SCI Hardware Support Tools

B. Skaali
Department of Physics, University of Oslo, P.O.Box 1048 Blindern, 0316 Oslo, Norway
and
SCILAB Technology AS, Oslo Research Park, Gaustadalléen 21, 0349 Oslo, Norway

→ Monitoring / measuring SCI traffic - why and how

→ SCI instrumentation, why did it take so long time to materialize?

→ SCIEurope WP2 - *Tools and Instruments Development*
Trinity College, Dublin

→ The SCITRAC and SCIview tracers
SCILAB Technology AS, Oslo

Instrumentation and hardware support tools for SCI

Tracing / diagnostics tasks:

- Signal verification at the link hardware level:
  - Link Controller
  - Card circuitry
  - Link cables and connectors (ever recurring problem)
- Protocol verification
- Catching rare “corner” cases: chip bugs, initialization sequence
  - Sophisticated trigger setup
  - Sifting through very deep traces recorded and stored in memory / on files
- Matching trace captures with software performance and optimization
  - Processor dependent parameters: PCI chipset, caches, etc.
  - Multiprocessor systems

Where to trace?

- **Pure software monitoring, for instance at the driver level**
  - ↑ No hardware required, basically straightforward approach
  - ↓ Congestion on links can only be inferred from low performance
- **On the PCI / switch card (B-link level)**
  - ↑ Relative easy access of B-link on Dolphin cards
  - ↑ A special monitoring chip can be fully integrated on the card
  - ↓ Does not show retry traffic on the link
- **On the link**
  - ↑ One measures the real signals, both electrically and protocolwise
  - ↓ May perturb the signal quality if the impedance matching is not perfect, which becomes a critical issue at high frequencies
- **For the SCIEurope WP2 - Trinity College, Dublin - a system for (off-line) deep traces both at B-link and link level is being developed.**

Going back to the dawn of SCI....

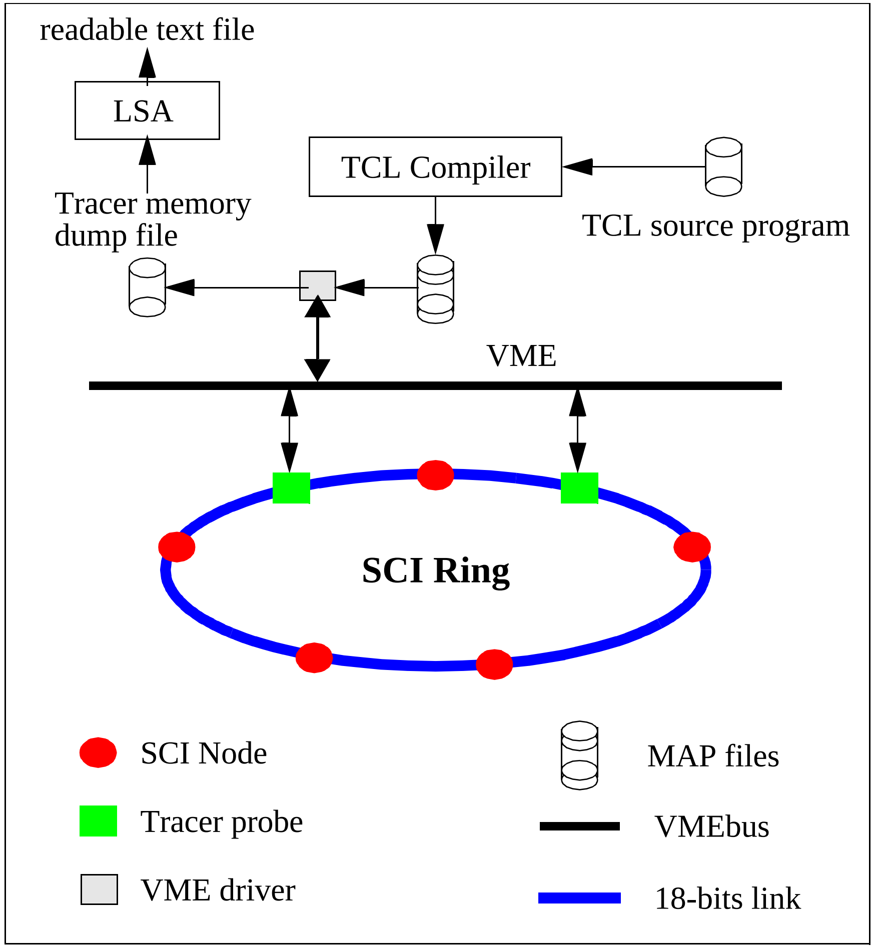
The Dolphin Tracer concept - anno 1993 (I. Birkeli, Dolphin)

System components:

- **The SCI Tracer module in VMEbus implementation -** Dolphin
- **Tracing conditions symbolically defined in the Tracer Control Language (TCL) -** (Master thesis Bin Wu, Univ. of Oslo)
- **Analysis of the collected information by the SCI Link Symbolic Analyzer (LSA) software package -** Dolphin

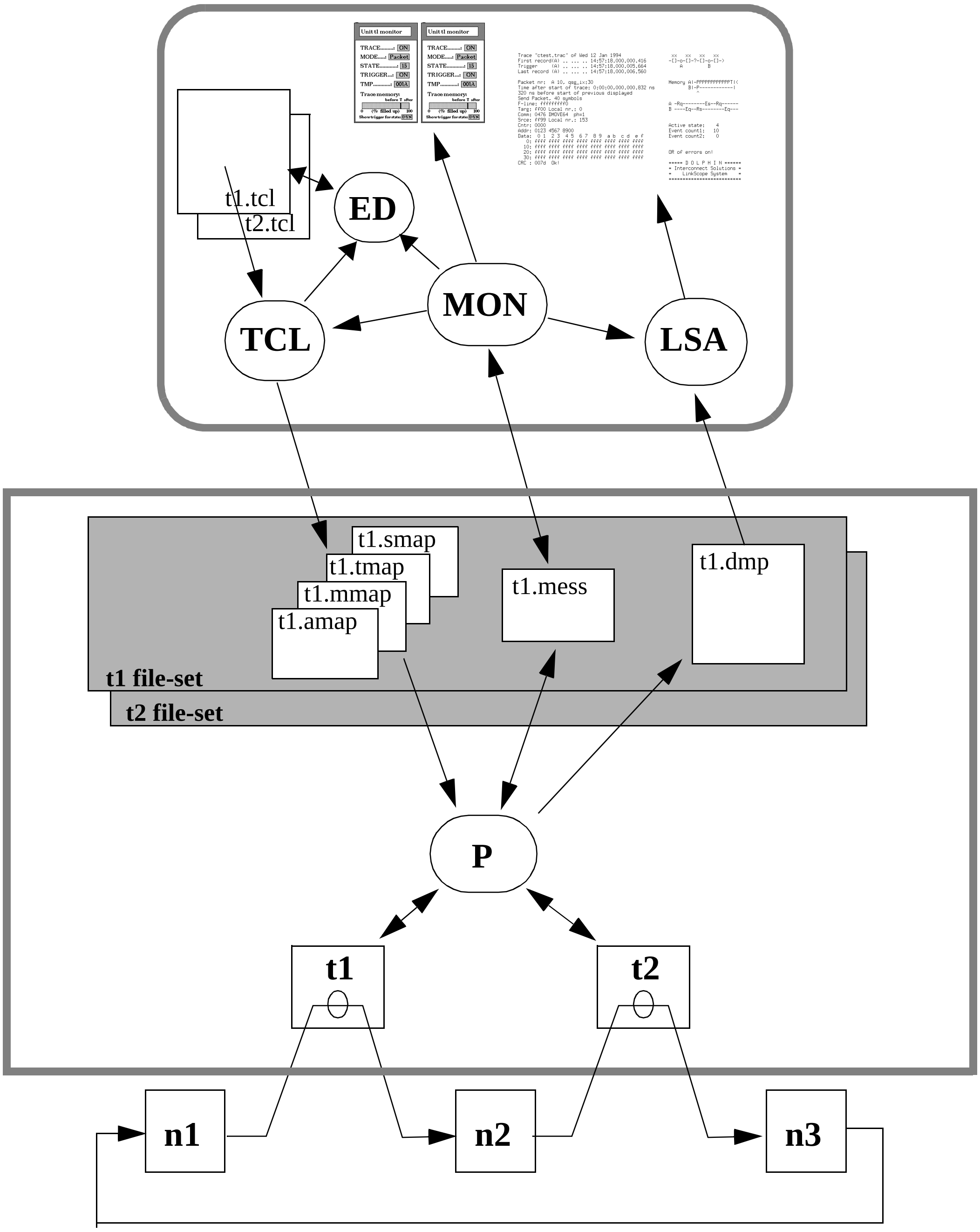
Two tracers in VME were produced for the SCI project at University of Oslo as partner in the CERN RD24 project.

The Tracer board is the basis for the SCITRAC instrument from SCILAB Technology



The Dolphin Tracer concept (1)

System overview:



Unit t1 monitor
TRACE.........: ON
MODE.....: Packet
STATE...........: 15
TRIGGER...: ON
TMP............: 001A
Trace memory:
before T after
0 (% filled up) 100
Show trigger for state: DYN

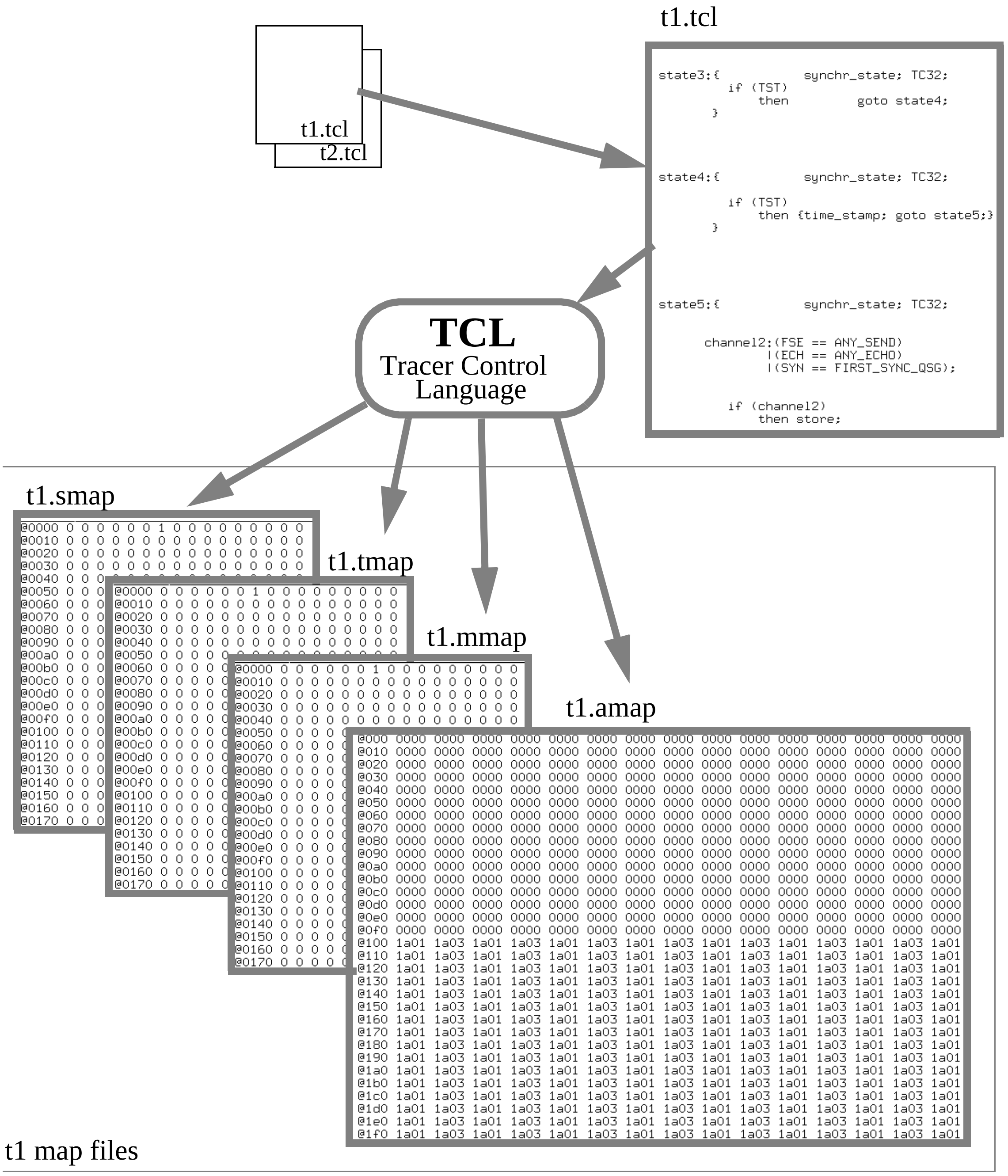
Unit t1 monitor
TRACE.........: ON
MODE.....: Packet
STATE...........: 15
TRIGGER...: ON
TMP............: 001A
Trace memory:
before T after
0 (% filled up) 100
Show trigger for state: DYN

```
Trace "ctest.trac" of Wed 12 Jan 1994
First record(A) .. ... .. 14:57:18.000.000.416
Trigger     (A) .. ... .. 14:57:18.000.005.664
Last record (A) .. ... .. 14:57:18.000.006.560

Packet nr:  A 10, qsg_ix:30
Time after start of trace: 0:00:00.000.000.832 ns
320 ns before start of previous displayed
Send Packet, 40 symbols
F-line: fffffffff0
Targ: ff00 Local nr.: 0
Comm: 0476 DMOVE64  ph=1
Srce: ff99 Local nr.: 153
Cntr: 0000
Addr: 0123 4567 8900
Data:  0 1  2 3  4 5  6 7  8 9  a b  c d  e f
   0: ffff ffff ffff ffff ffff ffff ffff ffff
  10: ffff ffff ffff ffff ffff ffff ffff ffff
  20: ffff ffff ffff ffff ffff ffff ffff ffff
  30: ffff ffff ffff ffff ffff ffff ffff ffff
CRC : 007d  Ok!
```

```
 xx   xx   xx   xx
-[]-o-[]-?-[]-o-[]->
   A         B

Memory A|-PPPPPPPPPPPPT|<
       B|-P------------|
           ^

A -Rq--------Es--Rq------
B ----Eq--Rs--------Eq---

Active state:    4
Event count1:   10
Event count2:    0

OR of errors on!

***** D O L P H I N ******
* Interconnect Solutions *
*    LinkScope System    *
**************************
```

The Dolphin Tracer concept (2)

TCL - the Tracer Control Language ™



t1.tcl

```
state3:{               synchr_state; TC32;
          if (TST)
              then          goto state4;
        }


state4:{               synchr_state; TC32;

          if (TST)
              then {time_stamp; goto state5;}
        }


state5:{               synchr_state; TC32;


       channel2:(FSE == ANY_SEND)
               |(ECH == ANY_ECHO)
               |(SYN == FIRST_SYNC_QSG);


          if (channel2)
              then store;
```

t1.smap

```
@0000 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
@0010 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
@0020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
@0030 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
@0040 0 0 0
@0050 0 0 0
@0060 0 0 0
@0070 0 0 0
@0080 0 0 0
@0090 0 0 0
@00a0 0 0 0
@00b0 0 0 0
@00c0 0 0 0
@00d0 0 0 0
@00e0 0 0 0
@00f0 0 0 0
@0100 0 0 0
@0110 0 0 0
@0120 0 0 0
@0130 0 0 0
@0140 0 0 0
@0150 0 0 0
@0160 0 0 0
@0170 0 0 0
```

t1.tmap

```
@0000 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
@0010 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
@0020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
@0030 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
@0040 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
@0050 0 0 0 0
@0060 0 0 0 0
@0070 0 0 0 0
@0080 0 0 0 0
@0090 0 0 0 0
@00a0 0 0 0 0
@00b0 0 0 0 0
@00c0 0 0 0 0
@00d0 0 0 0 0
@00e0 0 0 0 0
@00f0 0 0 0 0
@0100 0 0 0 0
@0110 0 0 0 0
@0120 0 0 0 0
@0130 0 0 0 0
@0140 0 0 0 0
@0150 0 0 0 0
@0160 0 0 0 0
@0170 0 0 0 0
```

t1.mmap

```
@0000 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
@0010 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
@0020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
@0030 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
@0040 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
@0050 0 0 0 0
@0060 0 0 0 0
@0070 0 0 0 0
@0080 0 0 0 0
@0090 0 0 0 0
@00a0 0 0 0 0
@00b0 0 0 0 0
@00c0 0 0 0 0
@00d0 0 0 0 0
@00e0 0 0 0 0
@00f0 0 0 0 0
@0100 0 0 0 0
@0110 0 0 0 0
@0120 0 0 0 0
@0130 0 0 0 0
@0140 0 0 0 0
@0150 0 0 0 0
@0160 0 0 0 0
@0170 0 0 0 0
```

t1.amap

```
@000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@020 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@030 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@040 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@050 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@060 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@070 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@080 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@090 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@0a0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@0b0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@0c0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@0d0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@0e0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@0f0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
@100 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@110 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@120 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@130 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@140 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@150 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@160 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@170 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@180 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@190 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@1a0 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@1b0 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@1c0 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@1d0 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@1e0 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
@1f0 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01 1a03 1a01
```

The Dolphin Tracer concept (3)

TCL - the Tracer Control Language II

```
#include "tclstdlib.h"
TCL {
      state1: {
         synchr_state;               /* stay at synchronous
state */
         channel1: (ECHO == 5678 ???? 1234 ????);
           /* ECHO: TRG CMD SRC CRC */
         if (channel1 && ECHO) then  {store;stop};
      }
    }
```

Tracing algorithm of this very simple program:

- recognize first echo packet with source address 1234H and destination address 5678H
- store it in Tracer memory, and
- stop tracing

Program syntax:

- synchr_state; whole state1 in synchronous trace
- channel1:(ECHO == 5678 ???? 1234 ????); specify the recognizing process channel, ? = don’t care
- && :connect channels linkage and event collective

The Dolphin Tracer concept (4)

TCL - the Tracer Control Language III

- **Developed by Bin Wu, Master thesis, Physics Dep, Oslo.**
- **The TCL compiler translates the source program to bit codes that are downloaded into the Tracer maps. The semantics in relation to the SCI protocols are also checked.**
- **TCL is strongly hardware dependent, it has been developed specially for the Tracer. The environment is UNIX, some of the syntax has been borrowed from C.**
- **Keywords which define the 16 states and the 4 event chains (channels) of the Tracer: state0, state1, ... state15, channel1, ... channel4**
- **Events: simple events, compound events are Boolean combination of two or more simple events**
- **Simple events: QSG events, keywords event0, event1, ... event15, time/counter event, external signal (trigger) event, CRC event. Examples of QSG events:**
  - event0 : asynchronous idle or QSG with all F = zero
  - event6 : first QSG of a send packet
  - event7 : last QSG of a send packet
  - event8 : second of three or more QSGs in a send packet
  - event9 : third of four or more QSGs in a send packet
- **Compound event example, values indicated by ... :**

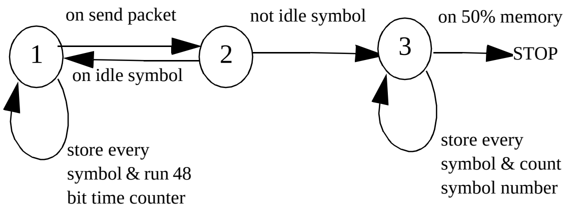
```
channel1:((event6 == ...)*(event8 == ...)
                                   *(event9 == ...)*(event7 == ...));
```

- **Action statements, examples:**
  store, synchr_state, transp_state, time_stamp, goto, reset

The Dolphin Tracer concept (5)

TCL - the Tracer Control Language IV

The programming is on the basis of finite state machines. Below is shown state diagram and a pseudo TCL program for detecting miss-idle-error during packet transmission.



```
TCL_program_begin

    state1_begin
      Run48bitTimeCounter;
      StoreEverySymbol;
      if (ComingPacketIsSendPacket)
          then goto state2;
   state1_end

   state2_begin
      Continue48bitTimeCounter;
      StoreEverySymbol;
      if (ComingPacketIsNotIdleSymbol)
           then { TimeStamp;
                  MarkTracerMemory;
                  ChangeTo16bitCounter1;
                  Counter1=0;
                  goto state3;
      } else goto state1;
   state2_end

   state3_begin
      channel_definition_begin
             channel1: Counter1 = 16K QSG; /* = 128 Kbyte */
      channel_definition_end

      StoreEverySymbol;
      if(Channel1IsTrue) then STOP;
      else Counter1 ++; /* increment Counter 1 on symbol */
   state3_end

TCL_program_end
```

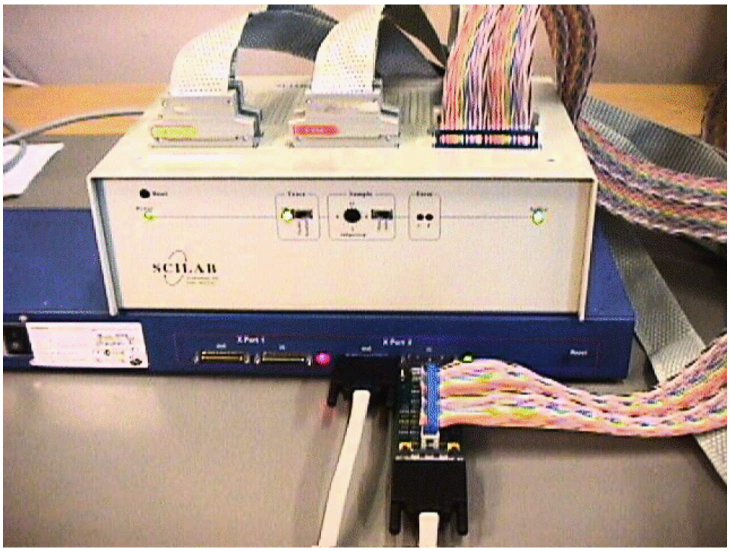
How SCI was developed without adequate hardware tools?

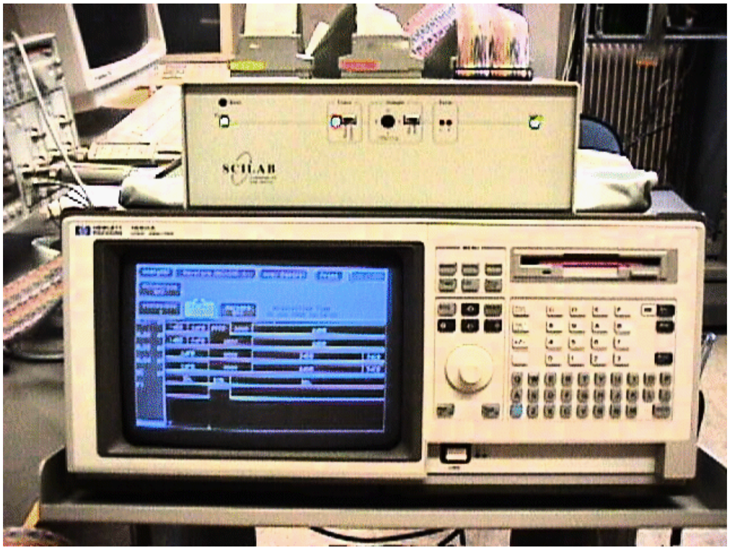
- The Dolphin tracer concept was never fully developed, and remained in the cupboard
- In fact, the lack of adequate hardware tools for SCI has probably had a significant negative feedback on the acceptance of SCI, and has certainly delayed the development of SCI products.

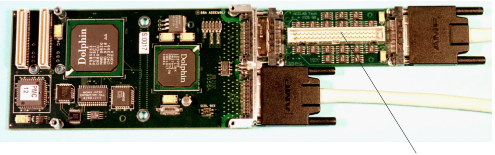
The SCITRAC link tracer

The SCITRAC SCI link tracer

- ♣ Commercialized in 1998 by SCILAB Technology AS, Oslo, based on a tracer card developed by Dolphin Interconnect Solutions Inc.

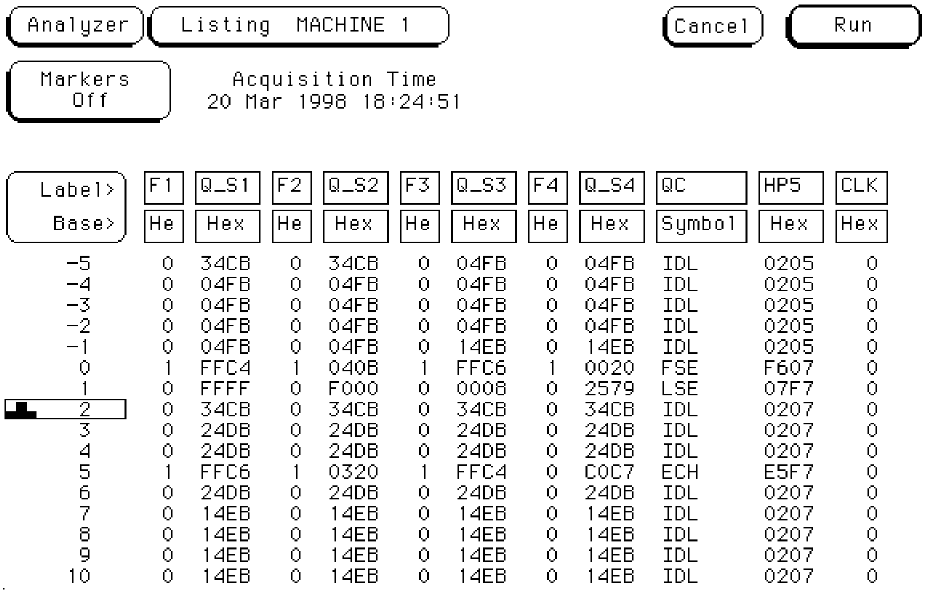






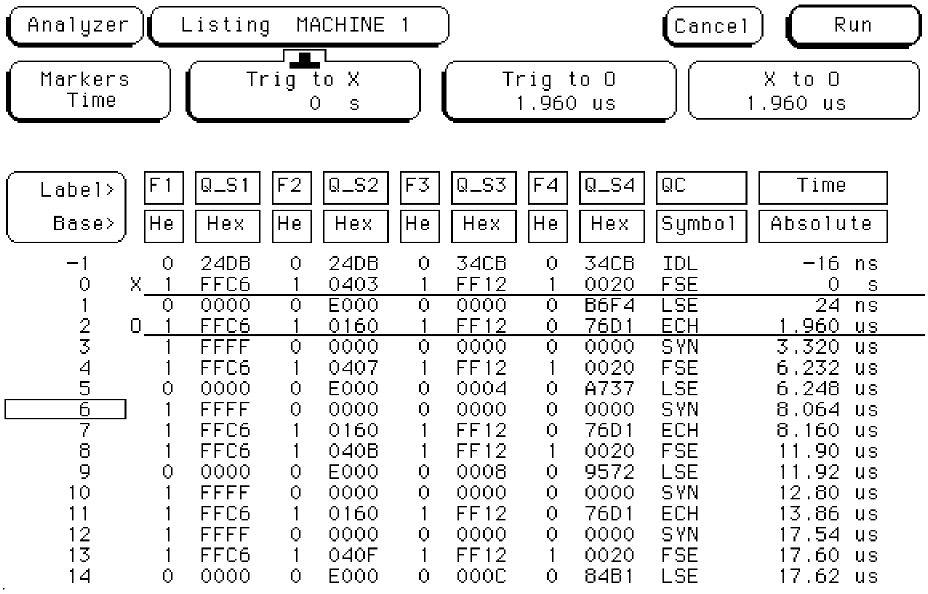
SCITRAC link probe inserted on link cable

Examples of SCITRAC Traces



```
Analyzer  Listing  MACHINE 1                    Cancel   Run

Markers      Acquisition Time
 Off      20 Mar 1998 18:24:51

 Label>  F1  Q_S1  F2  Q_S2  F3  Q_S3  F4  Q_S4  QC      HP5   CLK
  Base>  He  Hex   He  Hex   He  Hex   He  Hex   Symbol  Hex   Hex

   -5     0  34CB   0  34CB   0  04FB   0  04FB  IDL     0205   0
   -4     0  04FB   0  04FB   0  04FB   0  04FB  IDL     0205   0
   -3     0  04FB   0  04FB   0  04FB   0  04FB  IDL     0205   0
   -2     0  04FB   0  04FB   0  04FB   0  04FB  IDL     0205   0
   -1     0  04FB   0  04FB   0  14EB   0  14EB  IDL     0205   0
    0     1  FFC4   1  040B   1  FFC6   1  0020  FSE     F607   0
    1     0  FFFF   0  F000   0  0008   0  2579  LSE     07F7   0
    2     0  34CB   0  34CB   0  34CB   0  34CB  IDL     0207   0
    3     0  24DB   0  24DB   0  24DB   0  24DB  IDL     0207   0
    4     0  24DB   0  24DB   0  24DB   0  24DB  IDL     0207   0
    5     1  FFC6   1  0320   1  FFC4   0  C0C7  ECH     E5F7   0
    6     0  24DB   0  24DB   0  24DB   0  24DB  IDL     0207   0
    7     0  14EB   0  14EB   0  14EB   0  14EB  IDL     0207   0
    8     0  14EB   0  14EB   0  14EB   0  14EB  IDL     0207   0
    9     0  14EB   0  14EB   0  14EB   0  14EB  IDL     0207   0
   10     0  14EB   0  14EB   0  14EB   0  14EB  IDL     0207   0
```

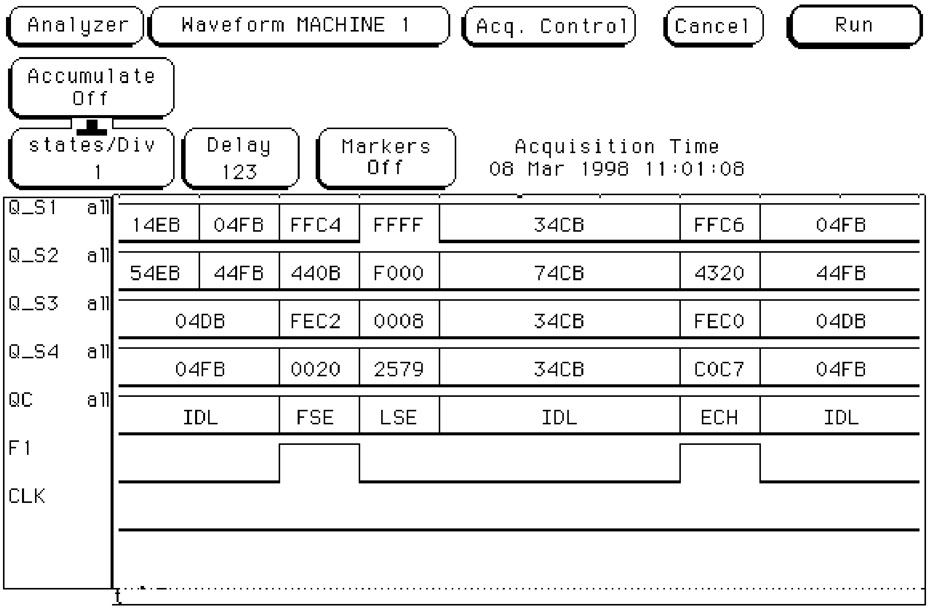
State listed trace, four and four (Quad) SCI symbols per line. The QC symbol identifies the packet type: **IDL**e, **ECH**o, **F**irst**SE**ndpacketquad, **L**ast**SE**ndpacketquad.



```
Analyzer  Listing  MACHINE 1                               Cancel   Run

Markers      Trig to X        Trig to O         X to O
 Time          0   s          1.960 us         1.960 us

 Label>    F1  Q_S1  F2  Q_S2  F3  Q_S3  F4  Q_S4  QC        Time
  Base>    He  Hex   He  Hex   He  Hex   He  Hex   Symbol    Absolute

   -1       0  24DB   0  24DB   0  34CB   0  34CB  IDL       -16 ns
    0   X   1  FFC6   1  0403   1  FF12   1  0020  FSE         0  s
    1       0  0000   0  E000   0  0000   0  B6F4  LSE        24 ns
    2   O   1  FFC6   1  0160   1  FF12   0  76D1  ECH     1.960 us
    3       1  FFFF   0  0000   0  0000   0  0000  SYN     3.320 us
    4       1  FFC6   1  0407   1  FF12   1  0020  FSE     6.232 us
    5       0  0000   0  E000   0  0004   0  A737  LSE     6.248 us
    6       1  FFFF   0  0000   0  0000   0  0000  SYN     8.064 us
    7       1  FFC6   1  0160   1  FF12   0  76D1  ECH     8.160 us
    8       1  FFC6   1  040B   1  FF12   1  0020  FSE     11.90 us
    9       0  0000   0  E000   0  0008   0  9572  LSE     11.92 us
   10       1  FFFF   0  0000   0  0000   0  0000  SYN     12.80 us
   11       1  FFC6   1  0160   1  FF12   0  76D1  ECH     13.86 us
   12       1  FFFF   0  0000   0  0000   0  0000  SYN     17.54 us
   13       1  FFC6   1  040F   1  FF12   1  0020  FSE     17.60 us
   14       0  0000   0  E000   0  000C   0  84B1  LSE     17.62 us
```

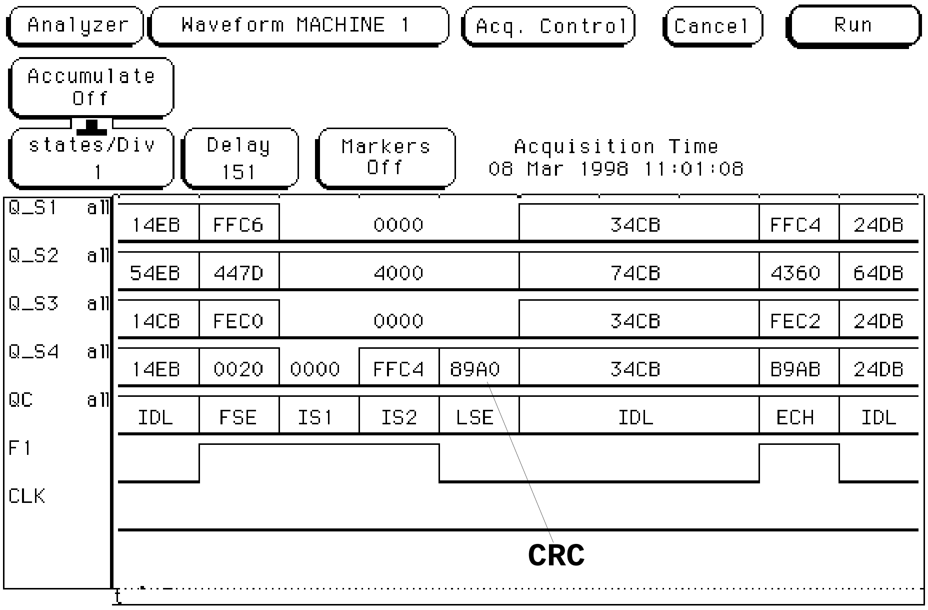
Time distribution of non-idle packets, only first Quad symb listed

Examples of SCITRAC Traces



```
Accumulate   states/Div   Delay   Markers   Acquisition Time
Off          1            123     Off       08 Mar 1998 11:01:08

Q_S1  all | 14EB | 04FB | FFC4 | FFFF | 34CB | FFC6 | 04FB
Q_S2  all | 54EB | 44FB | 440B | F000 | 74CB | 4320 | 44FB
Q_S3  all |    04DB     | FEC2 | 0008 | 34CB | FEC0 | 04DB
Q_S4  all |    04FB     | 0020 | 2579 | 34CB | C0C7 | 04FB
QC    all |    IDL      | FSE  | LSE  | IDL  | ECH  | IDL
F1
CLK
```

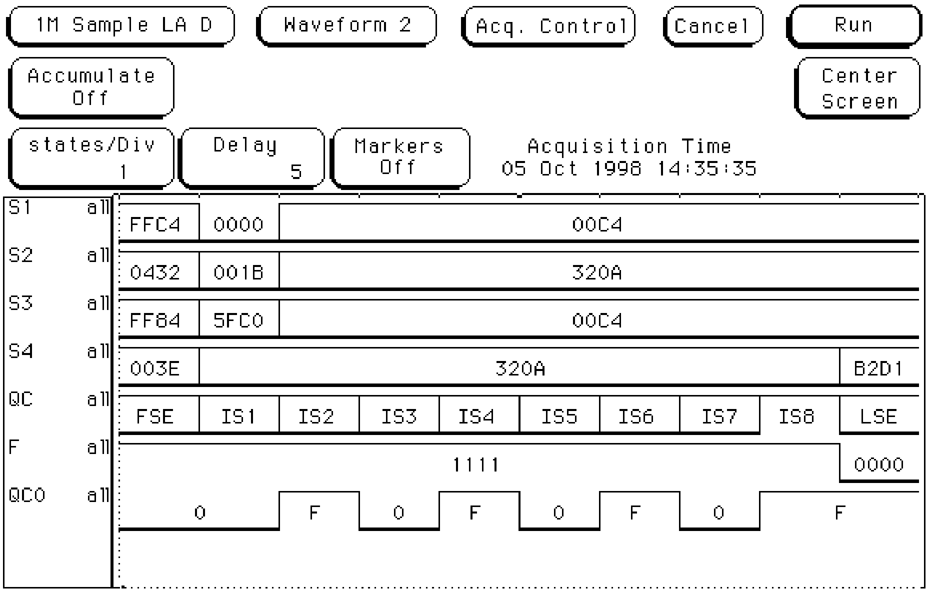
readsb request packet followed by an **echo** packet from a response packet



```
Accumulate   states/Div   Delay   Markers   Acquisition Time
Off          1            151     Off       08 Mar 1998 11:01:08

Q_S1  all | 14EB | FFC6 |          0000          | 34CB | FFC4 | 24DB
Q_S2  all | 54EB | 447D |          4000          | 74CB | 4360 | 64DB
Q_S3  all | 14CB | FEC0 |          0000          | 34CB | FEC2 | 24DB
Q_S4  all | 14EB | 0020 | 0000 | FFC4 | 89A0 | 34CB | B9AB | 24DB
QC    all | IDL  | FSE  | IS1  | IS2  | LSE  | IDL  | ECH  | IDL
F1
CLK
```

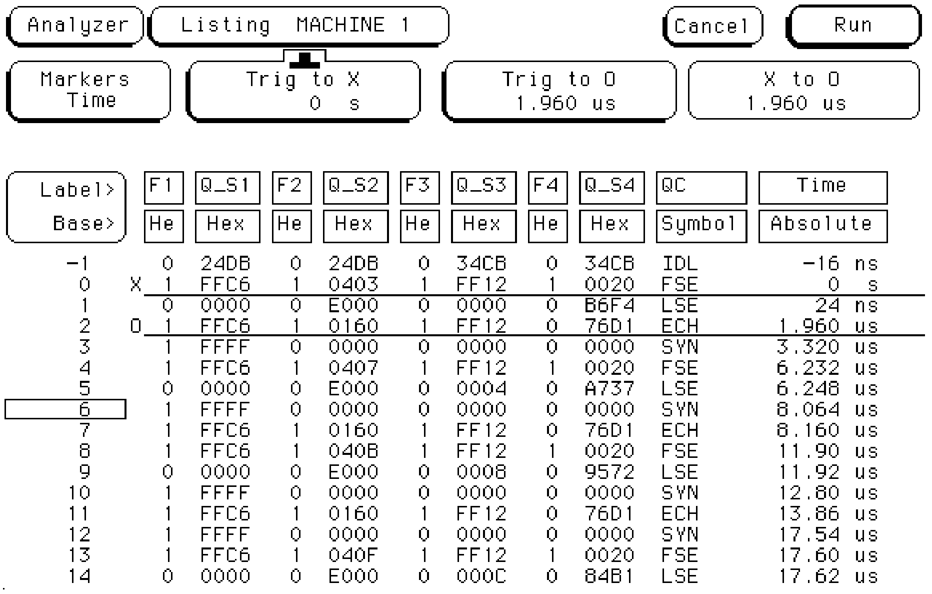
Response packet to the request shown above. The 16-bit CRC is contained in the last two bytes.

Examples of SCITRAC Traces



```
states/Div   1
Delay        5
Acquisition Time
05 Oct 1998 14:35:35
```

`nwrite64` packet (HP16500B Analyzer)



```
Trig to X      Trig to O      X to O
    0  s        1.960 us       1.960 us

Label>   F1  Q_S1  F2  Q_S2  F3  Q_S3  F4  Q_S4  QC      Time
Base>    He  Hex   He  Hex   He  Hex   He  Hex   Symbol  Absolute

 -1       0  24DB   0  24DB   0  34CB   0  34CB  IDL       -16 ns
  0   X   1  FFC6   1  0403   1  FF12   1  0020  FSE         0  s
  1       0  0000   0  E000   0  0000   0  B6F4  LSE        24 ns
  2   O   1  FFC6   1  0160   1  FF12   0  76D1  ECH     1.960 us
  3       1  FFFF   0  0000   0  0000   0  0000  SYN     3.320 us
  4       1  FFC6   1  0407   1  FF12   1  0020  FSE     6.232 us
  5       0  0000   0  E000   0  0004   0  A737  LSE     6.248 us
  6       1  FFFF   0  0000   0  0000   0  0000  SYN     8.064 us
  7       1  FFC6   1  0160   1  FF12   0  76D1  ECH     8.160 us
  8       1  FFC6   1  040B   1  FF12   1  0020  FSE     11.90 us
  9       0  0000   0  E000   0  0008   0  9572  LSE     11.92 us
 10       1  FFFF   0  0000   0  0000   0  0000  SYN     12.80 us
 11       1  FFC6   1  0160   1  FF12   0  76D1  ECH     13.86 us
 12       1  FFFF   0  0000   0  0000   0  0000  SYN     17.54 us
 13       1  FFC6   1  040F   1  FF12   1  0020  FSE     17.60 us
 14       0  0000   0  E000   0  000C   0  84B1  LSE     17.62 us
```

List trace of time distribution of non-IDLE packets

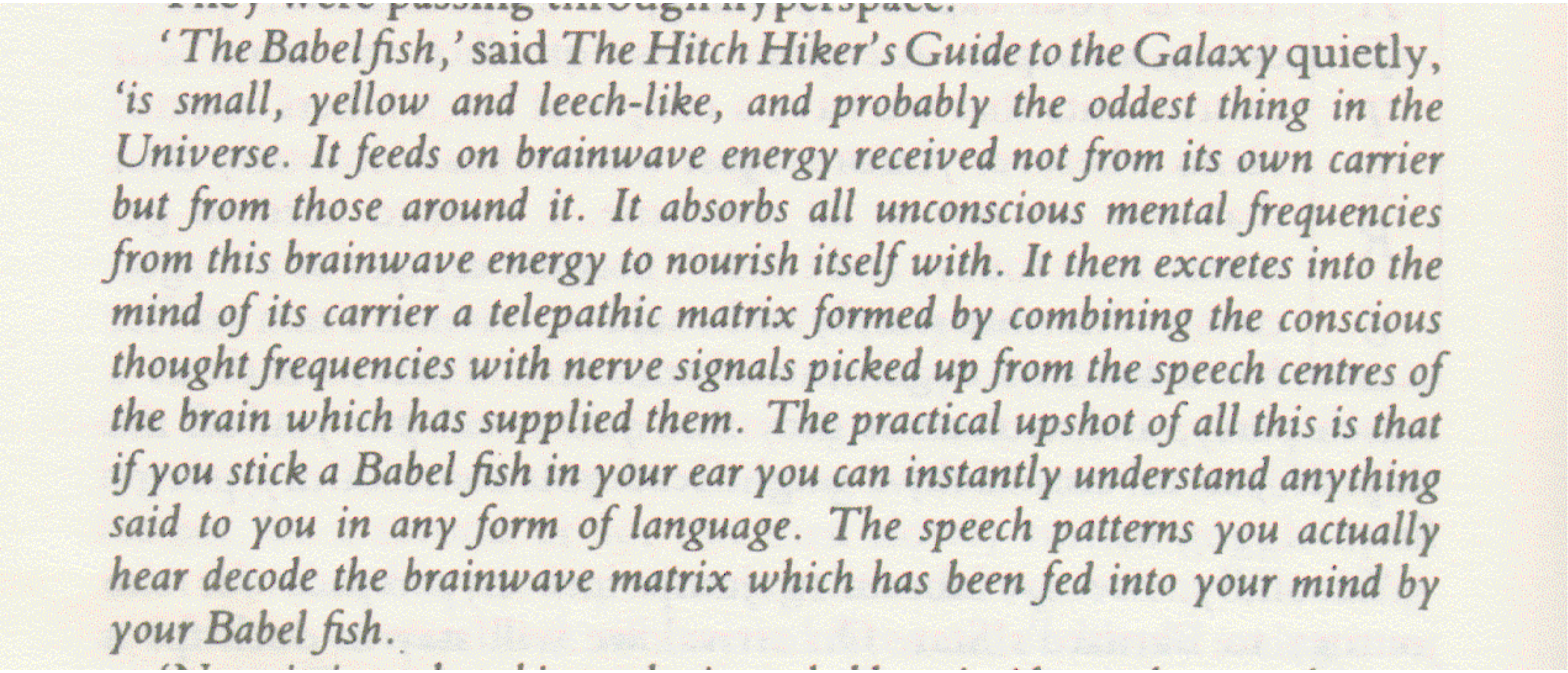
SCIview

From *SCITRAC* to *SCIview* - why?

- The maximum clock frequency for SCITRAC is around 160 MHz. To match the LC3 clock rate a new design was needed.
- In addition to the standard SCITRAC features, SCILAB Technology believe(d) that there should be a significant market for a flexible instrumentation that can log multi-dimensional traffic, i.e. time-correlated node-to-node traffic in a SCI cluster. Another application could be verification of no packet loss through a switch, and measurement of packet transmission time distribution.
- The use of programmable logic in implementation offers a superior flexibility in defining measuring configurations and trigger definitions. The challenge is that with even the highest speed-grade field programmable circuits (FPGA) a highly optimized design is required in order to reach a 1000 MB/s target.
- A programmable implementation can also serve as the basis for future instrumentation for other high-speed packet oriented links (Infiniband, others) to be developed by the company.
- Therefore the design of the SCIview link tracer system was launched in 1999. The SCIview link probe was given the project name “Babelfish”.

“Babelfish”

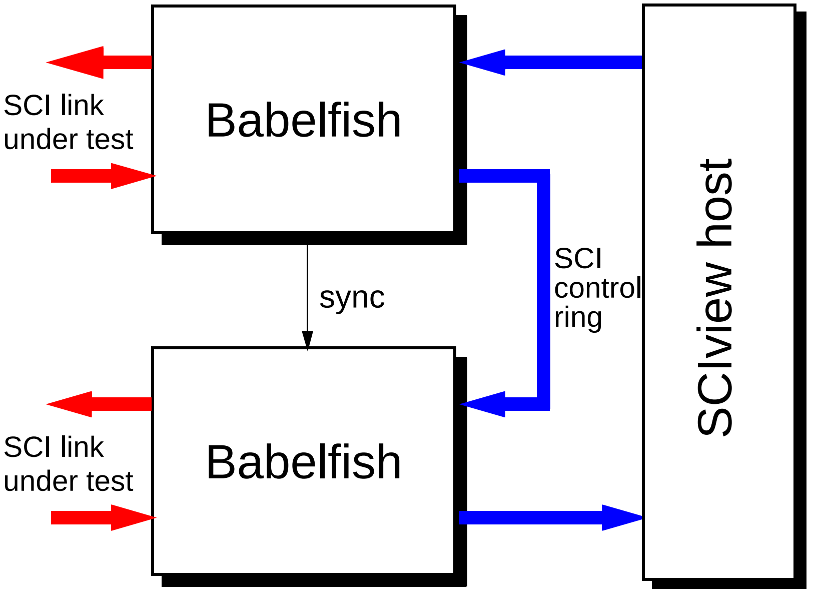
From the classic “The Hitch Hiker’s Guide to the Galaxy” by Douglas Adams, William Heinemann, London:

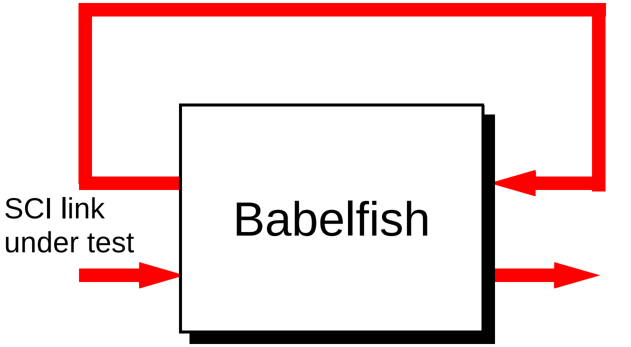


They were passing through hyperspace.

‘The Babel fish,’ said *The Hitch Hiker’s Guide to the Galaxy* quietly, *‘is small, yellow and leech-like, and probably the oddest thing in the Universe. It feeds on brainwave energy received not from its own carrier but from those around it. It absorbs all unconscious mental frequencies from this brainwave energy to nourish itself with. It then excretes into the mind of its carrier a telepathic matrix formed by combining the conscious thought frequencies with nerve signals picked up from the speech centres of the brain which has supplied them. The practical upshot of all this is that if you stick a Babel fish in your ear you can instantly understand anything said to you in any form of language. The speech patterns you actually hear decode the brainwave matrix which has been fed into your mind by your Babel fish.*

SCIview tracing configurations #1

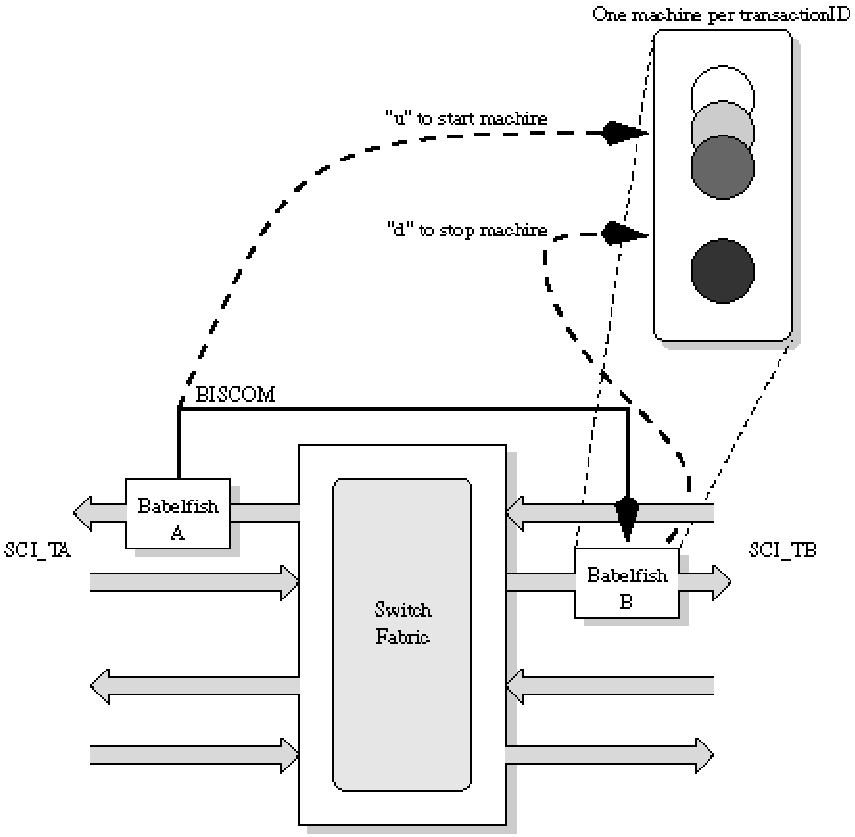




Multilink tracing: a number of Babelfish'es inserted non-obtrusively in a SCI fabric and controlled by a separate SCI control ring. The control ring is used both for control and for streaming of trace data to the SCIview host.

In-fabric configuration: a lonely Babelfish inserted in-fabric, acting as system monitor, or traffic generator (like the SCIEurope TLE).

SCIview tracing configuration #2 - switch characteristics



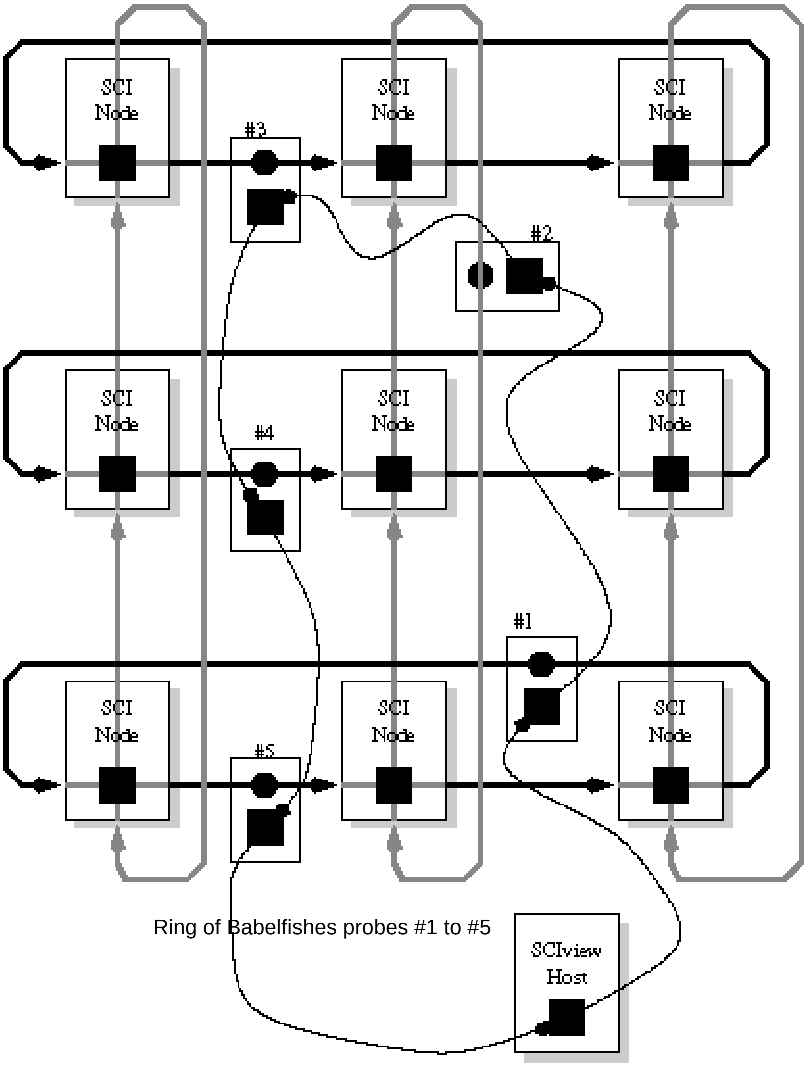
- ♠ Verification of no packet loss
- ♠ Measure retry traffic
- ♠ Measure latency distribution

When a packet specific trigger is generated by in-port Babelfish A, the transactionId is sent to Babelfish B. The task of B is to match the information it receives from A with what it snoops from the out-port. Any mismatch must be reported as error.

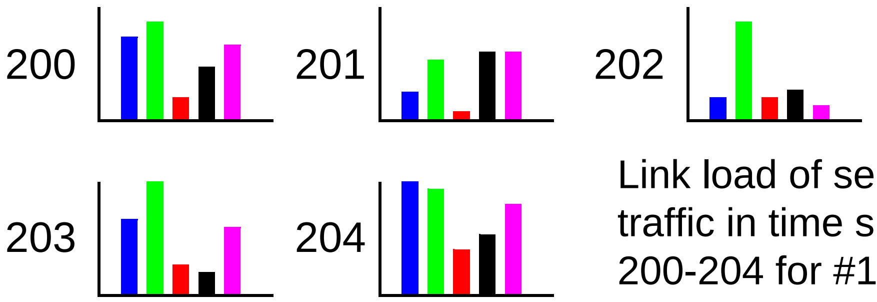
To handle all 64 transactionID’s of the LC, B contains a firmware with 64 state machines. The transmission time can be measured with a resolution of 25 ns for 200 MHZ link clock.

SCIview tracing configurations #3

Measuring correlated link activity in a SCI cluster



Ring of Babelfishes probes #1 to #5

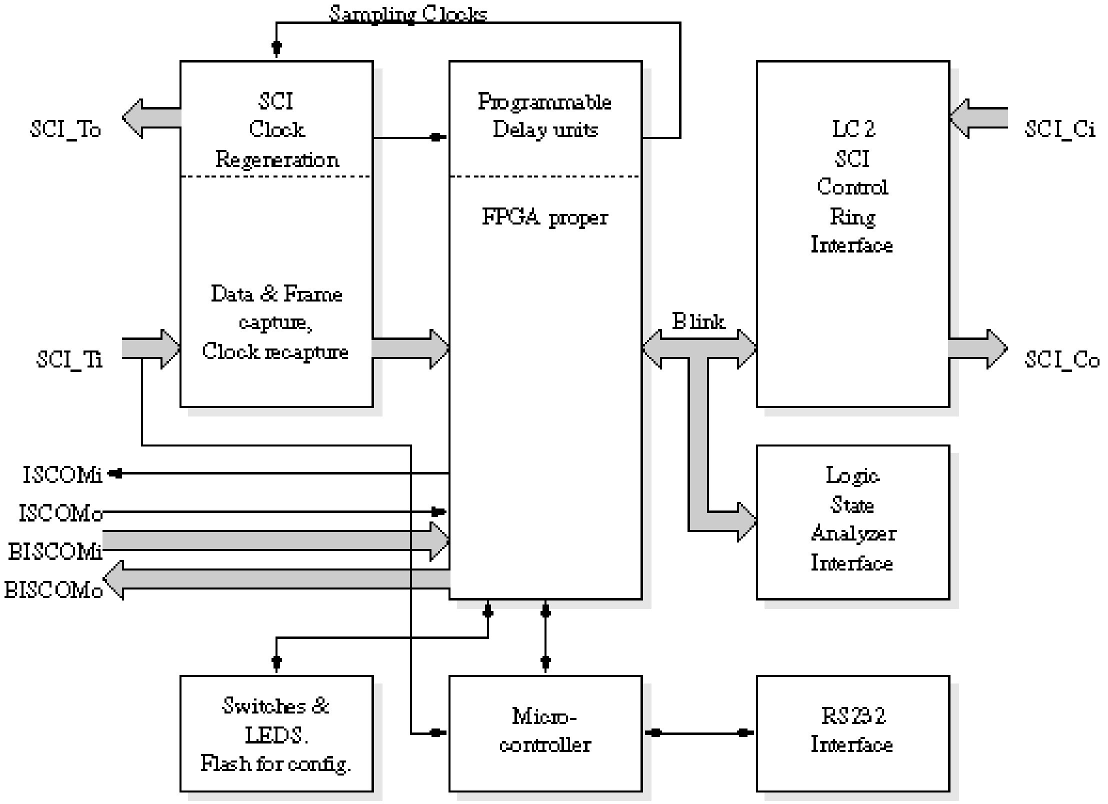


Link load of selected traffic in time slots 200-204 for #1 to #5

Main hardware block diagram of *Babelfish*

The SCI link signals-under-test pass through Babelfish via the SCI_T_i and SCI_T_o connectors. After level conversion from LVDS to CMOS the link data enter the FPGA to be processed by the user defined trigger program.

The result can be used to for instance increment event counters, or transmitted to the SCI control ring SCI_C via an on-board LC2 (later LC3), or monitored by a logical state analyzer via five 18-signals pods (SCIview-200 only).



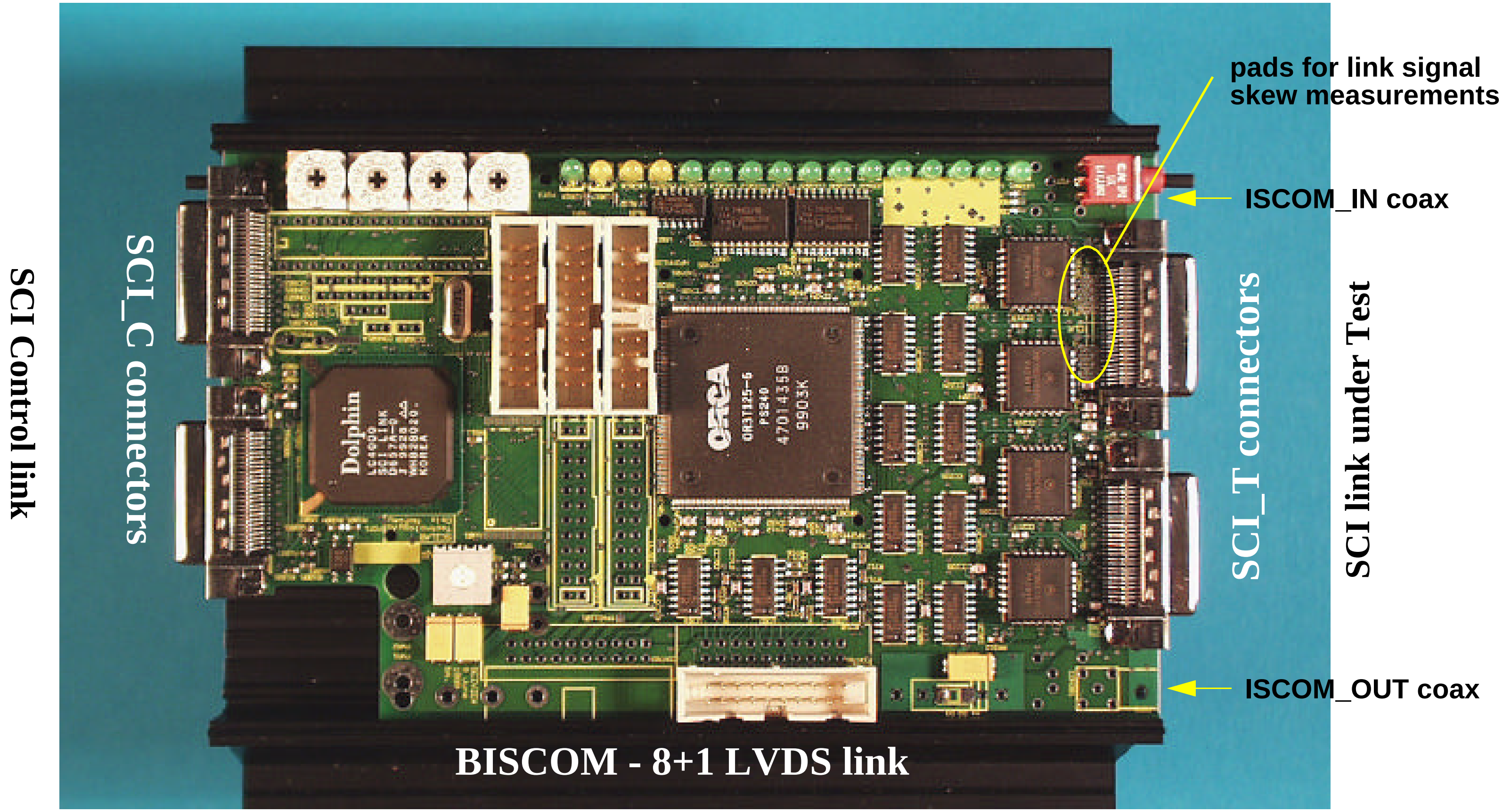
Babelfish Firmware

The FPGA (OR3T125 from Lucent Technology Inc.) contains the VHDL-based code that defines the firmware. A specific configuration can be rapidly downloaded via the on-board microcontroller to the flash memory. Some firmware modules:

- **Generation and verification of Extended SCI Quad Symbol Codes.**
- **Trigger unit, using SRAM internal to the FPGA. The content of SRAM defines the actual trigger condition, and is downloaded at run-time. Up to 16 triggers that span four SCI symbols can be entered.**
- **48-bit Timer Register counting up the number of SCI_T clock cycles, used to time-stamp events. The content of the Timer Register of all interconnected Babelfish'es can by synchronized (reset) via a coax daisy chain.**
- **32-bits Event Counters. An event counter is normally incremented by a trigger.**
- **LSA (Logic State Analyzer) Output Unit. In LSA presentation mode the captured traffic is streamed to the LSA Pods.**
- **InterSCIview Communication - ISCOM. Synchronization of time-stamp register, and some lower bandwidth (some 10's of MHz) can be transmitted between Babelfish'es through a daisy-chained LVDS 1-signal point-to-point connection.**
- **Bused InterSCIview Communication - BISCOM. In some situations, see description of a switch verification configuration, a higher bandwidth, low latency channel between Babelfish'es is needed. BISCOM consists of 8 data bits and clock, each board has one input and one output connector.**

Top view of *Babelfish*

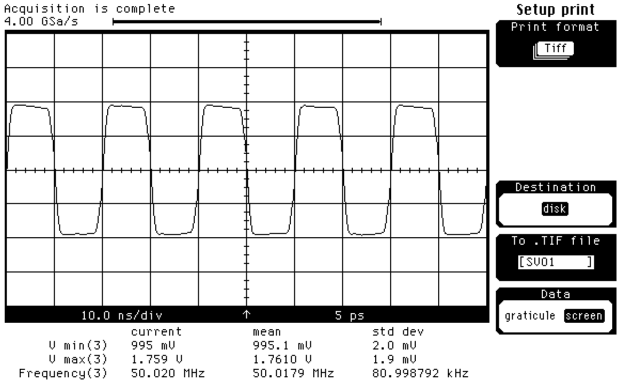
Size 160x100 mm^2 (single Euro). Two LSA connectors, the flash memory and some other components are not mounted for this prototype. The card is mounted in a heat sink profile.



SCI link signal verification and skew measurements

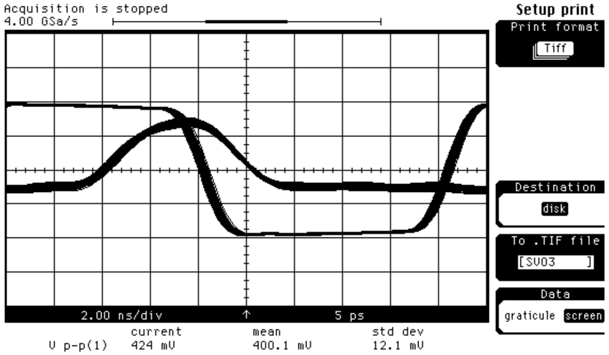
Babelfish supplies via a coax a regenerated divide-by-two clock ECL version of the incoming SCI clock, and measurement points for all link signals with inherent skew less than 2 ps.

Regenerated /2
100 MHz SCI clock
10 ns/cm



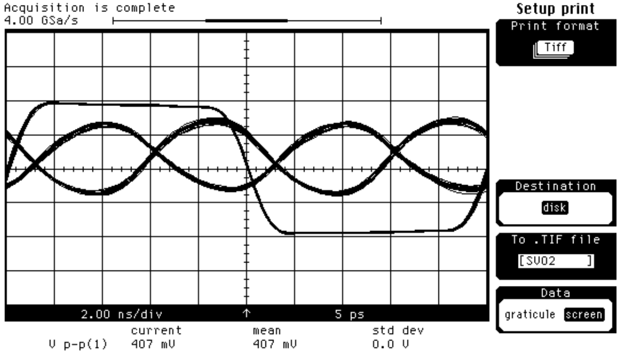
```
Acquisition is complete
4.00 GSa/s
10.0 ns/div                 5 ps
                current       mean          std dev
     V min(3)   995 mV        995.1 mV      2.0 mV
     V max(3)   1.759 V       1.7610 V      1.9 mV
 Frequency(3)   50.020 MHz    50.0179 MHz   80.998792 kHz
```

link FLAG pos
+ regen. clock



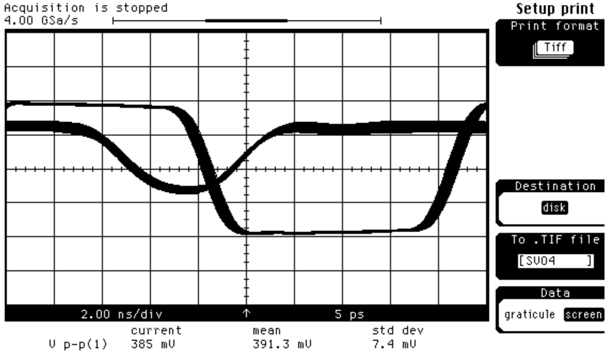
```
Acquisition is stopped
4.00 GSa/s
2.00 ns/div                 5 ps
           current    mean       std dev
 V p-p(1)  424 mV     400.1 mV   12.1 mV
```

Regenerated +
SCI link clock
2 ns/cm
multiple traces



```
Acquisition is complete
4.00 GSa/s
2.00 ns/div                 5 ps
           current    mean       std dev
 V p-p(1)  407 mV     407 mV     0.0 V
```

link FLAG neg
+ regen. clock



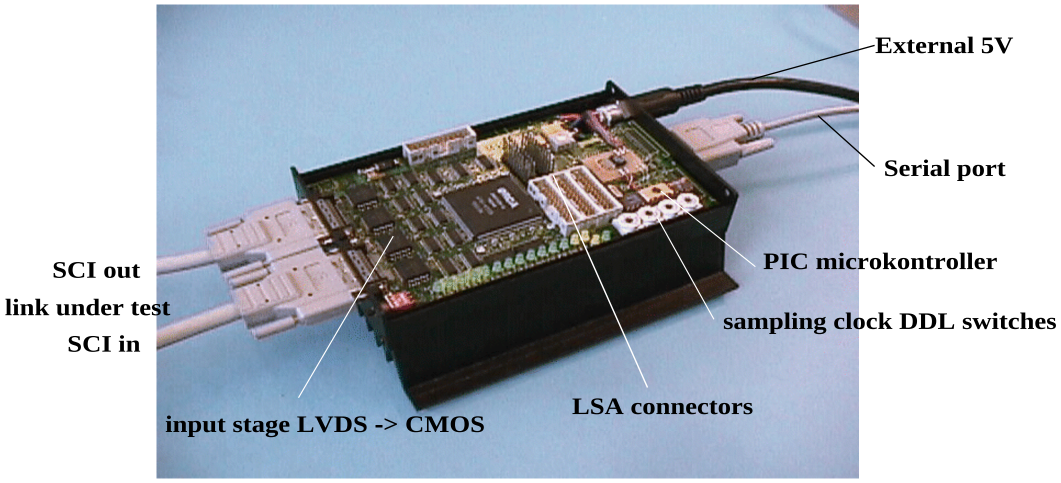
```
Acquisition is stopped
4.00 GSa/s
2.00 ns/div                 5 ps
           current    mean       std dev
 V p-p(1)  385 mV     391.3 mV   7.4 mV
```

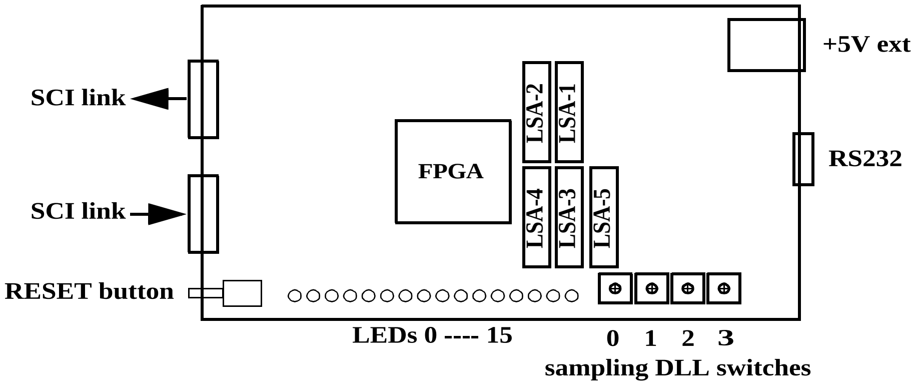
The SCIview *Babelfish* link probe

The basic version is delivered without on-board Link Controller, all communication goes through the PIC microcontroller and serial port. Standard tracing features:

- **SCITRAC functionality, plus:**
- **On-the-fly CRC generation and verification against the packet CRC.**

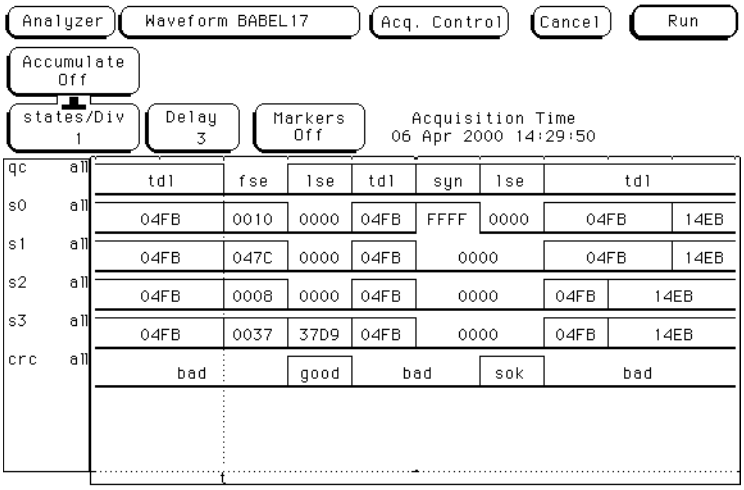
The tracer is a reconfigurable instrument, and firmware for features such as triggers that span multiple SCI symbols, generation of packet statistics, and also fast synchronization between several tracers, can be delivered on request. New versions of the firmware is downloaded to the EEPROM via the serial post.





Some *Babelfish* traces

Examples of *Babelfish* traces, recorded with a HP 1660A analyzer, are shown in . Note that only the CRC values for “lse” QSGs are relevant.



Accumulate Off

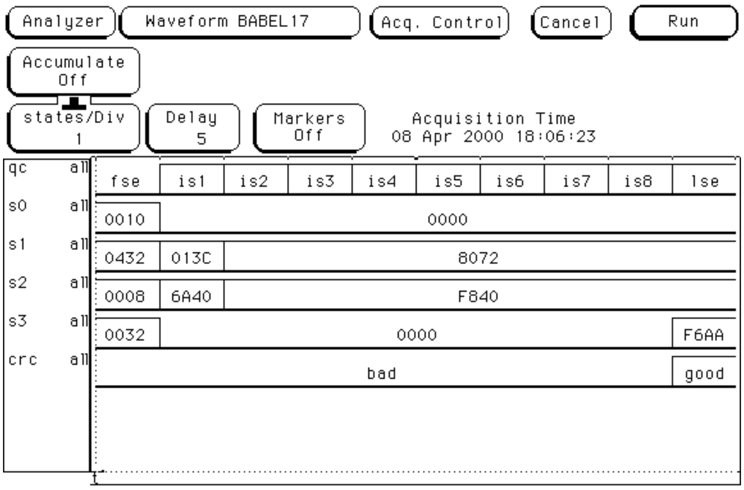
states/Div 1

Delay 3

Markers Off

Acquisition Time 06 Apr 2000 14:29:50

```
qc   all |      tdl     |  fse  |  lse  |  tdl  |  syn  |  lse  |          tdl          |
s0   all |     04FB     |  0010 |  0000 |  04FB |  FFFF |  0000 |     04FB     |  14EB  |
s1   all |     04FB     |  047C |  0000 |  04FB |      0000     |     04FB     |  14EB  |
s2   all |     04FB     |  0008 |  0000 |  04FB |      0000     |  04FB  |     14EB     |
s3   all |     04FB     |  0037 |  37D9 |  04FB |      0000     |  04FB  |     14EB     |
crc  all |          bad         |  good |      bad      |  sok  |          bad          |
```



Accumulate Off

states/Div 1

Delay 5

Markers Off

Acquisition Time 08 Apr 2000 18:06:23

```
qc   all |  fse | is1  | is2  | is3  | is4  | is5  | is6  | is7  | is8  | lse  |
s0   all | 0010 |                          0000                               |
s1   all | 0432 | 013C |                       8072                           |
s2   all | 0008 | 6A40 |                       F840                           |
s3   all | 0032 |                       0000                          | F6AA |
crc  all |                             bad                            | good |
```

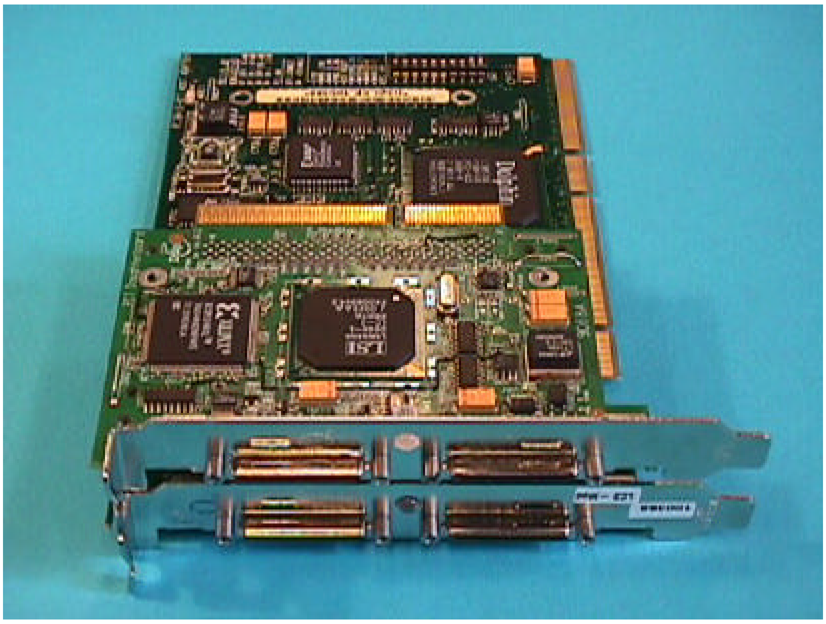
Babelfish performance

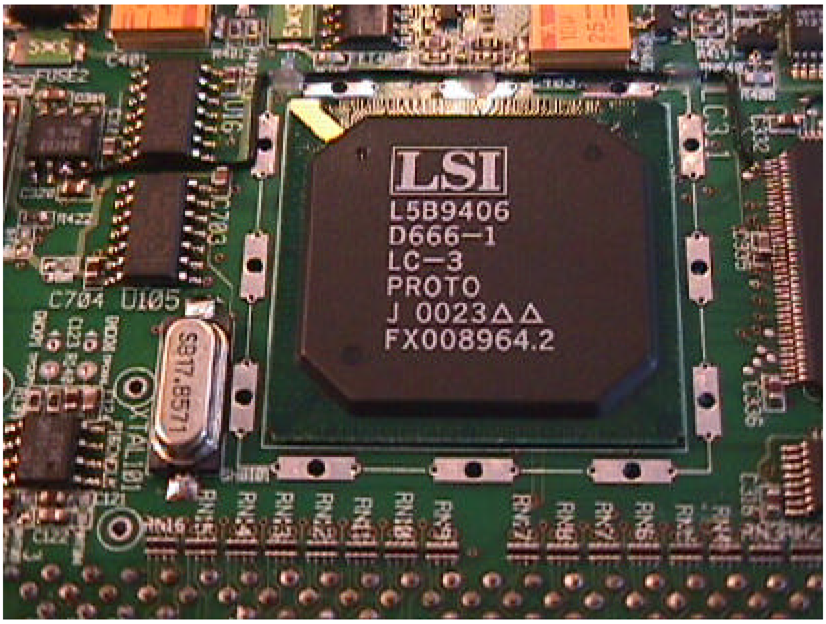
Design goal: 200 MHz link clock

Actual performance: 166 MHz (Dolphin standard)

Why?

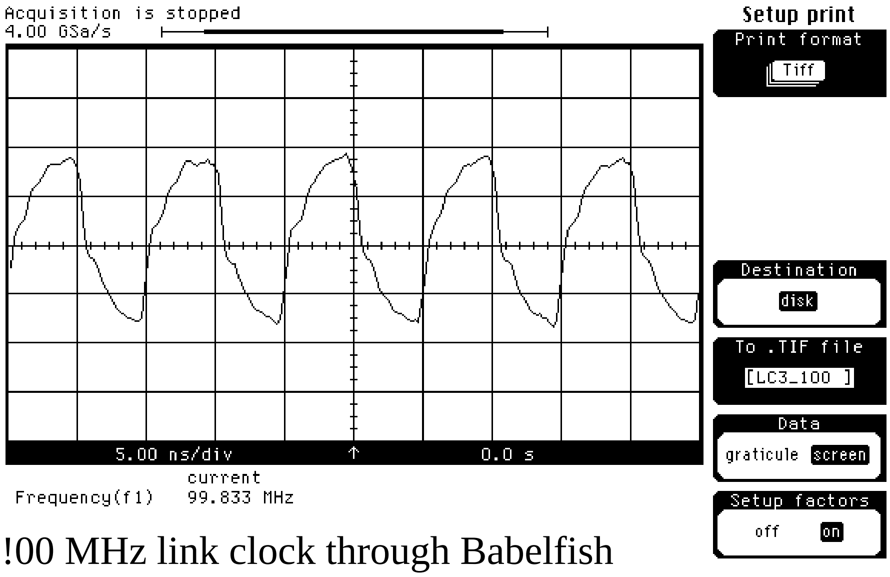
Verification with Dolphin LC3 mezzanine prototype card:





The real life

The passage through the *Babelfish* card exposes the link signals to at least three likely impedance mismatches, 1) input connector, 2) card routing and pick-up circuitry, and 3) output connector.



Acquisition is stopped
4.00 GSa/s

5.00 ns/div ↑ 0.0 s

Frequency(f1) current 99.833 MHz

Setup print

Print format: Tiff

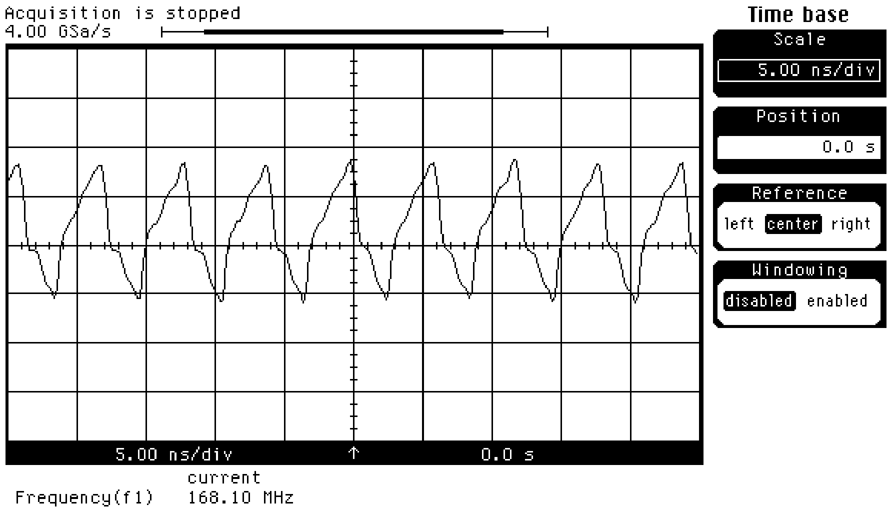
Destination: disk

To .TIF file: [LC3_100]

Data: graticule [✓ screen]

Setup factors: off [✓ on]

!00 MHz link clock through Babelfish



Acquisition is stopped
4.00 GSa/s

5.00 ns/div ↑ 0.0 s

Frequency(f1) current 168.10 MHz

Time base

Scale: 5.00 ns/div

Position: 0.0 s

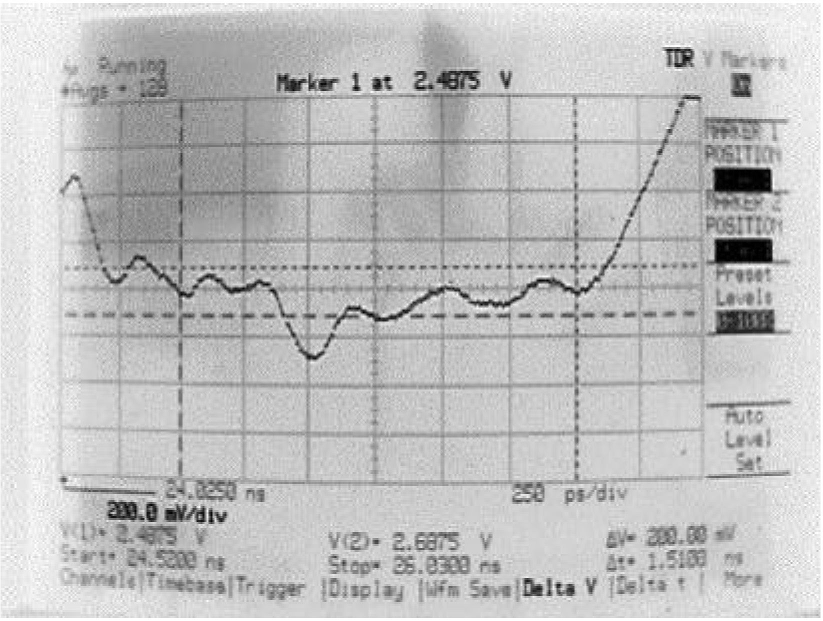
Reference: left [✓ center] right

Windowing: [✓ disabled] enabled

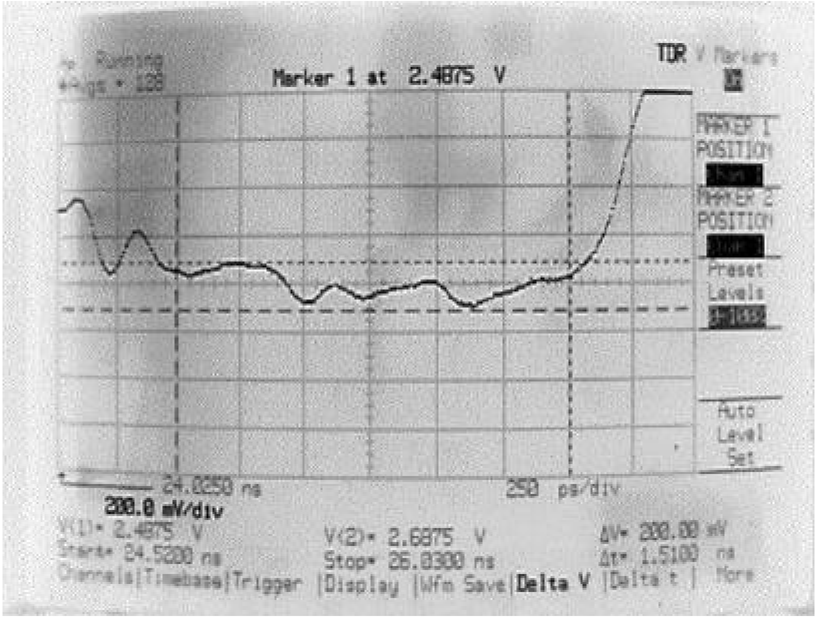
!68 MHz link clock through Babelfish

Impedance measurements of *Babelfish* card

Using Time Domain Reflectrometry the impedance through along a signal trace through the card can be measured. Below the link CLK impedance trace is shown on the left, and the FLAG trace on the right. A large impedance jump is seen on the clock line, furthermore, impedance mismatches are present for the in/out connectors (cable + card).



```
Av Running
#Avgs = 128      Marker 1 at  2.4875  V
24.0250 ns                    250  ps/div
200.0 mV/div
V(1)= 2.4875  V     V(2)= 2.6875  V     ΔV= 200.00 mV
Start= 24.5200 ns   Stop= 26.0300 ns    Δt= 1.5100  ns
```



```
Av Running
#Avgs = 128      Marker 1 at  2.4875  V
24.0250 ns                    250  ps/div
200.0 mV/div
V(1)= 2.4875  V     V(2)= 2.6875  V     ΔV= 200.00 mV
Start= 24.5200 ns   Stop= 26.0300 ns    Δt= 1.5100  ns
```

SCIview: conclusions so far

- With the current design - ECL circuitry for LVDS to CMOS conversion of link signals to match the specs and the speed-grade of the OR3T125 from Lucent Technology Inc.- the *Babelfish* tracer can just reach the 200 MHz as far as the FPGA is concerned, but the card needs to be redesigned in order to obtain better impedance control.
- However, with a design based on the newer Xilinx VIRTEX-E FPGA family, even a 250 MHz target can be reached. Furthermore, since the VIRTEX-E have LVDS compatible inputs, the design can be made much more compact.
- However, the market interest and demand for SCIview has been lower than expected. The number of companies doing SCI card hardware design is rather limited! No request from cluster software R&D labs.
- SCILAB Technology will therefore only launch a redesign if there is significant market demand. High price for small production volume.
- On the credit side, the SCIview project has given the company a strong platform for very-high-speed FPGA design, which has resulted in a significant new business.

The end (of the talk)