	·	·	·	O	·	·	·	·	·	·	·	·	·
98	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
100	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
102	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
104	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
106	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
108	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
110	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
112	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
114	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
116	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
118	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·
120	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·
122	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·	·
124	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·	·
126	X	X	X	X	X	·	X	X	·	·	·	·	·	·	·	·	·	·	·	·
128	X	X	X	X	X	·	X	X	·	·	·	·	·	·	·	·	·	·	·	·
130	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·
132	X	X	X	X	X	X	·	X	X	·	·	·	·	·	·	·	·	·	·	·
134	X	X	X	X	X	X	X	·	X	X	·	·	·	·	·	·	·	·	·	X
136	X	X	X	X	X	X	X	·	X	X	·	·	·	·	·	·	·	·	·	X
138	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	X
140	X	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	X
142	X	X	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	X	X

Clock 64.192 MHz

	Recommended		Current
Phase	10	->	10
Vref	96		96

Trace Line Calibration is used to identify and configure optimal sampling of on-chip trace signals.
Target application must be loaded and running.

Figure 6: Trace Line Calibration Result

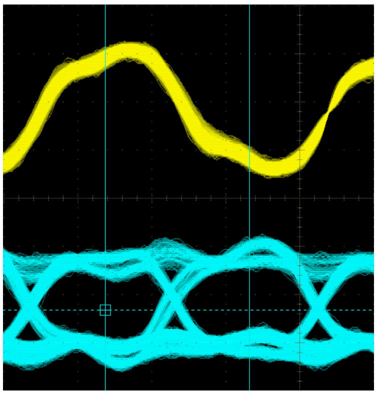
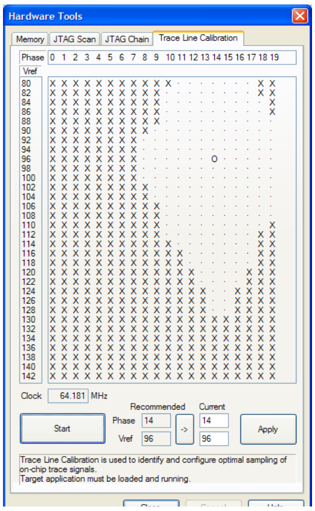


Figure 7: Measured by Oscilloscope

Trace lines with longer stubs (over connector to other board)

This trace was also made on the CPU board (because there is the debug port located) but now the CPU board is fitted (piggyback) to the lower measurement board. All trace lines are linked through the inter-board connector to the lower board. By passing through the connector, the stubs on the trace lines get much longer and due to that the data eye shrinks. Auto calibration can still find a safe sampling position, but any additional obstacle (CPU socket, longer PCB lines…) could shrink the data eye to a point where no safe sampling point could be established.



Phase	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Vref																				
80	X	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	X	X
82	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	X	X
84	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	X
86	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	X
88	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·
90	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·
92	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·
94	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·
96	X	X	X	X	X	X	X	X	·	·	·	·	·	·	O	·	·	·	·	·
98	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·
100	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·	·
102	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·
104	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·	·
106	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·
108	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	·
110	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	·	X
112	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	·	X	X
114	X	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	·	X	X
116	X	X	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	X	X
118	X	X	X	X	X	X	X	X	X	X	X	X	·	·	·	·	·	·	X	X
120	X	X	X	X	X	X	X	X	X	X	X	X	X	·	·	·	·	X	X	X
122	X	X	X	X	X	X	X	X	X	X	X	X	X	·	·	·	·	X	X	X
124	X	X	X	X	X	X	X	X	X	X	X	X	X	X	·	·	X	X	X	X
126	X	X	X	X	X	X	X	X	X	X	X	X	X	X	·	·	X	X	X	X
128	X	X	X	X	X	X	X	X	X	X	X	X	X	X	·	·	X	X	X	X
130	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
132	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
134	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
136	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
138	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
140	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
142	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Clock 64.181 MHz

	Recommended		Current
Phase	14	->	14
Vref	96		96

Trace Line Calibration is used to identify and configure optimal sampling of on-chip trace signals.
Target application must be loaded and running.

Figure 8: Trace Line Calibration Result

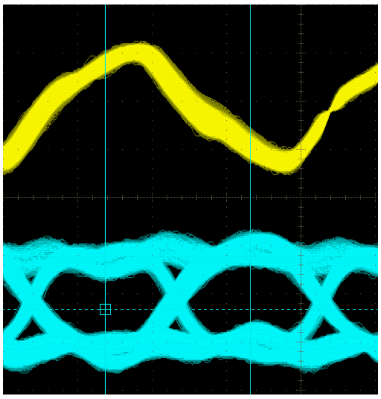


Figure 9: Measured by Oscilloscope

Recommended Measures

To avoid signal distortion and to retrieve a high quality trace already the PCB designer should consider several aspects to provide a stable environment.

- The CPU should be soldered on the PCB. Any socket can reduce the signal integrity dramatically.
- All trace port lines on the PCB should be as short as possible (max ~2,5 cm),
- Trace lines should run on the same layer or on layers with the same impedance.
- Preferred layer impedance is 50 Ohm.
- Mictor ground pins should be connected directly to PCB's GND plane.
- Trace clock should be serially terminated by 47 Ohm resistor as close as possible to the driver. The value of the resistor may be changed depending on driver characteristics.
- Trace clock should be clean of crosstalk – if possible with double distance to closest nets.
- Trace clock should have only point-to-point connection – any stubs should be avoided.
- It is strongly recommended also for other (data) lines to be point-to-point only. If any stubs are needed, they should be as short as possible. If longer are required, there should be a possibility to optionally disconnect them (e.g. by jumpers).
- Trace port data bus inner crosstalk is not so important, but it is critical to isolate the whole bus from other signals (including from the trace port clock).

Auto Calibration

Auto calibration is a feature of iSYSTEM's powerful on-chip debuggers to reduce noise, skew, crosstalk, etc. and to ensure a high quality trace data. Auto calibration allows the on-chip debugger to adjust the sampling position by shifting the sampling voltage and sampling delay in reference to the clock. This allows the tool to fine tune the instant when the signal is sampled to a moment where the signal is more stable. Auto calibration is implemented for the iC5000 and the iTRACE Probe2 (see Figure 10).

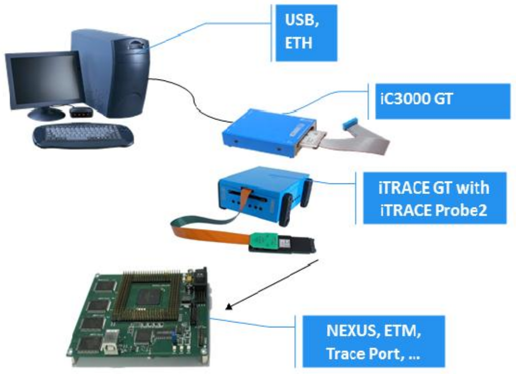


Figure 10: iC3000 with iTRACE GT and iTRACEProbe2

Auto calibration is realized by different means.

The iC5000 can shift the capturing threshold voltage and delay the clock line. On all channels and for all reference voltages the signal is sampled twice and checked when the signal is stable.
To achieve that, any input signal is routed parallel into two registers that use different clocks. The clock delay is programmable, but it is ~1ns apart. The voltage level based on Vref is also

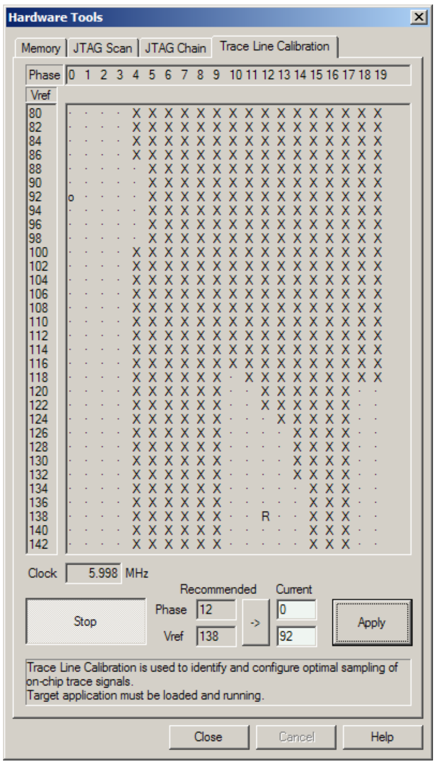
shifted. It can be adjusted in both directions to allow sampling at several different threshold voltages. Then the (two) signals are sampled using different threshold values.
If the signal is not changing in the 1ns period, the signal is considered stable and “sampled good”. If the signal changes then it is “sampled unpredictable”.

The sampling results are shown in winIDEA’s ***Trace Line Calibration*** results view (see Figures 6, 8). The graph is based on Vref and Phase which is the clock delay. If the signal is *stable* it shows a “**.**”. If the signal is *traveling from low to high* (or opposite), an “**X**” is shown. Although Figure 11 shows a trace with much noise, you can establish from the graph “data eye” areas, where it is save to sample.

Usually the auto calibration logic in the Blue Box decides automatically on “***R****ecommended*” Vref and Phase values (shown as “**R**”). But in cases of huge noise (Figure 11) this might not be possible and the user has to set these values manually in “*Current*” input boxes (shown as “**o**” in the graph).

On the iTRACE Probe 2 every signal line can be shifted and delayed individually (important for PCB designs where the onboard debug lines have different lengths and therefore different shifts). Additionally differential signaling is used that transports every signal over two lines where one line runs the inverted signal in the opposite direction. This way any signal distortion should be very limited across both lines.

What has Auto calibration to do with the PCB design? Auto calibration helps to receive high quality trace data because flaws in the PCB design are mitigated and even overcome. An on-chip debugger using Auto calibration can achieve much higher data transmission rates over the debug interface than tools without it.



Phase	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Vref																				
80	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
82	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
84	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
86	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
88	.	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
90	.	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
92	o	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
94	.	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
96	.	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
98	.	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
100	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
102	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
104	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
106	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
108	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
110	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
112	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
114	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
116	.	.	.	.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
118	.	.	.	.	X	X	X	X	X	X	.	X	X	X	X	X	X	X	X	X
120	.	.	.	.	X	X	X	X	X	X	.	.	X	X	X	X	X	X	.	.
122	.	.	.	.	X	X	X	X	X	X	.	.	X	X	X	X	X	X	.	.
124	.	.	.	.	X	X	X	X	X	X	.	.	.	X	X	X	X	X	.	.
126	.	.	.	.	X	X	X	X	X	X	.	.	.	.	X	X	X	X	.	.
128	.	.	.	.	X	X	X	X	X	X	.	.	.	.	X	X	X	X	.	.
130	.	.	.	.	X	X	X	X	X	X	.	.	.	.	X	X	X	X	.	.
132	.	.	.	.	X	X	X	X	X	X	.	.	.	.	X	X	X	X	.	.
134	.	.	.	.	X	X	X	X	X	X	.	.	.	.	.	X	X	X	.	.
136	.	.	.	.	X	X	X	X	X	X	.	.	.	.	.	X	X	X	.	.
138	.	.	.	.	X	X	X	X	X	X	.	.	R	.	.	X	X	X	.	.
140	.	.	.	.	X	X	X	X	X	X	.	.	.	.	.	X	X	X	.	.
142	.	.	.	.	X	X	X	X	X	X	.	.	.	.	.	X	X	X	.	.

Clock 5.998 MHz

	Recommended		Current
Phase	12	->	0
Vref	138		92

Trace Line Calibration is used to identify and configure optimal sampling of on-chip trace signals.
Target application must be loaded and running.

Figure 11: winIDEA's Trace Line Calibration view- much noise on line

With Auto calibration the iC5000 can capture up to **2Gbps data over a 16bit port**. With the iTRACE Probe 2 a data bandwidth of **5600Mbps can be achieved on a 16bit port** and using 4 iTRACE Probes 2 up to 20000Mbps (=**20Gbps**!) data bandwidth are possible.

iSYSTEM’s Reference Targets

To provide a stable environment iSYSTEM will offer so called reference targets. These are simple boards with little peripheral devices that are verified to provide a high quality trace.

The big advantages for customers are the potential

- to start prototyping when the final target hardware is not yet available
- to assess application’s behavior on a certified target
- to evaluate and audit iSYSTEM products, e.g. winIDEA, testIDEA in a stable environment
- to prove iSYSTEM’s product quality within a company’s internal qualification process

For more information on iSYSTEM Reference Targets contact info@isystem.com.

Links

The *“Trace Port PCB Design Guidelines”* document can be downloaded here.

Background information on the *“Possibilities and Limits of Software Debugging Techniques Depending on the Microcontroller/Microprocessor“*can be found here.

Explanation of Differential signaling .