SCI Europe number:	P25257
Deliverable number:	D2.2.1c
Contractual date:	30th April, 1999
Work package:	2.2.1
Document version:	1.0
Document status:	Complete
Confidentiality:	Consortium
Document date:	April, 1999

Deliverable

Prototype Tracer Software

Partner: Trinity College Dublin
Author(s): M.Manzke, B.A.Coghlan
Editor: M.Manzke

Keywords: SCI, tracer, analyzer, software, database

Abstract:

In this document we present the preliminary reference manual for the prototype tracer software developed within the project.

esprit project 25257

ESPRIT Project P25257 SCIEurope

Deliverable D 2.2.1c

Prototype Tracer Software

April 1999

M.Manzke
Department of Computer Science
Trinity College Dublin
michael.manzke@cs.tcd.ie

Dr.B.A.Coghlan
Department of Computer Science
Trinity College Dublin
coghlan@cs.tcd.ie

esprit project
25257

Introduction

The definition of Task 2.2.1 is as follows :

Task	**2.2.1**	**Test Tools Development**		
Market and User Need	There are no commercially available SCI test tools on the market for the SCI community today.			
Objectives	To develop the first generation of tracing and debugging tools for use in work package 3 Applications.			
Approach	The tools will be based on needs identified in the Test Requirements Specification from Task 2.1. There will probably be developed two tools – one tool able to trace the SCI traffic and either show online or store the results. This tool will be based as much as possible on present hardware and software platforms. The other tool will be able to send and receive SCI traffic according to some traffic profile in order to load systems with traffic without using real nodes. The prototype tools will be evaluated during the debugging phase of the Embedded Avionics System demonstrator in Task 3.3, and the results will be summarised in a report.			
Lead Partner	Trinity	24 person months		
Other Partners	D.E. SINTEF	4 person months 12 person months		
Major Deliverables	D 2.2.1 D 2.2.2 D 2.2.3	Q4 Q6 Q8	Trinity SINTEF Trinity	Prototype Tracer/Analyzer Traffic Generation Tool Tracer/Analyzer Mk.II

The objective of this document is to present the preliminary Reference Manual and design specification for the Prototype Tracer Software. This represents the software resources of the Prototype Tracer/Analyzer.

Contents

1. Overview

This general introduction to the SCI Trace Instrument is followed by a detailed specification of the instrument's software components.

1.1 SCI Trace Instrument

Trinity College Dublin is developing this tool as part of the SCI Europe Esprit project.

Cluster performance analysis involves the collection and analysis of traces obtained from a target computer system. By providing a view of the target's interconnect traffic versus time, a designer can determine how efficiently the target system is operating.

This technology enables the designer to analyse a number of system characteristics such as where a processor is spending most of its time, how long critical routines take to execute, how often a cache contains the desired information, how well a cache coherency protocol performs, whether there is excessive contention for locks, how well the load is balanced among processors, and so on. If performance bottlenecks can be located, corrective action can hopefully be taken.

State-of-the-art logic analysers collect traces in real-time, but their capacity is generally limited to the order of 1to 8K samples. The DT200.1 Deep Tracer is a modular data collection system designed specifically for gathering very long state traces for performance analysis of processor and I/O busses. Conceptually the system functions as a very deep FIFO. Data is sampled via the 48 bit serial interface on the positive edge of a clock and stored in a 12MByte VRAM buffer. The buffer wraps around and can be read out via the EISA interface without stopping the sample clock [B.A.Coghlan et al, 1998].

1.2 Trace System Hardware

The trace board's pick-up-card snoops on the Blink™ of Dolphin's PCI to SCI Bridge (PSB) or may potentially be used as trace data acquisition instrument for SCILabs SCI link tracer. The trace data decoding software and the database are designed to process SCI Packets and Blink encapsulated SCI Packets.

1.3 Blink™

Blink™ is Dolphin's physical realisation of the transfer cloud specified in [IEEE 1596, 1992]. Typical components interfacing to Blink™ include link controllers (Blink™ to SCI interface), memory controllers, cache controllers, and bridges to other interconnect systems, such as PCI. A Blink™ packet is composed of a SCI packed core, a Blink™ extension header and a Blink™ extension trailer [Dolphin, 1996].

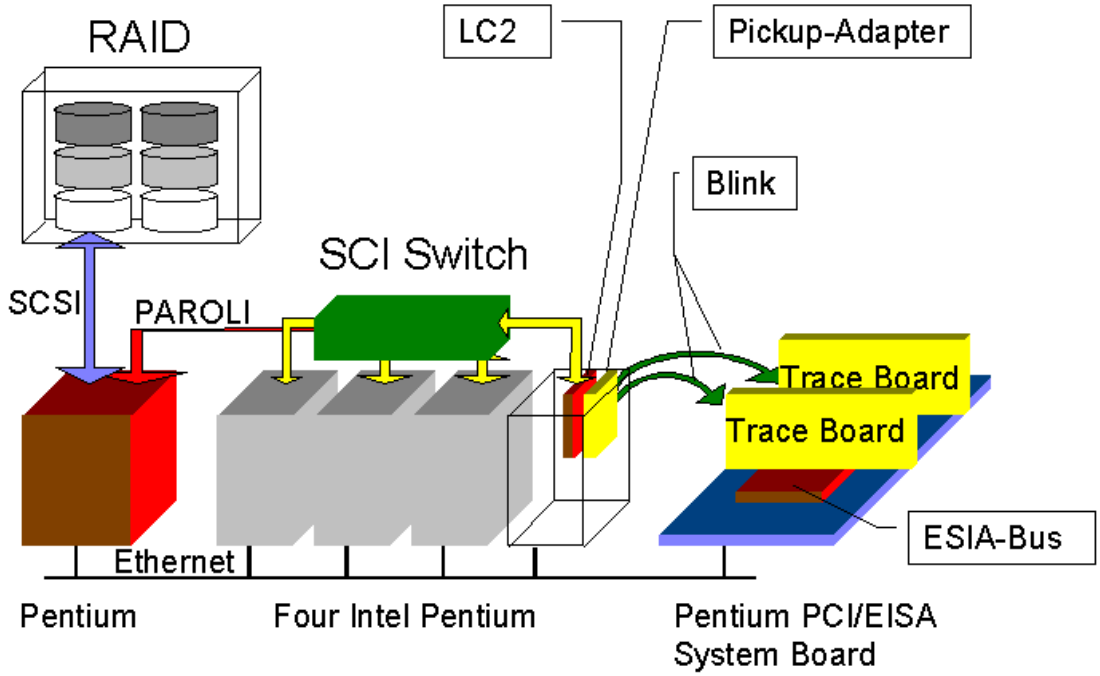


Figure 1: Part of Trinity's Cluster and the SCI Trace Instrument (DT200.1) connected via pick-up card to the Dolphin's of the SCI interface card (PSB)

Figure 1 shows the instrument with two trace boards connected to the Dolphin's SCI to PCI interface card. The Blink™ carries the encapsulated SCI packets on a 64 bit wide data-bus.

The following Figure 2 demonstrates the distribution of the 64 bit wide Blink™ data onto the two trace-boards.

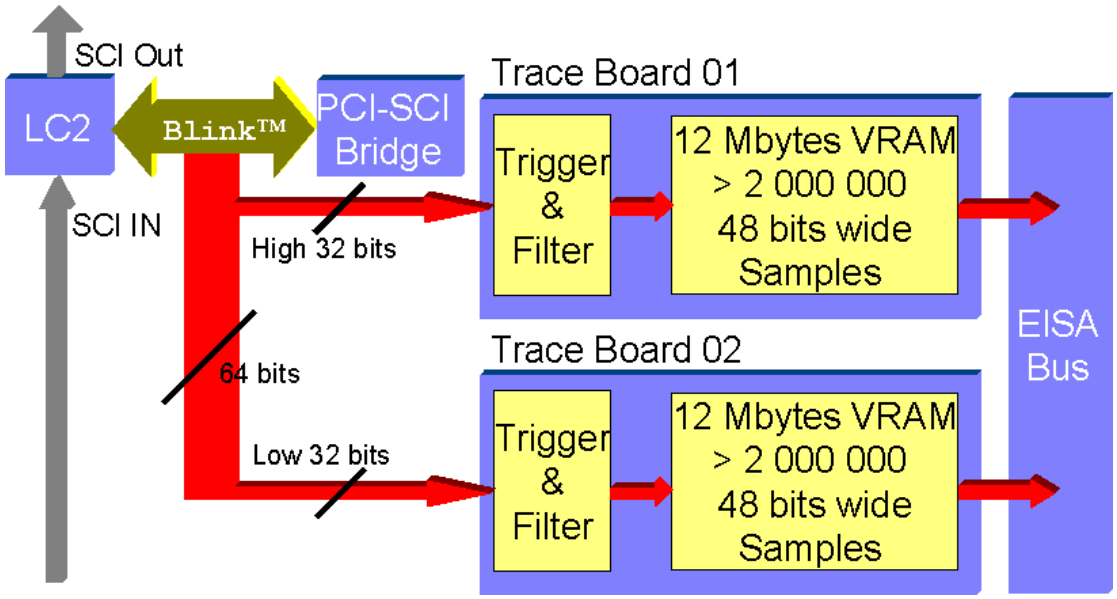


Figure 2: Data flow from Blink™ onto the Trace Boards

The system configuration shown in Figure 2 allows a data acquisition of more than 2 million

samples, each 96 bits wide. The extra 32 bits per sample above and beyond the width of the Blink™ is used for additional Blink™ information and decoding information.

1.4 Trace Trigger

The trace hardware provides a triggering facility. Both boards can be interconnected for trigger synchronisation. This enables triggering over the full width of the two boards. Further the instrument may be connected to a second instrument for a synchronised SCI data acquisition. Therefore it is possible to collect data from two Blinks™ (two nodes) in a synchronised fashion and the two traces may be correlated in a subsequent SCI data analysis (see Section 7) Trace Data Retrieval and Analysis.

.

The trigger mechanism provides four level triggering. Alternatively a trigger level may be used for trace filtering.

1.5 Trace Filter

The trace hardware provides also a filtering facility. Similar to the trigger implementation the boards may be interconnected for filter synchronisation. This enables filtering over the full width of the two boards.

1.6 Trace System software

Figure 3 shows how the specific software components relate to the hardware.

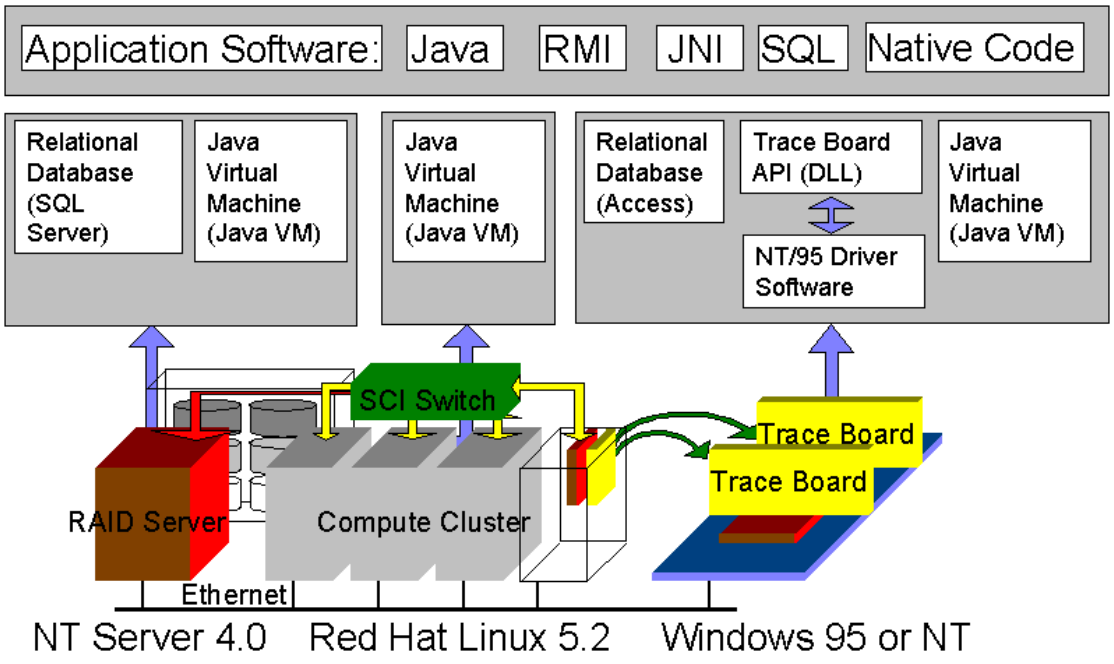


Figure 3: Software on Trinity's Cluster and the SCI Trace Instrument (DT200.1)

1.7 SCI Trace Instrument Driver

Windows 95 and Windows NT Trace board driver has been developed. The drivers provide the underlying mechanism for the Application Programming Interface (API).

1.8 SCI Trace Instrument API

A trace board API has been developed that allows all the required trace hardware manipulations (e.g. trace data retrieval through the EISA instrument bus, trigger definition).

1.9 SCI Trace Instrument User Control Interface

This GUI uses the SCI trace instrument API to provide the user with an interface to control the trace hardware.

1.10 SCI Trace database

The following figure shows how trace data are retrieved from the trace hardware. These trace data are subsequently decoded and imported into the trace database. The trace database may be local or remote. The depicted database tables are only a subset of the tables required to accommodate all packet types (see section 3 for SCI Packet Specification and section 4 for the Trace Database design).

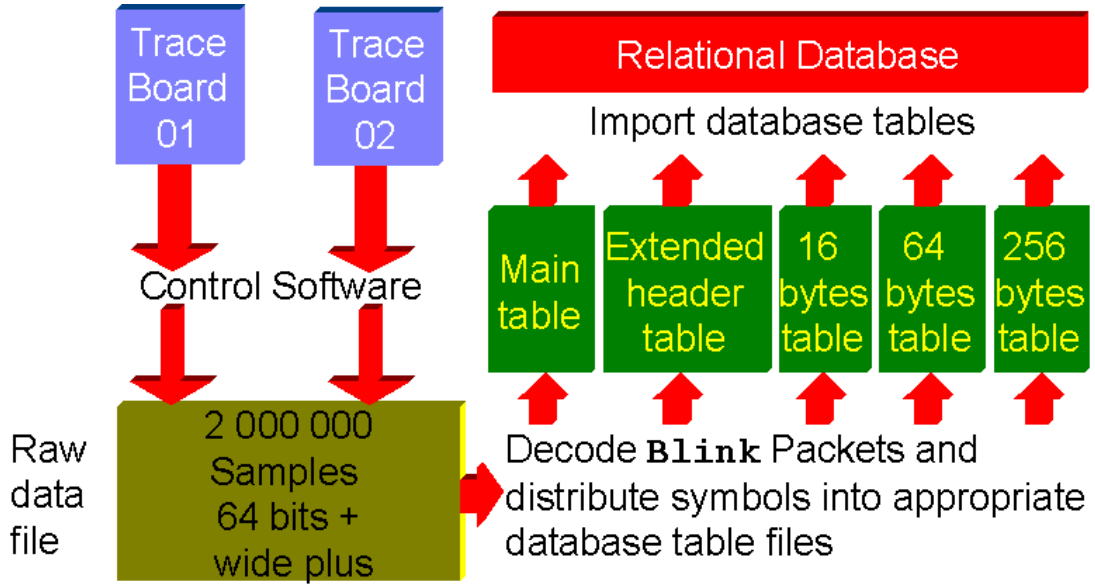


Figure 4: Trace data decoding and database import

1.11 Trace data analysis

The Trace data analysis is performed through SQL queries into the trace-databases. All the application software is implemented in Java. JNI is used to interface the Java-code with the native trace hardware API.

2. Trace Instrument Control Software

2.1 Traceboard Windows 95 Virtual Device Driver (VxD)

Since the board uses both I/O (for the registers) and normal memory space (for the VRAM and SRAM), the most efficient implementation was to use the *IOS.VxD* driver for the I/O space and develop a custom driver for the memory mapping for the VRAM and SRAM.

The memory map driver is called *DTMap01.VxD.* The IOS.VxD driver comes with Windows 95 and is located in the SYSTEM folder of a Windows 95 system.

It was descided to use the *Win32 API* to program the driver since it gives you the possibillity to port the driver to Windows NT. The Win32 API uses control messages to access VxDs. In this case in particular, *DeviceIOControl.*

See ***Appendix A*** for the C source code

2.2 Traceboard API

In a Microsoft Windows 95 or NT environment, device access may be implemented via functions that are loaded when needed from a Dynamic Link Library (DLL), i.e. they are dynamically linked to the application during execution. The DLL then calls device drivers to perform the I/O. A generic DLL can be defined that will serve as an API for both Windows 95 and NT environments; an example is given in the next section. Unfortunately a generic device driver cannot be defined for both Windows 95 and NT, since their requirements are different. Hence these are separately discussed further below.

The appropriate API functions for the DT200.1 are :

- **Reset board**
  ```
  void reset_board(void);
  ```
- **Writes to ADDRMAP register**
  ```
  void set_map(unsigned long address);
  ```
- **Read the current address mapping**
  ```
  unsigned long get_map(void);
  ```
- **Writes to the MODE register**
  ```
  void set_mode(unsigned long flags);
  ```
- **Read the current mode**
  ```
  unsigned long get_mode(void);
  ```
- **Write to the trigger config register**
  ```
  void set_trig_config(unsigned long flags);
  ```
- **Read from the trigger config register**
  ```
  unsigned long get_trig_config(void);
  ```
- **Reset the timestamp counter**

```
void reset_timestamp(void);
```

- **Write to the STOPCOUNT register**
```
void set_stop_count(unsigned long flags);
```

- **Read from the STOPCOUNT register**
```
unsigned long get_stop_count(void);
```

- **Write to the HEADPTR register**
```
void set_head_ptr(unsigned long flags);
```

- **Read from the HEADPTR register**
```
unsigned long get_head_ptr(void);
```

- **Write to the OUTPUTENA register**
```
void set_sbuf_output_enas(unsigned long flags);
```

- **Read from the OUTPUTENA register**
```
unsigned long get_sbuf_output_enas(void);
```

- **Read from the STATUS buffer**
```
unsigned long get_trace_status(void);
```

- **Read from the EISA ID register**
```
unsigned long get_EISA_ID(void);
```

- **Read from EISA ID register amd decode**
```
void get_EISAID(char *text_ID);
```

- **Stop trace**
```
void stop_trace(void);
```

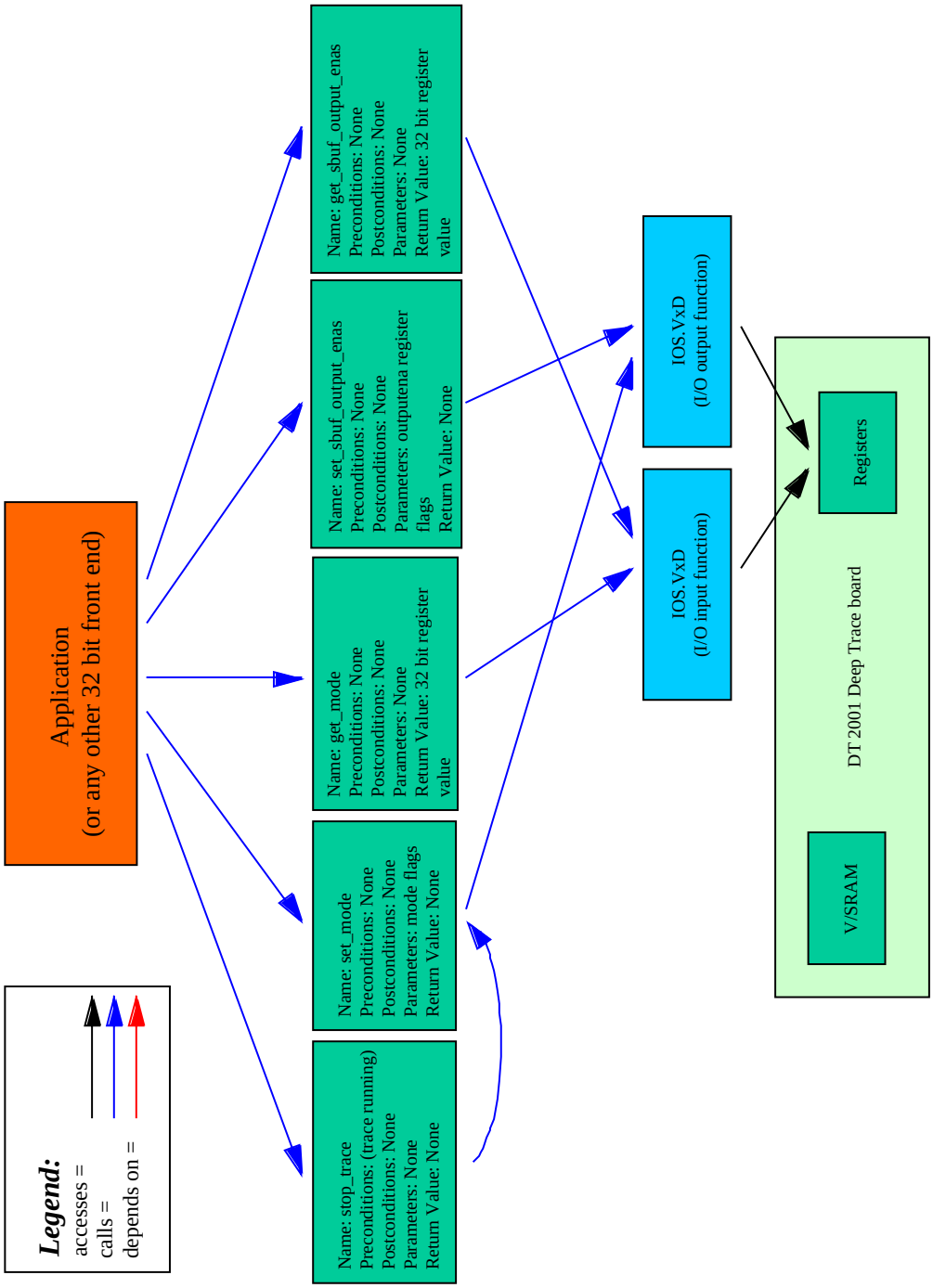
- **Map VRAM and SRAM into linear memory**
```
unsigned long *map_RAM(void);
```

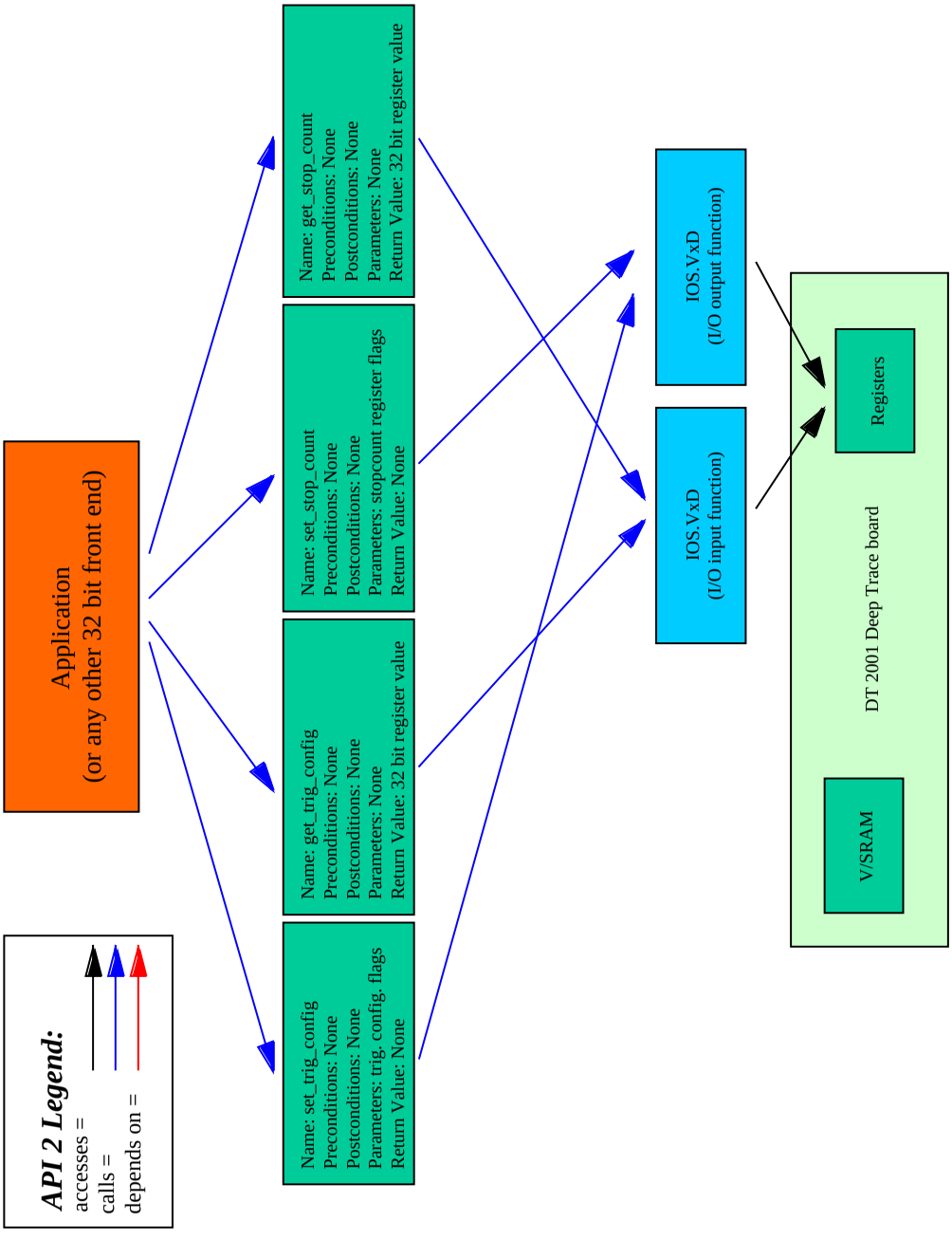
- **Unmap VRAM and SRAM**
```
void unmap_RAM(void);
```

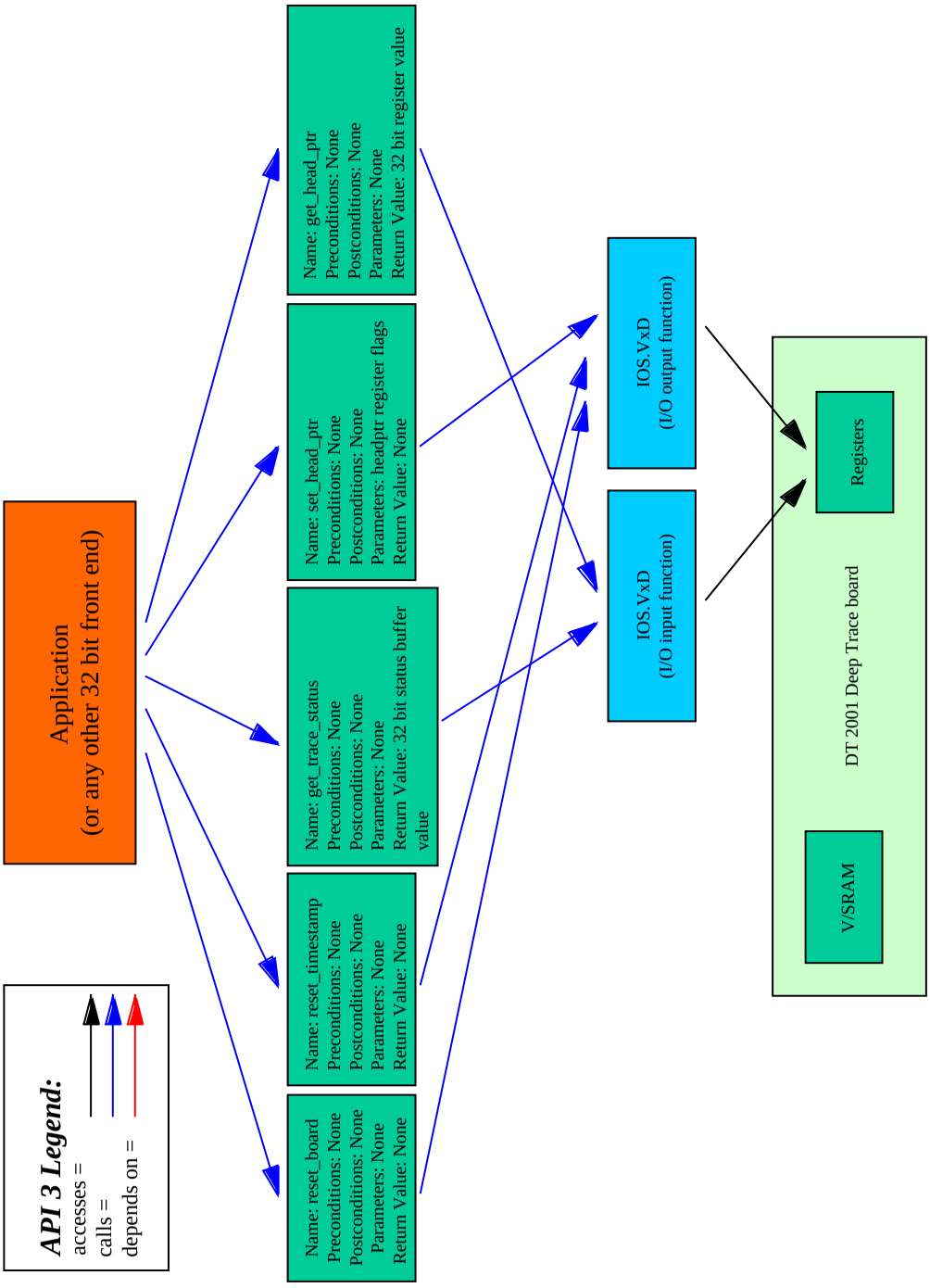
- **read out VRAM or SRAM (start address specified) and store it in array:**
```
BOOL store_RAM(unsigned long *array,
               unsigned long start_address_offset,
               unsigned long size_in_bytes);
```

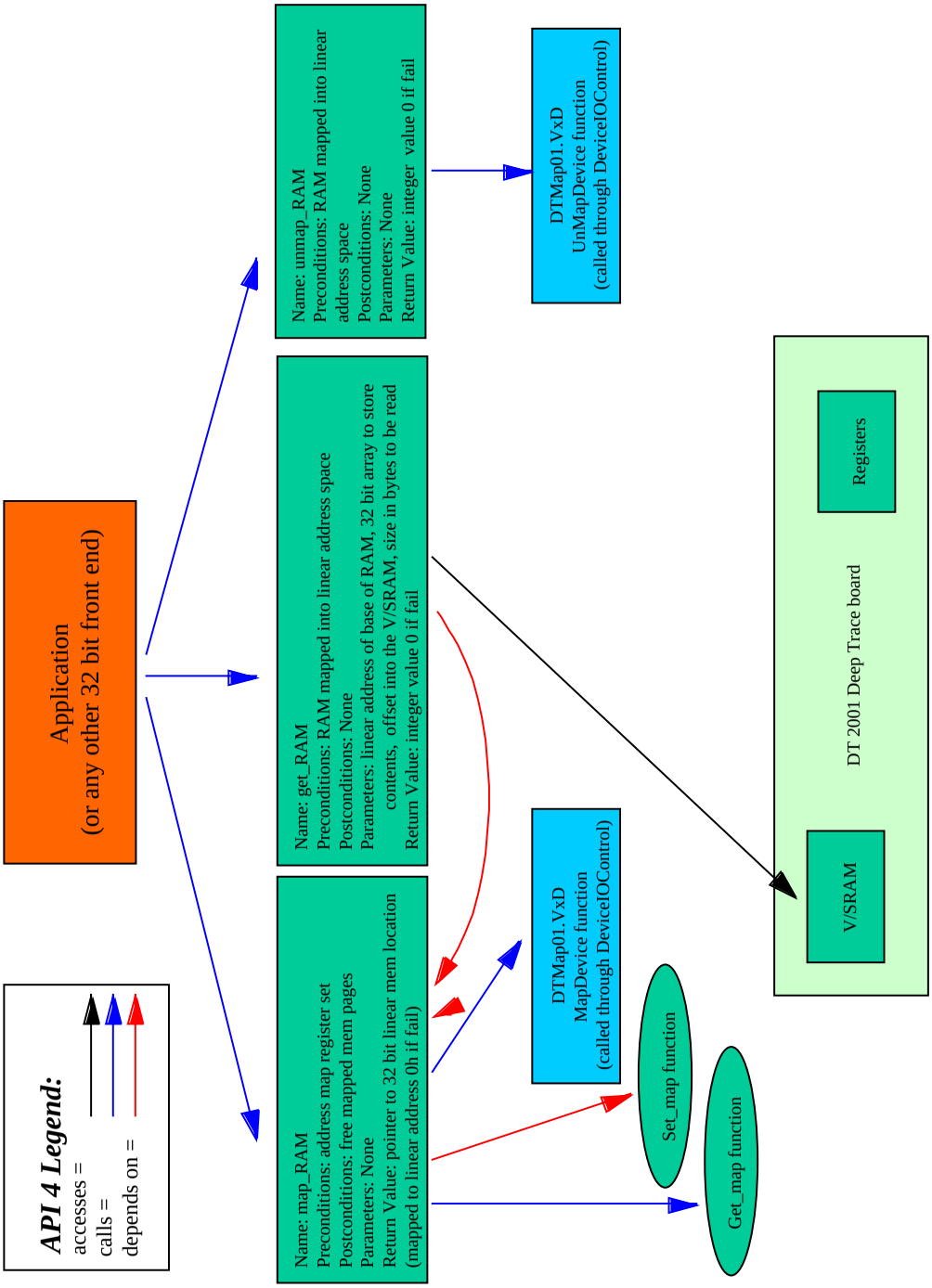
Please see ***Appendix B*** for Traceboard API source code.

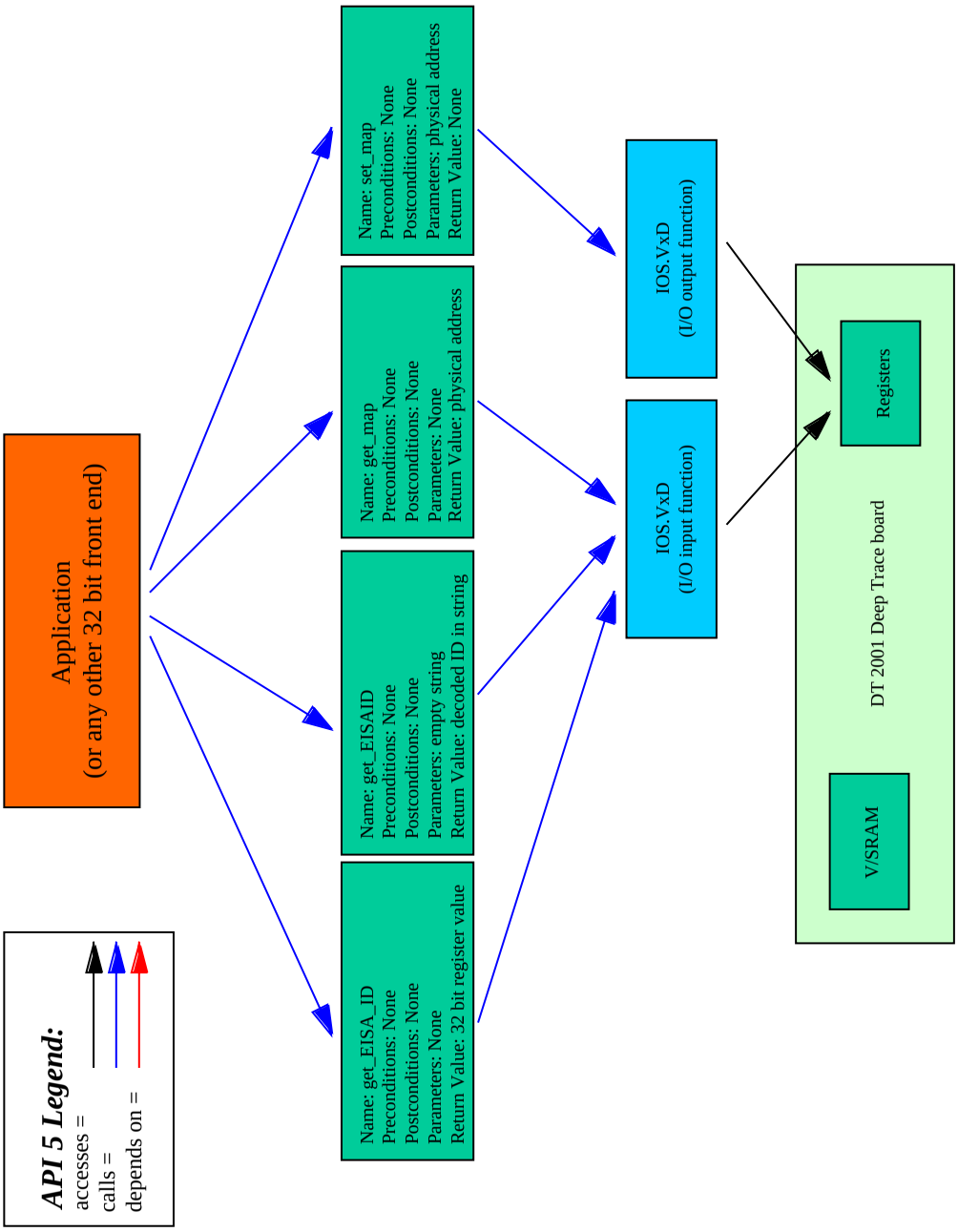
These API functions provide an applications interface suitable for programs written in high level languages like C or Java, such as the example Java application given further again below. The API is illustrated in the following four diagrams :




Application
(or any other 32 bit front end)
Name: set_trig_config
Preconditions: None
Postconditions: None
Parameters: trig. config. flags
Return Value: None
Name: get_trig_config
Preconditions: None
Postconditions: None
Parameters: None
Return Value: 32 bit register value
Name: set_stop_count
Preconditions: None
Postconditions: None
Parameters: stopcount register flags
Return Value: None
Name: get_stop_count
Preconditions: None
Postconditions: None
Parameters: None
Return Value: 32 bit register value
IOS.VxD
(I/O input function)
IOS.VxD
(I/O output function)
DT 2001 Deep Trace board
V/SRAM
Registers
API 2 Legend:
accesses =
calls =
depends on =


Application
(or any other 32 bit front end)
API 3 Legend:
accesses =
calls =
depends on =
Name: reset_board
Preconditions: None
Postconditions: None
Parameters: None
Return Value: None
Name: reset_timestamp
Preconditions: None
Postconditions: None
Parameters: None
Return Value: None
Name: get_trace_status
Preconditions: None
Postconditions: None
Parameters: None
Return Value: 32 bit status buffer value
Name: set_head_ptr
Preconditions: None
Postconditions: None
Parameters: headptr register flags
Return Value: None
Name: get_head_ptr
Preconditions: None
Postconditions: None
Parameters: None
Return Value: 32 bit register value
IOS.VxD
(I/O input function)
IOS.VxD
(I/O output function)
DT 2001 Deep Trace board
V/SRAM
Registers


Application
(or any other 32 bit front end)
Name: map_RAM
Preconditions: address map register set
Postconditions: free mapped mem pages
Parameters: None
Return Value: pointer to 32 bit linear mem location
(mapped to linear address 0h if fail)
Name: get_RAM
Preconditions: RAM mapped into linear address space
Postconditions: None
Parameters: linear address of base of RAM, 32 bit array to store
contents, offset into the V/SRAM, size in bytes to be read
Return Value: integer value 0 if fail
Name: unmap_RAM
Preconditions: RAM mapped into linear
address space
Postconditions: None
Parameters: None
Return Value: integer value 0 if fail
DTMap01.VxD
MapDevice function
(called through DeviceIOControl)
DTMap01.VxD
UnMapDevice function
(called through DeviceIOControl)
Set_map function
Get_map function
DT 2001 Deep Trace board
V/SRAM
Registers
API 4 Legend:
accesses =
calls =
depends on =


Application
(or any other 32 bit front end)
Name: get_EISA_ID
Preconditions: None
Postconditions: None
Parameters: None
Return Value: 32 bit register value
Name: get_EISAID
Preconditions: None
Postconditions: None
Parameters: empty string
Return Value: decoded ID in string
Name: get_map
Preconditions: None
Postconditions: None
Parameters: None
Return Value: physical address
Name: set_map
Preconditions: None
Postconditions: None
Parameters: physical address
Return Value: None
IOS.VxD
(I/O input function)
IOS.VxD
(I/O output function)
DT 2001 Deep Trace board
Registers
V/SRAM
API 5 Legend:
accesses =
calls =
depends on =

2.3 Trace Data Export

The traceboard API allows to write the trace memory contents into files in the trace instrument. These files are subsequently decoded and import into the trace database.

2.4 Traceboard Control GUI

The current Graphical User Interface (GUI) is a Visual Basic application that invokes the traceboard Application Programming Interface (API). The application will further be implemented in Java to web-enable the application for remote control. This port includes the use of Java's Remote Method Invocation (RMI) and Java's Native Interface (JNI) to call the traceboard API from Java.

Please see ***Appendix C*** for Visual Basic GUI source code.

This interface allows the instrument operator to control the traceboard hardware.
The design specifications allow the user to :

(a) Read from and write to registers

(b) Read from VRAM

(c) Read from and write to SRAM

(d) Perform various board test procedures (including repeated stress testing)

(e) Save the VRAM contents to a file

(f) Analyse VRAM data to check board functionality and log results to a file

The interface consists of four main parts, a Register Window (Figure 5), a VRAM listing window (Figure 6), a SRAM control window (Figure 7) and Trace Memory Verification window (Figure 8).

2.4.1 Register Interface

The register window displays the contents of all of the registers. The *Address Map*, *Stop Count*, *Head Pointer* and *EISA ID* registers are displayed as text boxes containing hexadecimal data. The rest of the registers are displayed as check boxes indicating individual bit values. The *TIMECFG* time divisor is displayed in a drop down list with possible values of 20, 40, 80 and 160 ns.

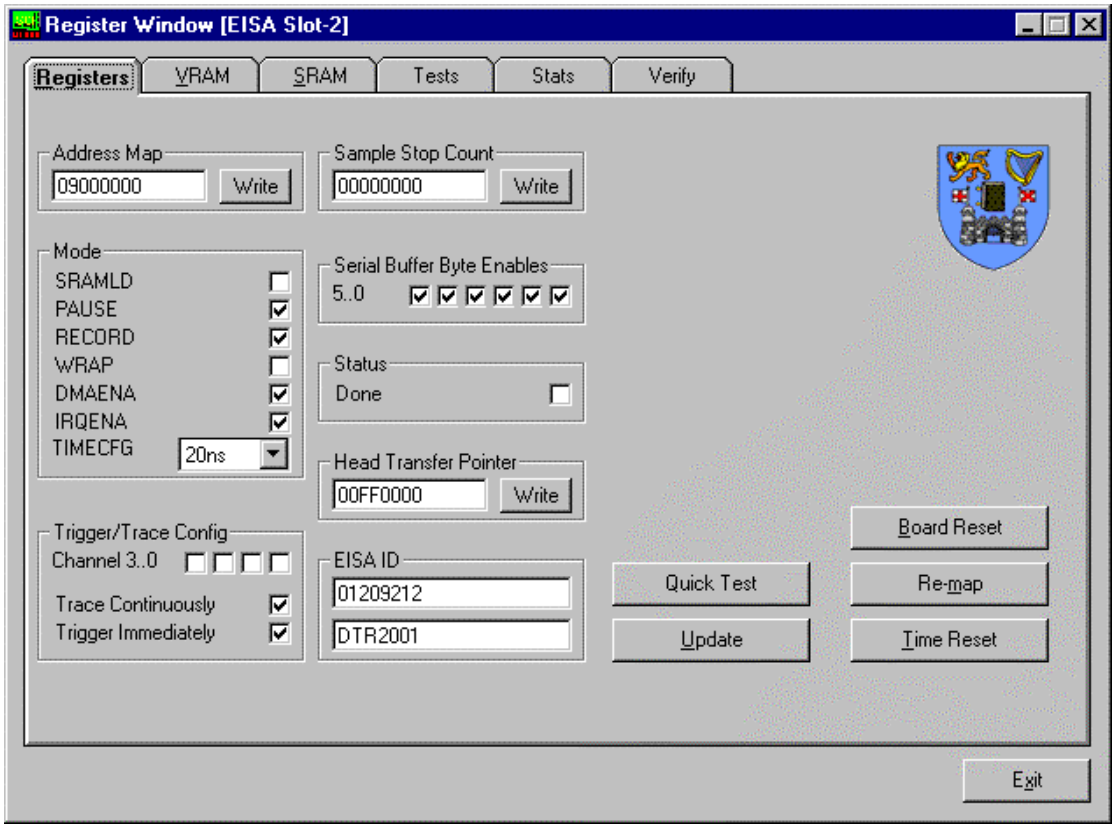


Figure 5: Register Interface

The *Mode*, *Trigger/Trace Config* and *Serial Buffer Byte Enables* are automatically written to whenever the check boxes are modified; the *Address Map*, *Stop Count* and *Head Pointer* registers must be explicitly written to by clicking the relevant Write buttons. If the address map is changed then a *Re-map* should be invoked directly afterwards.

A *Board Reset* proceeds as defined in the DT200.1 Technical Manual. A *Time Reset* simply clears the timestamp counter. The current values of the registers are refreshed when *Update* is clicked.

Invoking *Quick Test* starts a pre-programmed acquisition of samples, tracing continuously and immediately triggering the down-counting of the stop counter. It takes about three seconds for initialization of this process – after that the *Quick Test* button changes to *Running*. Clicking the *Running* button immediately stops the test. The results may then be viewed with the *VRAM* tab (the SCI Trace Data Interface).

2.4.2 SCI Trace Data Interface

In the *VRAM* listing window, the contents of a range of VRAM addresses can be displayed in a VRAM list box by entering the start and end addresses in text boxes and clicking on the *Update* button. For example, entering hexadecimal 0 in the *From* box and hexadecimal 1000 in the *To* box will display the contents of the VRAM from address 0 to 1000 in the list box, one longword per row.

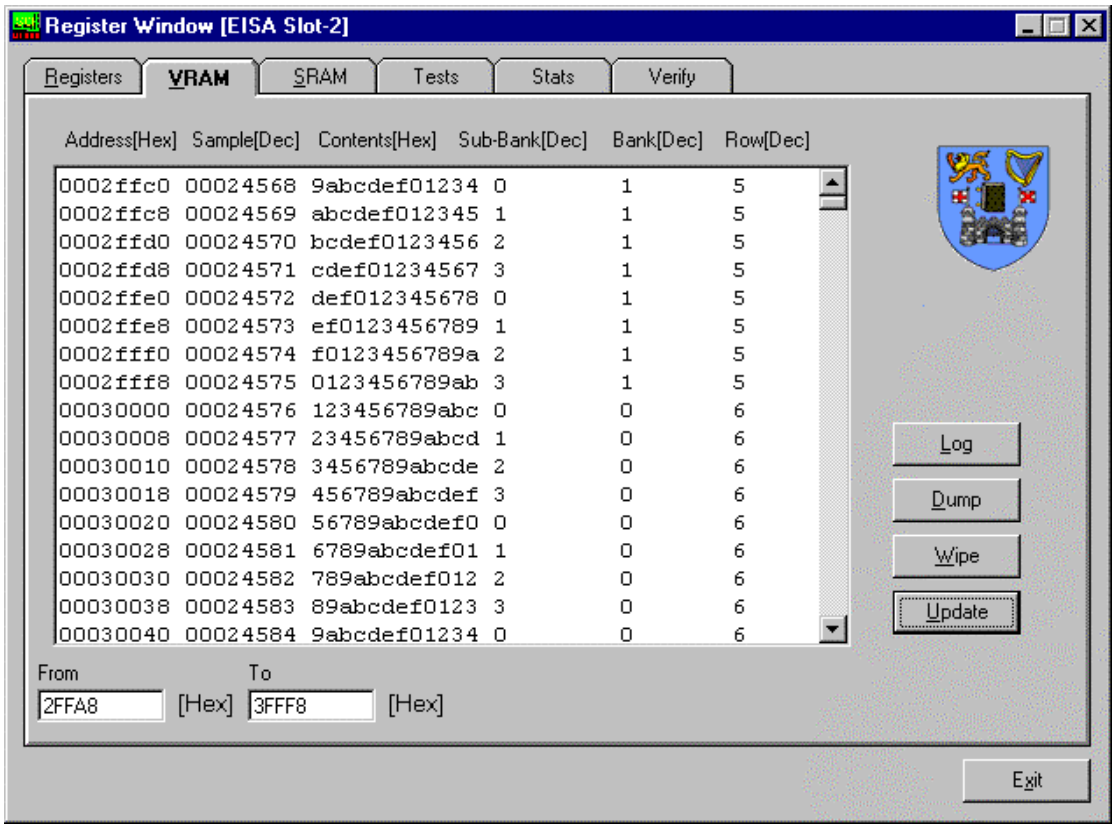


Figure 6: SCI Trace Data Interface

Wipe clears the VRAM contents. *Dump* saves the contents to a binary file, whereas *Log* saves the contents to a text file.

2.4.3 SCI Trigger & Filter Interface

The SRAM window enables the operator to define trigger and filter pattern.

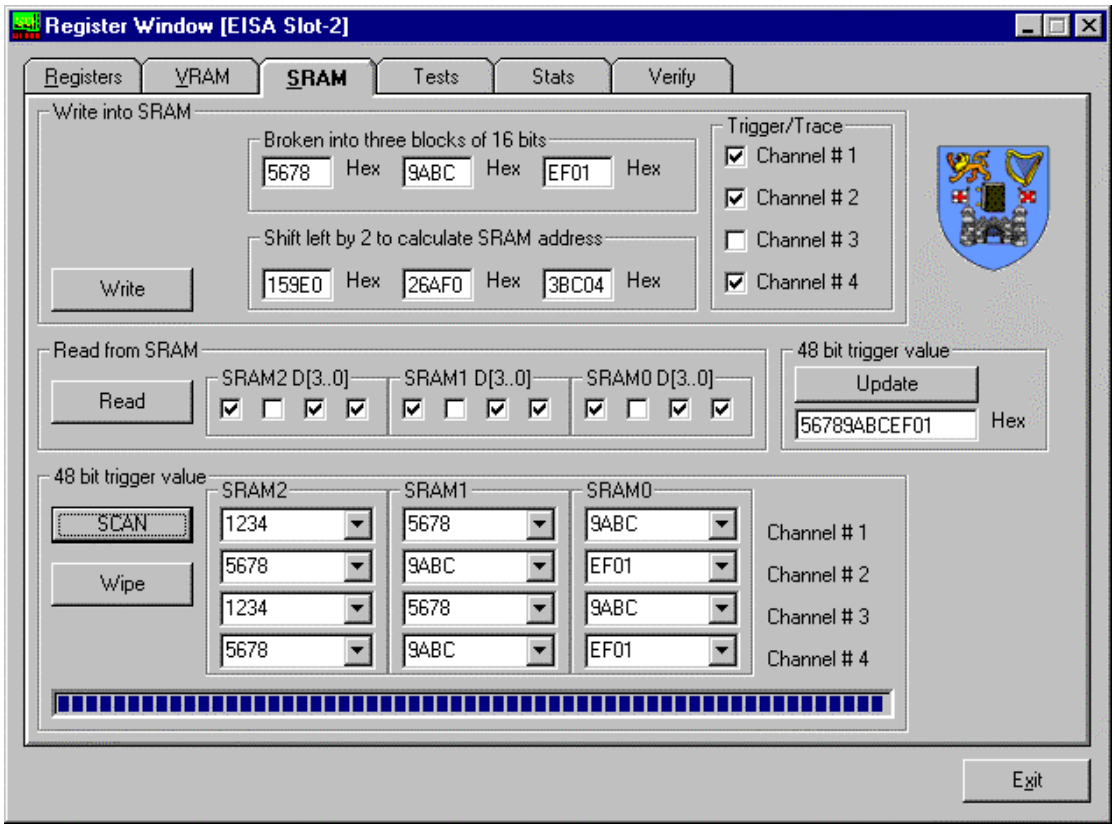


Figure 7: SCI Trigger & Filter Interface

This interface is necessary for co-ordinated creation of the trigger/trace SRAM data patterns that define the way the tracer acquires trace data and is triggered to begin counting down the stop count. In general these events are not defined by a single SRAM word, but a range of words, sometimes encompassing the entire SRAM, and so it would be very tedious to do this manually.

Check boxes indicate the current state of the trigger/trace flags that signify whether a channel is for acquiring filtered trace samples or for triggering the counting down of the stop count.

Each 48bit sample field naturally breaks into 16bit symbols. If a 48bit trigger value is entered into the text box, and *Update* clicked, it will be broken down into its symbols as shown, shifted left by 2 bits to calculate the SRAM address. Clicking *Write* will store it in the SRAM.

Clicking *Scan* starts a process which interrogates all locations of the SRAM and constructs a table of trigger/trace values, per channel, as drop-down lists. This process takes some time, and its progress is shown on the bar meter beneath the list boxes. *Wipe* simply clears the SRAM contents.

The panel shown here only controls a single trace board, with a 48bit sample width; this will be extended to cover a 96bit sample field for two trace boards.

2.4.4 Trace Board Memory Test

This interface controls a repeated stress test of the traceboard memory and provides feedback to the operator. A test signal generator is used to produce electrical test pattern at the input of the traceboard. The control software ensures that the entire trace memory has been written with test signals. The software subsequently verifies the correctness of the sampled test data. The procedure is repeated after completion until terminated by the operator. Test results are committed to a log file.

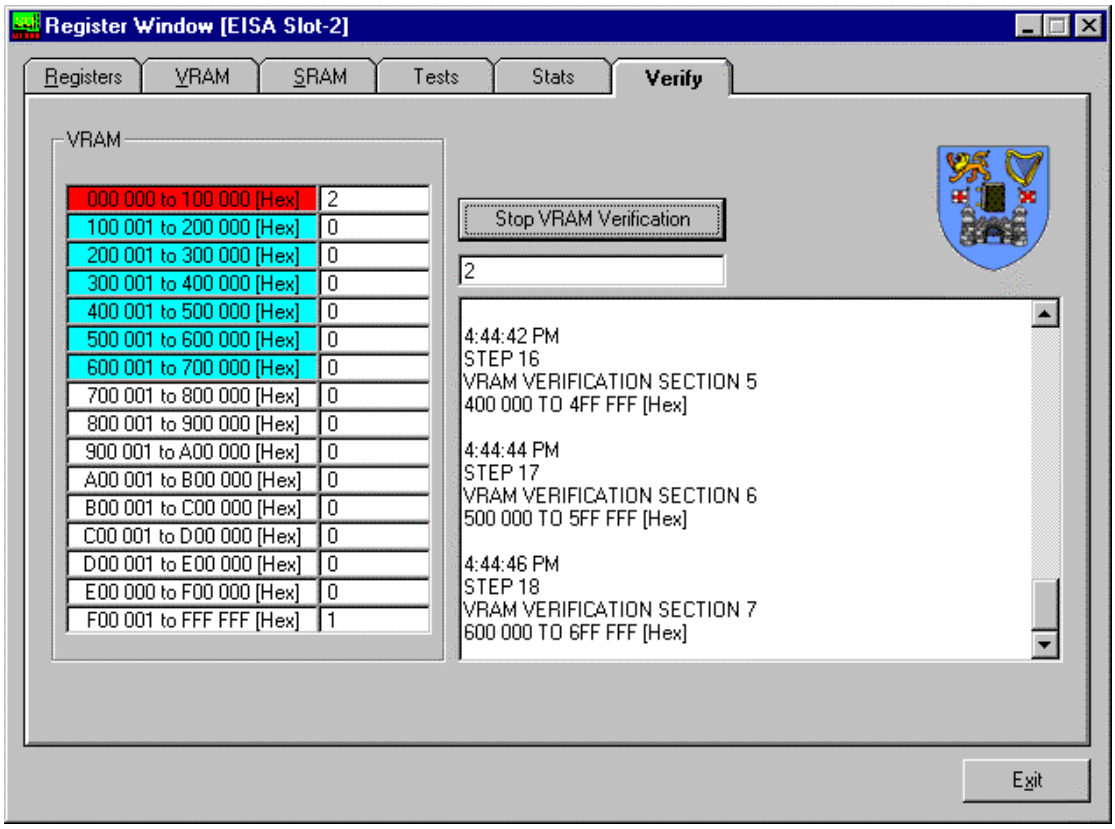


Figure 8: Trace Board Memory Test

The test Signal generator consists of a MACH445 programmable logic device mounted on a small PCB that plugs into the trace board. Although in theory it can be programmed to generate arbitrary patterns, for each such pattern there must be a corresponding software module that can recognize the pattern and any defects in it. Thus far only a few such patterns have been defined, to count in 4, 8 or 48 bit widths, with a continuous or psuedo-random clock.

3 SCI Packet Specification

The following section categorises the SCI packets relevant to the trace data analysis software.

The trace board analysis software and the trace memory boards are design to operate with input from Blink snoop probes or SCITRAC SCI link tracers. SCITRAC instrument snoops on 500 Mbyte/s LC2 LVDS links. Therefore a decoding for the following packet types is required:

3.1 SCI Packets (SCITRAC SCI link tracers)

- Request-send-packet with extended header and 0 bytes data
- Request-send-packet with extended header and 16 bytes data
- Request-send-packet with extended header and 64 bytes data
- Request-send-packet with extended header and 256 bytes data
- Request-send-packet with 0 bytes data
- Request-send-packet with 16 bytes data
- Request-send-packet with 64 bytes data
- Request-send-packet with 256 bytes data

- Request-echo-packet

- Response-send-packet with extended header and 0 bytes data
- Response-send-packet with extended header and 16 bytes data
- Response-send-packet with extended header and 64 bytes data
- Response-send-packet with extended header and 256 bytes data
- Response-send-packet with 0 bytes data
- Response-send-packet with 16 bytes data
- Response-send-packet with 64 bytes data
- Response-send-packet with 256 bytes data

- Response-echo-packet

- Idle Symbols
- Sync packets

3.2 Blink Encapsulated Packets (Blink snoop probes)

- Encapsulated request-send-packet with extended header and 0 bytes data
- Encapsulated request-send-packet with extended header and 16 bytes data
- Encapsulated request-send-packet with extended header and 64 bytes data
- Encapsulated request-send-packet with extended header and 256 bytes data
- Encapsulated request-send-packet with 0 bytes data
- Encapsulated request-send-packet with 16 bytes data
- Encapsulated request-send-packet with 64 bytes data
- Encapsulated request-send-packet with 256 bytes data

- Encapsulated response-send-packet with extended header and 0 bytes data
- Encapsulated response-send-packet with extended header and 16 bytes data
- Encapsulated response-send-packet with extended header and 64 bytes data
- Encapsulated response-send-packet with extended header and 256 bytes data
- Encapsulated response-send-packet with 0 bytes data
- Encapsulated response-send-packet with 16 bytes data
- Encapsulated response-send-packet with 64 bytes data
- Encapsulated response-send-packet with 256 bytes data

4. Trace Database

A TraceId and a PacketId uniquely identify every SCI packet in every trace. Every trace-data-table contains these two Ids as Primary Keys. The SCI_Packet table is the only trace-data-table commen to all packets and contains a Packet_Type_Id. An initial query to this table provides the identification of the packet type and a selection of the subsequent SQL query as defined in sections ***Appendix D: SCI Packet Trace Database Distribution***.

These 36 SQL Querys constitude the bases for more advanced Querys.

4.1 Database Table Definition

The database is divided into a number of tables to accommodate the different types of SCI packets. Please see (section 3) for the SCI Packet specification.

Sections 4.1.1 to 4.1.14 specify the 14 trace database tables required to store the trace data. .

4.1.1 SCI_Packet

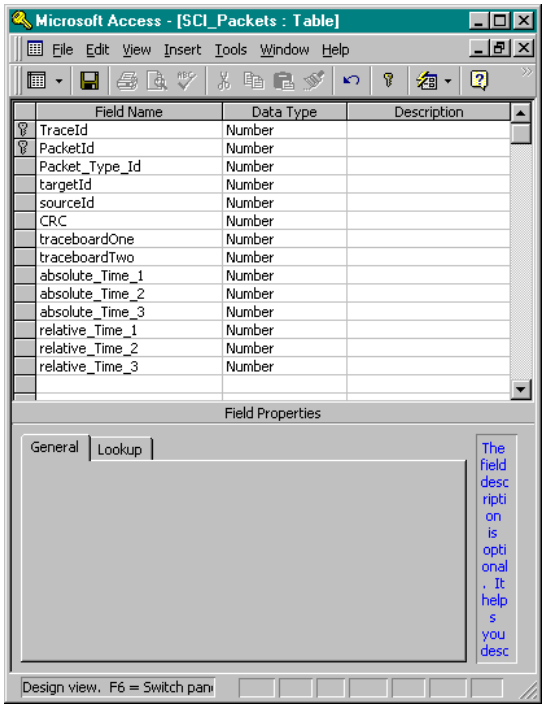


Figure 9: SCI_Packets:Table – this comprises symbols common to all types of SCI packets

4.1.2 SCI_AddressOffset

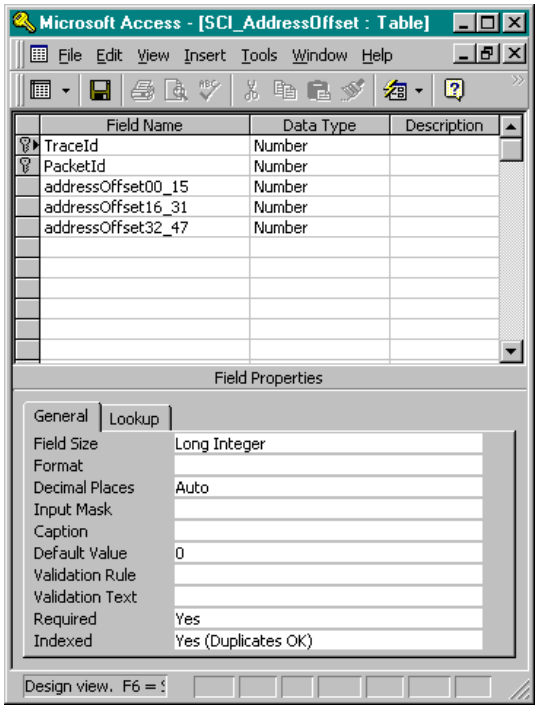


Figure 10: SCI_AddressOffset:Table

4.1.3 SCI_Blink

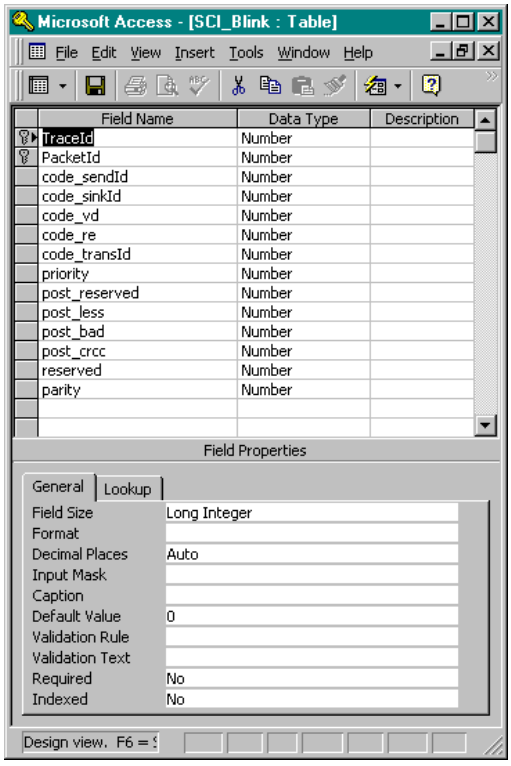


Figure 11: SCI_Blink:Table

4.1.4 SCI_Cmd

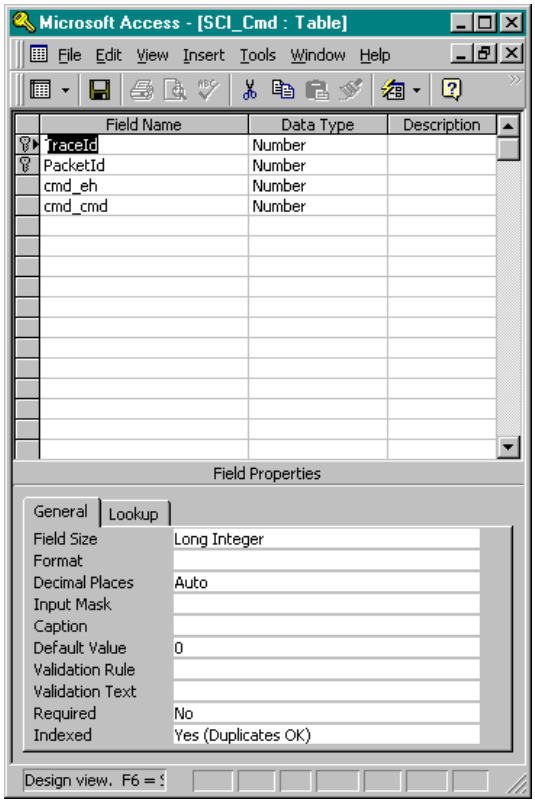


Figure 12: SCI_Cmd:Table

4.1.5 SCI_Cmd_Echo

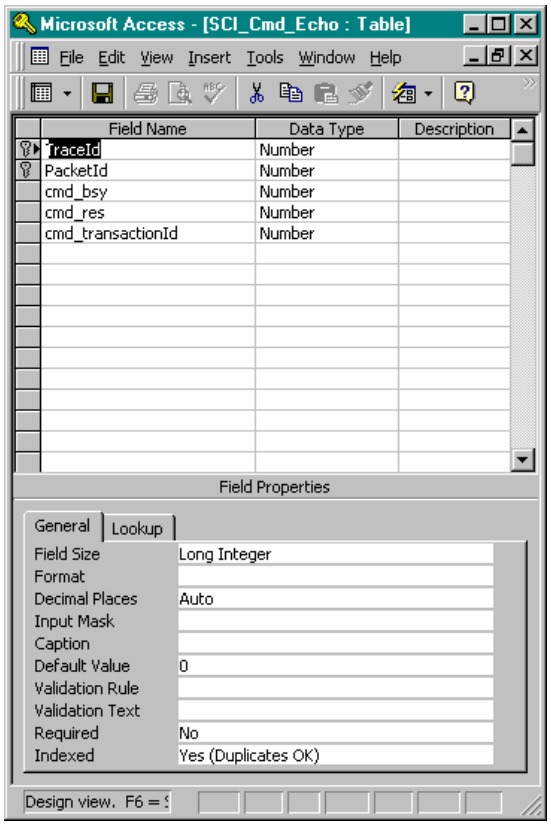


Figure 13: SCI_Cmd_Echo:Table

4.1.7 SCI_Extended

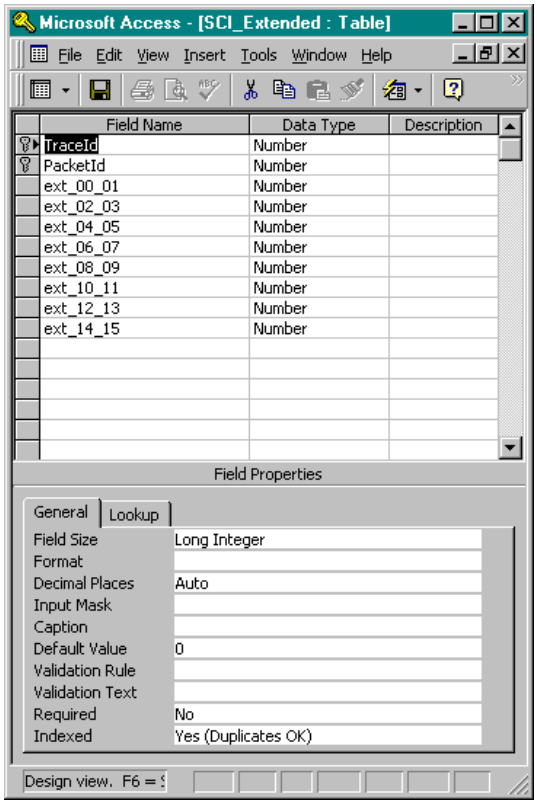


Figure 14: SCI_Extended:Table

4.1.6 SCI_Control

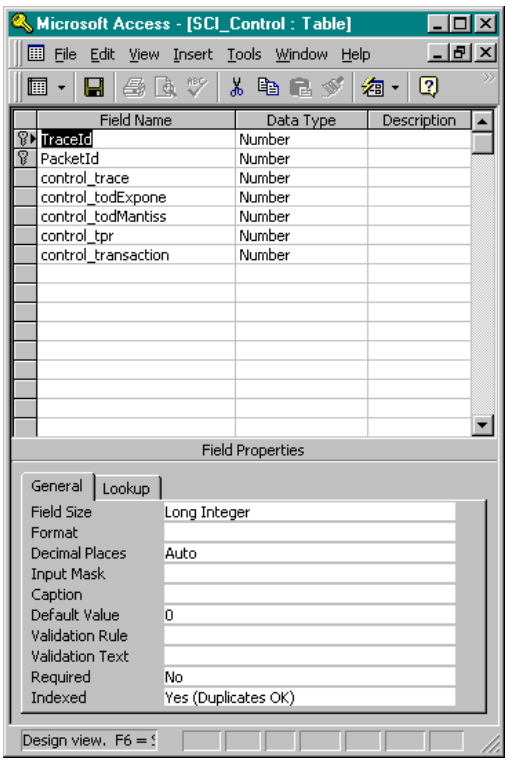


Figure 14: SCI_Control:Table

4.1.8 SCI_FlowControl

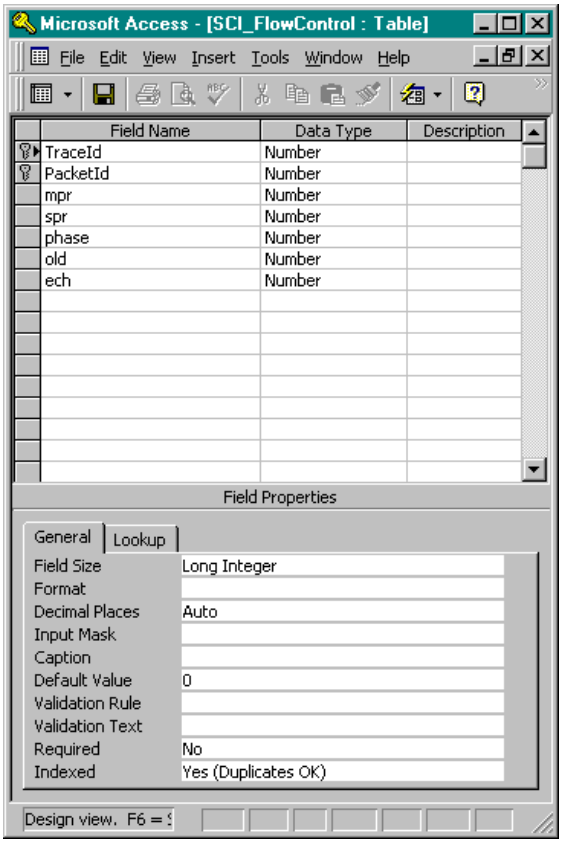


Figure 16: SCI_FlowControl:Table

4.1.9 SCI_Packet_Type_Id

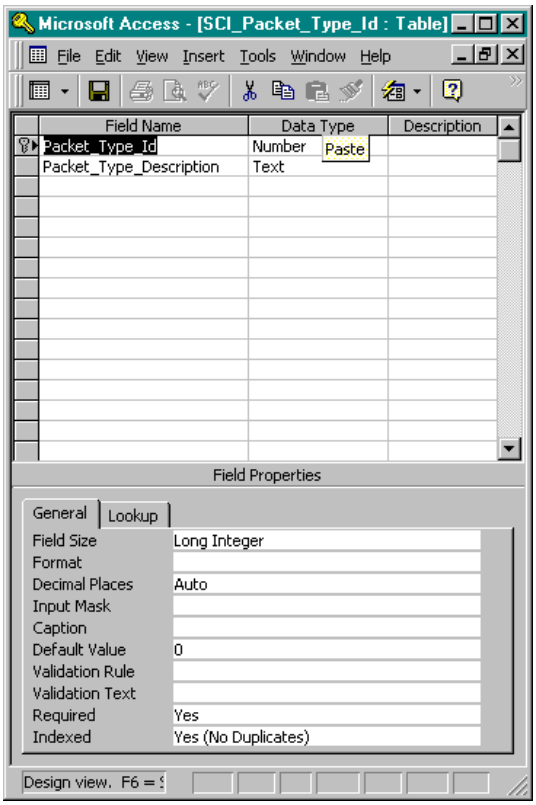


Figure 17: SCI_Packet_Type_Id:Table

4.1.11 SCI_Trace_Information

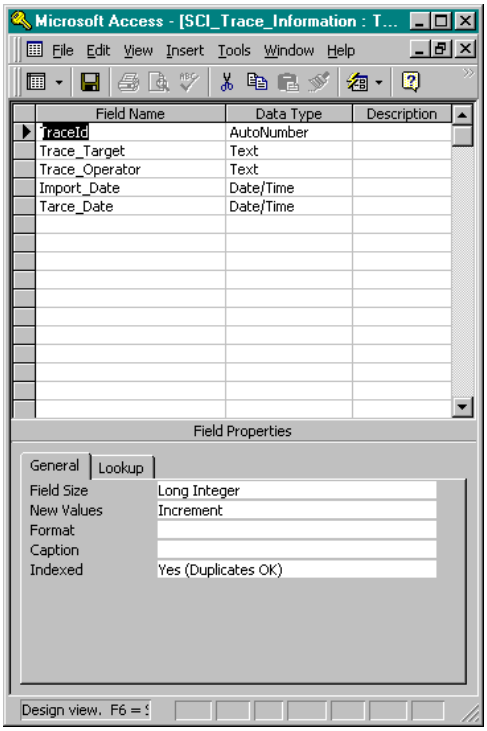


Figure 19: SCI_Trace_Information:Table

4.1.10 SCI_Sta_For_Back

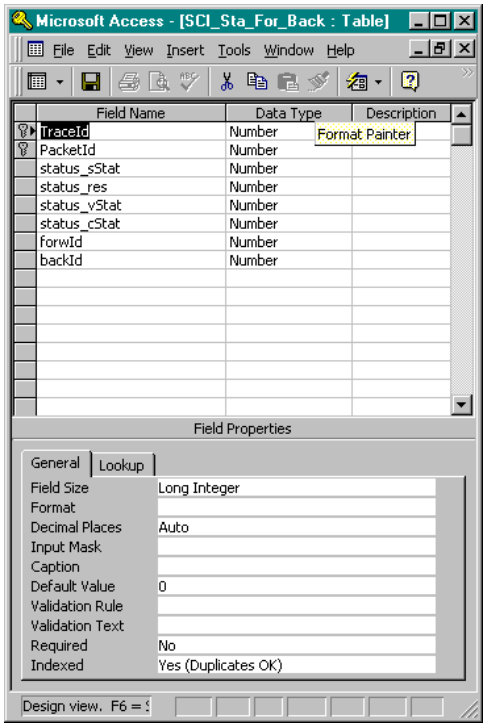


Figure 18: SCI_Sta_For_Back:Table

4.1.12 SCIData_000_015

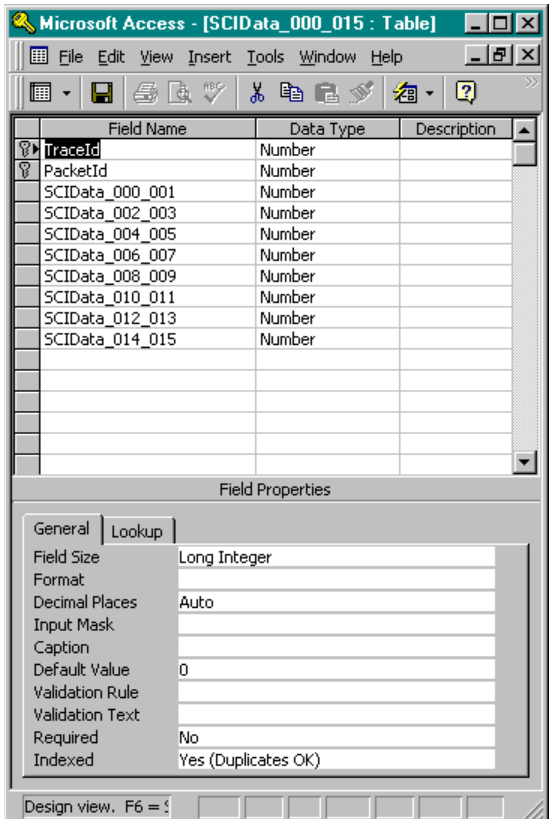


Figure 20: SCIData_000_015:Table

4.1.13 SCIData_016_063

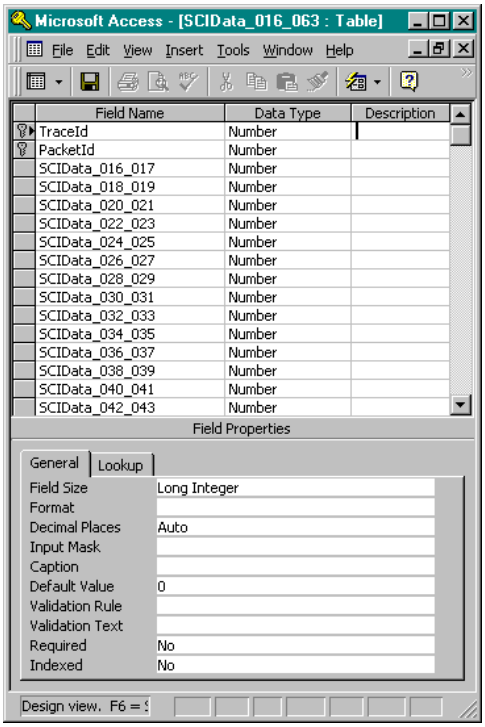


Figure 21: SCIData_016_063:Table

4.1.14 SCIData_064_255

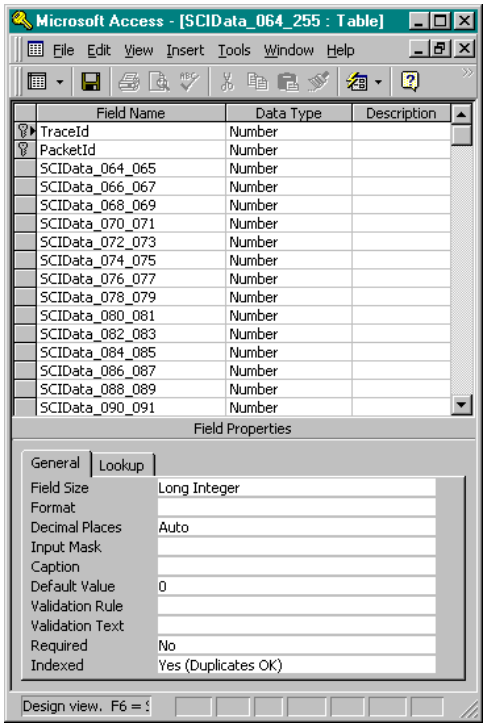


Figure 22: SCIData_064_255:Table

4.2 Trace Data Table Relationship

Figure 22 specifies the relationships amongst the previously defined tables.

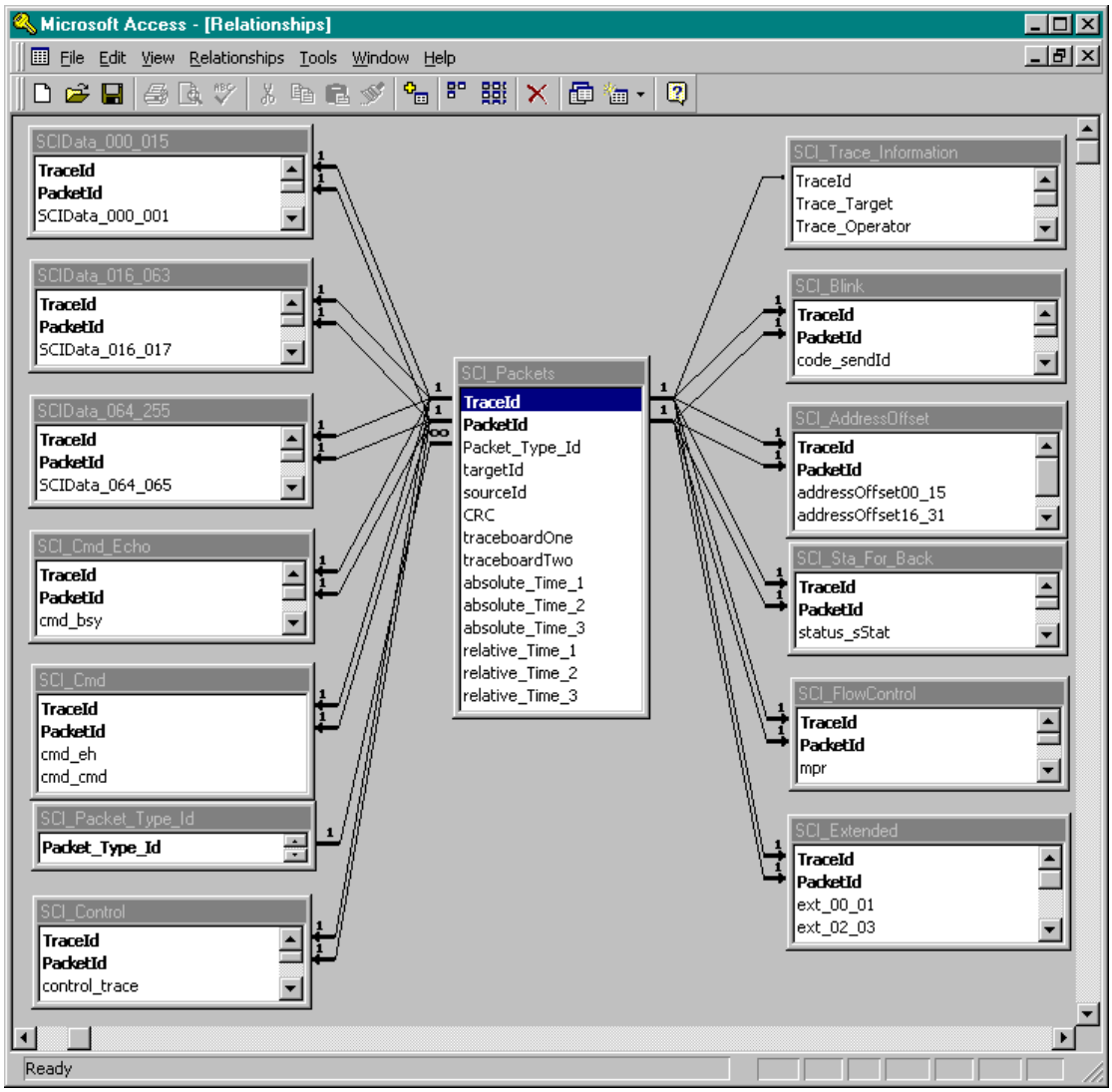


Figure 22: Trace Database Table Relationship

4.3. SCI Packet Table Distribution Including Packet Specific SQL Queries

Appendix D: SCI Packet Trace Database Distribution specifies how the decoding software distributes the SCI packets over the 14 trace database tables. A detailed description is given for each of the 36 packets. Every description in accompanied by a SQL Query for the reconstruction of the decoded SCI packets of this specific type.

This database design allows for space optimised storage and the retrieval of information relavant to a specific packet.

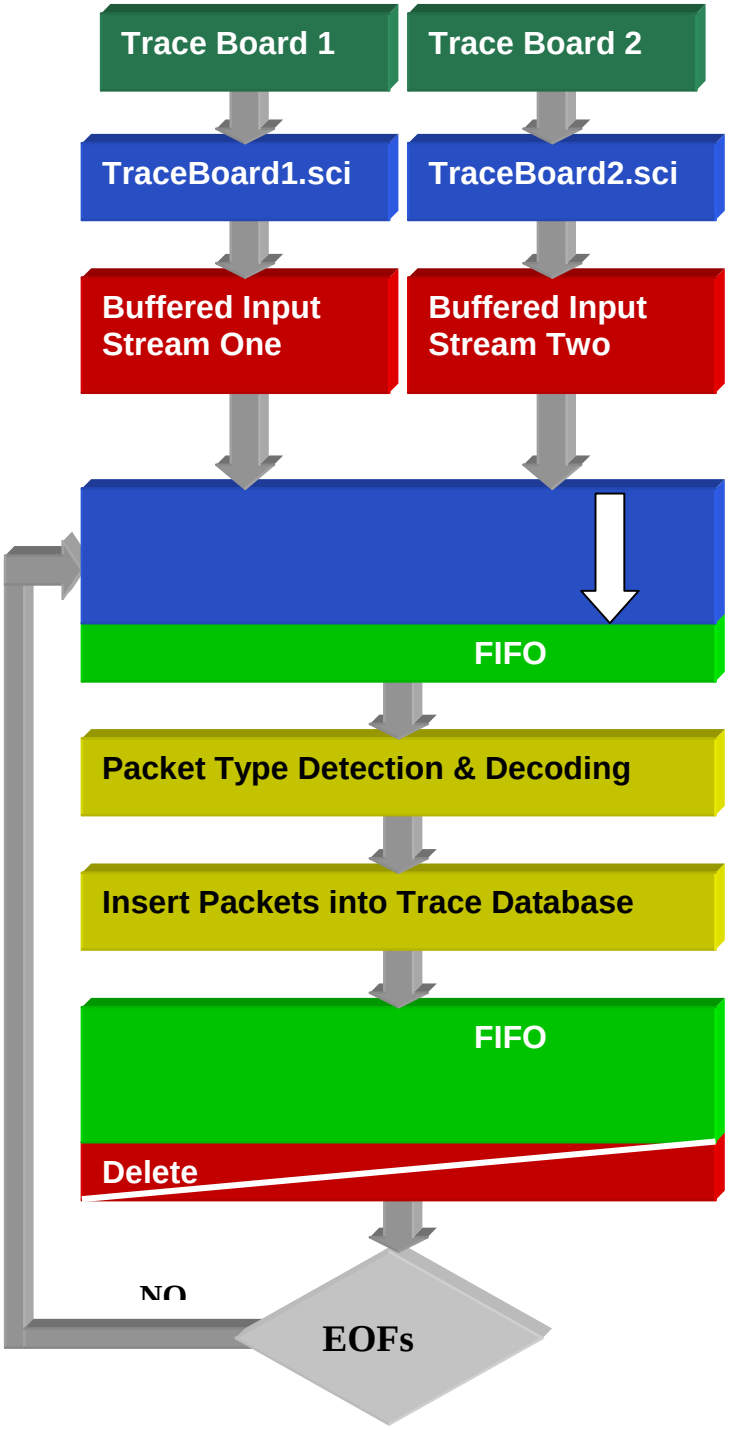
5. Java Client Server Trace Database Connectivity

The trace database server provides trace database access the the trace data decoding and database import application (see section 6) and to the trace data retrieval and analysis software (see section 7).

See ***Appendix E*** for the Java source code

6. Trace Data Decoding and Database Import

See ***Appendix F*** for Java Trace Data Decoding and Database Import source code.



The trace board control software invokes methods on both boards to write trace memory data into files.

The Java decoding application opens buffered input streams to the two trace filesTraceBoard1.sci and TraceBoard2.sci. The software further opens a socket connection to a Trace Database Server. The server can be local or remote.

The Java decoding application reads alternating 6 bytes from input stream one and input stream two and fills the FIFO until the FIFO is full or an EOF has been reached.Trace data previously distributed onto the two trace boards will be reassembled.

The Java decoding application detects the SCI packet type at the top of the FIFO and decodes the packet accordingly. The software creates an object of this specific type including all packet information.

This object is send to the Trace Database server via the previously opened socket connection. The Trace Database server will subsequently invoke a predefined SQL query to insert the SCI Packet Object into the database. The query is specific to the packet type.

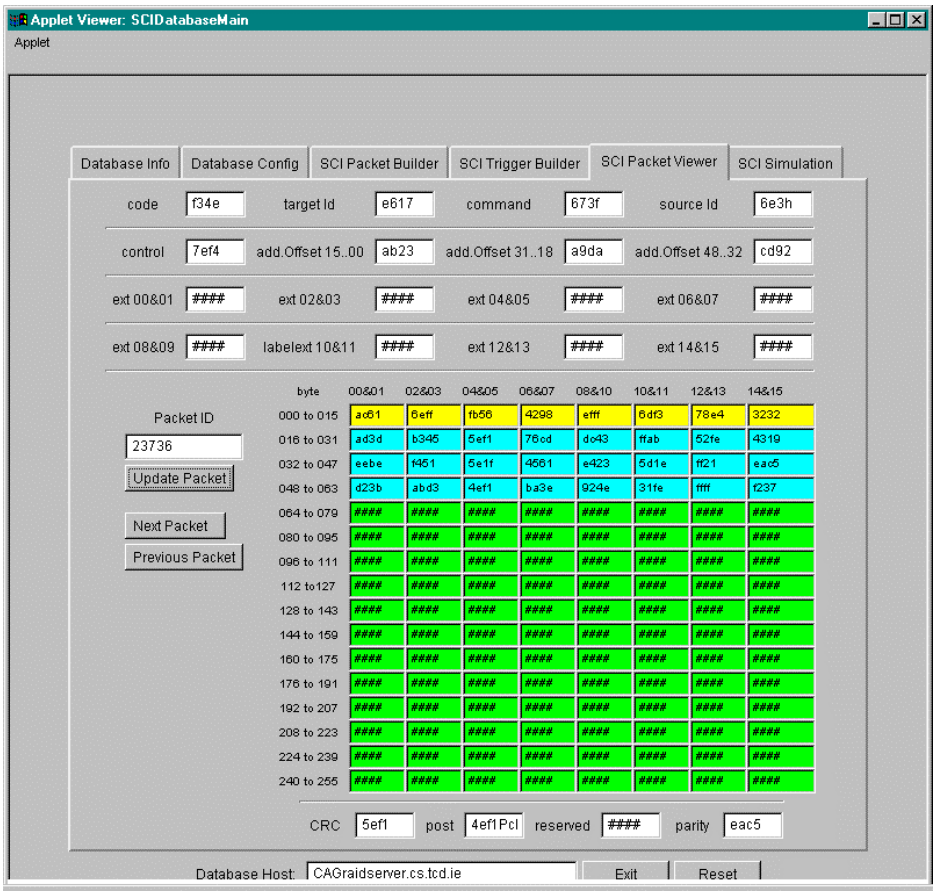
The Java decoding application deletes the detected packet from the top of the FIFO and returns to the FIFO fill method until an EOF has been reached.

7. Trace Data Retrieval and Analysis

The software provides a convenient view of the trace data. The application retrieves data from the trace database.

See ***Appendix G*** for Java Trace Database GUI and Analysis Tool source code.

7.1 Packet Viewer



Appendix A: Windows 95 Virtual Device Driver (VxD)

A generic DLL **definition file** `MapDev.h` for the DT200.1 is as follows :

```
// MAPDEV.h - include file for VxD MAPDEV
// Copyright (c) 1996 Vireo Software, Inc.

#ifndef NotVxD

#include <vtoolsc.h>

#define MAPDEV_Major          1
#define MAPDEV_Minor          0
#define MAPDEV_DeviceID         UNDEFINED_DEVICE_ID
#define MAPDEV_Init_Order     UNDEFINED_INIT_ORDER

#define LPVOID PVOID

#endif

// This is the request structure that applications use
// to request services from the MAPDEV VxD.

typedef struct _MapDevRequest
{
    DWORD  mdr_ServiceID;        // supplied by caller
    LPVOID mdr_PhysicalAddress;  // supplied by caller
    DWORD  mdr_SizeInBytes;      // supplied by caller
    LPVOID mdr_LinearAddress;    // returned by VxD
    WORD   mdr_Selector;         // returned if 16-bit caller
    WORD   mdr_Status;         // MDR_xxxx code below
} MAPDEVREQUEST, *PMAPDEVREQUEST;

#define MDR_SERVICE_MAP       CTL_CODE(FILE_DEVICE_UNKNOWN, 1,
                              METHOD_NEITHER, FILE_ANY_ACCESS)
#define MDR_SERVICE_UNMAP     CTL_CODE(FILE_DEVICE_UNKNOWN, 2,
                              METHOD_NEITHER, FILE_ANY_ACCESS)

#define MDR_STATUS_SUCCESS    1
#define MDR_STATUS_ERROR      0
```

The associated C++ source file MapDev.cpp for the DLL is :

```
#include <windows.h>
#include <winioctl.h>
#include <conio.h>
#include <memory.h>
#define NotVxD
#include "DT2001.h"
#include "DTMap01.h"

/*
// DLL Entry Point disabled since not used
//   Note: The DLL entry point functoin is disables at the
//         moment because Visual Basic doesn't seem to be
//         able to call it; (no loadlib function provided).
BOOL WINAPI DllEntryPoint (HINSTANCE hDLL,
                           DWORD dwReason,
                           LPVOID Reserved)
{
    BOOL Success = TRUE;

    switch (dwReason)
    {
        case DLL_PROCESS_ATTACH:
        {
            //reset_board();
            //set_map(PHY_ADDR);//set physical address of VRAM/SRAM
            //map_RAM();//set linear address of VRAM/SRAM
            break;
        }
        case DLL_PROCESS_DETACH:
        {
            //unmap_RAM();
```

```
            //reset_board();
            break;
        }
        case DLL_THREAD_ATTACH:  break;
        case DLL_THREAD_DETACH:  break;
    }
    return Success;
}
*/

//Global Declarations:
HANDLE hDevice;          //handle for VxD

//resets board
__declspec( dllexport ) void __stdcall
                        reset_board(void)
{
    _outpd(BASE_ADDR | REG_RESET, ANYVALUE);
}

//Writes to ADDRMAP register
//  input: is physical address the VRAM/SRAM is to appear at
__declspec( dllexport ) void __stdcall
                        set_map(unsigned long address)
{
    _outpd(BASE_ADDR | REG_ADDRMAP, address);
}

//Read the current physical address mapping
__declspec( dllexport ) unsigned long __stdcall
                        get_map(void)
{
    return (_inpd(BASE_ADDR | REG_ADDRMAP) & 0xFF000000);
}

//Writes to the MODE register
__declspec( dllexport ) void __stdcall
                        set_mode(unsigned long flags)
{
    _outp(BASE_ADDR | REG_MODE, (int)flags);
}

//Read the current mode
__declspec( dllexport ) unsigned long __stdcall
                        get_mode(void)
{
    return (((unsigned long) _inp(BASE_ADDR | REG_MODE)) &
                                0x000000FF);
}

//Write to the trigger config register
__declspec( dllexport ) void __stdcall
                        set_trig_config(unsigned long flags)
{
    _outp(BASE_ADDR | REG_TRIGCONFIG, (int)flags);
}

//Read from the trigger config register
__declspec( dllexport ) unsigned long __stdcall
                        get_trig_config(void)
{
    return (((unsigned long) _inp(BASE_ADDR | REG_TRIGCONFIG)) &
                                0x0000003F);
}

//Reset the timestamp counter
__declspec( dllexport ) void __stdcall
                        reset_timestamp(void)
{
    _outpd(BASE_ADDR | REG_TIMERESET, ANYVALUE);
}

//Write to the STOPCOUNT register
__declspec( dllexport ) void __stdcall
                        set_stop_count(unsigned long flags)
{
    _outpd(BASE_ADDR | REG_STOPCOUNT, flags);
}

//Read from the STOPCOUNT register
__declspec( dllexport ) unsigned long __stdcall
                        get_stop_count(void)
{
```

```
    return (_inpd(BASE_ADDR | REG_STOPCOUNT) & 0x00FF8000);
}

//Write to the HEADPTR register
__declspec( dllexport ) void __stdcall
                        set_head_ptr(long int head_ptr)
{
    _outpd(BASE_ADDR | REG_HEADPTR, head_ptr);
}

//Read from the HEADPTR register
__declspec( dllexport ) unsigned long __stdcall
                        get_head_ptr(void)
{
    return (_inpd(BASE_ADDR | REG_HEADPTR) & 0x00FF8000);
}

//Write to the OUTPUTENA register
__declspec( dllexport ) void __stdcall
                        set_sbuf_output_enas(unsigned long flags)
{
    _outp(BASE_ADDR | REG_OUTPUTENA, (int)flags);
}

//Read from the OUTPUTENA register
__declspec( dllexport ) unsigned long __stdcall
                        get_sbuf_output_enas(void)
{
    return (((unsigned long)_inp(BASE_ADDR | REG_OUTPUTENA)) &
                                0x0000003F);
}

//Read from the STATUS buffer
__declspec( dllexport ) unsigned long __stdcall
                        get_trace_status(void)
{
    return (_inpd(BASE_ADDR | REG_STATUS) & 0x00008000);
}

//Read from the EISA ID register
__declspec( dllexport ) unsigned long __stdcall
                        get_EISA_ID(void)
{
    return _inpd(BASE_ADDR | REG_EISAID);
}

//Read from EISA ID register amd decode string
//  input: pointer to a char array (min 7 chars long)
__declspec( dllexport ) unsigned long __stdcall
                        get_EISAID(char *text_ID)
{
    unsigned int chars;
    unsigned int digits;
    unsigned long reg = _inpd(BASE_ADDR | REG_EISAID);
    chars = reg;
    digits = reg >> 16;

    text_ID[0] = (char) ( ( chars & 0x007C) >>  2) + 0x40;
    text_ID[1] = (char) ((( chars & 0x0003) <<  3) |
                         (( chars & 0xE000) >> 13))+ 0x40;
    text_ID[2] = (char) ( ( chars & 0x1F00) >>  8) + 0x40;
    text_ID[3] = (char) ( (digits & 0x00F0) >>  4) + '0';
    text_ID[4] = (char) ( (digits & 0x000F) >>  0) + '0';
    text_ID[5] = (char) ( (digits & 0xF000) >> 12) + '0';
    text_ID[6] = (char) ( (digits & 0x0F00) >>  8) + '0';
    text_ID[7] = 0;
    return 1;
}

//Stop trace
__declspec( dllexport ) void __stdcall
                        stop_trace(void)
{
    set_mode(PAUSE);
}

//Map VRAM and SRAM into linear memory, returns a pointer
// to base (32bit)
__declspec( dllexport ) unsigned long __stdcall
                        map_RAM(void)
{
    DWORD cbBytesReturned;//count of bytes returned from VxD
    MAPDEVREQUEST req;    //VxD request structure
```

```
    PVOID inBuf[1];       //buffer for DevIOCrtl pointer
                          // to req structure

    const PCHAR VxDName = "\\\\.\\DTMAP01.VXD";
    const PCHAR VxDNameAlreadyLoaded = "\\\\.\\DTMAP01";


    hDevice = CreateFile(VxDName, 0,0,0,
                         CREATE_NEW, FILE_FLAG_DELETE_ON_CLOSE, 0);
    if (hDevice == INVALID_HANDLE_VALUE)
        hDevice = CreateFile(VxDNameAlreadyLoaded, 0,0,0,
                         CREATE_NEW, FILE_FLAG_DELETE_ON_CLOSE, 0);
    if (hDevice == INVALID_HANDLE_VALUE)
        return (unsigned long) GetLastError();//error!
    else
    {
        //set up request structure:
        req.mdr_ServiceID = MDR_SERVICE_MAP;
        req.mdr_PhysicalAddress = (PVOID) get_map();
        req.mdr_SizeInBytes = VRAM_SIZE;
        inBuf[0] = &req;

        //call Win32 API Message DeviceIOControl:
        if(! DeviceIoControl(hDevice, MDR_SERVICE_MAP, inBuf,
                             sizeof(PVOID), NULL, 0,
                             &cbBytesReturned, NULL))
            return 0;
        else
            return (unsigned long) req.mdr_LinearAddress;//Success!
    }
}

//Unmap VRAM and SRAM
//uses the VxD, frees the linear address space
__declspec ( dllexport ) int __stdcall
                         unmap_RAM(void)
{
    DWORD cbBytesReturned; //count of bytes returned from VxD
    MAPDEVREQUEST req;     //VxD request structure
    PVOID inBuf[1];
    const PCHAR VxDName = "\\\\.\\DTMAP01.VXD";
    const PCHAR VxDNameAlreadyLoaded = "\\\\.\\DTMAP01";
    if (hDevice == INVALID_HANDLE_VALUE)
        hDevice = CreateFile(VxDNameAlreadyLoaded, 0,0,0,
                         CREATE_NEW, FILE_FLAG_DELETE_ON_CLOSE, 0);
    if (hDevice == INVALID_HANDLE_VALUE)
    {
        return 0;//error! can't get a handle on VxD
    }
    else
    {
        //set up request structure:
        req.mdr_ServiceID = MDR_SERVICE_UNMAP;
        req.mdr_PhysicalAddress = (void *) get_map();
        req.mdr_SizeInBytes = VRAM_SIZE;
        inBuf[0]=&req;
        //call Win32 API Message DeviceIOControl:
        if (! DeviceIoControl(hDevice, MDR_SERVICE_UNMAP, inBuf,
                              sizeof(PVOID), NULL, 0,
                              &cbBytesReturned, NULL))
        {
            return 0;//error! call to VxD failed
        }
        else
        {
            return 1; //success!
        }
    }
}

//read out SRAM or VRAM and store in array
//input: array of 32bit storage locations, offset address
//       into the VRAM/SRAM and the #bytes to read out
//output: TURE if success, FALSE if fail
__declspec( dllexport ) int __stdcall
                        get_RAM(PBYTE Map_Address,
                                unsigned long *array,
                              unsigned long start_address_offset,
                                unsigned long size_in_bytes)
{
    int success = TRUE;
    //boudary checking:
    if (get_mode() & SRAMLD)
```

```
    {
        //access SRAM
        if ((start_address_offset + size_in_bytes) > SRAM_SIZE)
            success = FALSE;
    }
    else
    {
        //access VRAM
        if ((start_address_offset + size_in_bytes) > VRAM_SIZE)
            success = FALSE;
        else
            memcpy(array,Map_Address + start_address_offset,
                   size_in_bytes);//read out RAM and store in array
    }
    return success;
}
```

An appropriate **compiler description file** `MapDev.dsp` would be :

```
# Microsoft Developer Studio Project File - Name="DT2001"
  - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format
  Version 5.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=DT2001 - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project
!MESSAGE using NMAKE, use the Export Makefile command and run :
!MESSAGE
!MESSAGE NMAKE /f "DT2001.MAK".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line.
!MESSAGE For example:
!MESSAGE
!MESSAGE NMAKE /f "DT2001.MAK" CFG="DT2001 - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "DT2001 - Win32 Release"
!MESSAGE           (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "DT2001 - Win32 Debug"
!MESSAGE           (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "DT2001 - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir "Release"
# PROP Intermediate_Dir "Release"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
  "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
  "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x1809 /d "NDEBUG"
# ADD RSC /l 0x1809 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
```

```
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
  comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
  uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
  /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
  comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
  uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
  /dll /machine:I386

!ELSEIF  "$(CFG)" == "DT2001 - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir "Debug"
# PROP Intermediate_Dir "Debug"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D
  "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D
  "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x1809 /d "_DEBUG"
# ADD RSC /l 0x1809 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
  comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
  uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
  /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib
  comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
  uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
  /dll /debug /machine:I386 /pdbtype:sept

!ENDIF

# Begin Target

# Name "DT2001 - Win32 Release"
# Name "DT2001 - Win32 Debug"
# Begin Source File

SOURCE=.\DT2001.cpp
# End Source File
# Begin Source File

SOURCE=.\dt2001.def
# End Source File
# Begin Source File

SOURCE=.\DT2001.h
# End Source File
# End Target
# End Project
```

A sample **definition file** `DTMap01.h` would be :

```
// DTMAP01.h - include file for VxD DTMap01

#ifndef NotVxD

#include <vtoolsc.h>

#define MAPDEV_Major          1
#define MAPDEV_Minor          0
#define MAPDEV_DeviceID       UNDEFINED_DEVICE_ID
#define MAPDEV_Init_Order     UNDEFINED_INIT_ORDER

#define LPVOID PVOID

#endif

// This is the request structure that applications use
// to request services from the MAPDEV VxD.

typedef struct _MapDevRequest
{
    DWORD  mdr_ServiceID;          // supplied by caller
    LPVOID mdr_PhysicalAddress;    // supplied by caller
    DWORD  mdr_SizeInBytes;        // supplied by caller
    LPVOID mdr_LinearAddress;      // returned by VxD
    WORD   mdr_Selector;           // returned if 16-bit caller
    WORD   mdr_Status;             // MDR_xxxx code below
} MAPDEVREQUEST, *PMAPDEVREQUEST;

#define MDR_SERVICE_MAP       CTL_CODE(FILE_DEVICE_UNKNOWN, 1,
                              METHOD_NEITHER, FILE_ANY_ACCESS)
#define MDR_SERVICE_UNMAP     CTL_CODE(FILE_DEVICE_UNKNOWN, 2,
                              METHOD_NEITHER, FILE_ANY_ACCESS)

#define MDR_STATUS_SUCCESS    1
#define MDR_STATUS_ERROR      0
```

The associated **C source file** `DTMap01.c` for the VxD is :

```
//Deep Trace VxD driver for Windows95
//Erich Barnstedt

//VERSION V1.0

#define PAGENUM(p)     (((ULONG)(p)) >> 12)
#define PAGEOFF(p)     (((ULONG)(p)) & 0xFFF)
#define PAGEBASE(p)    (((ULONG)(p)) & ~0xFFF)
#define _NPAGES_(p, k) ((PAGENUM((char*)p+(k-1))
                        - PAGENUM(p)) + 1)

#define   DEVICE_MAIN
#include  "dtmap01.h"
#undef    DEVICE_MAIN

//Create device descriptor block (DDB):
Declare_Virtual_Device(MAPDEV)

//Handle "DeviceIOControl" control message:
DefineControlHandler(W32_DEVICEIOCONTROL, OnW32Deviceiocontrol);

//Control dispatcher:
BOOL __cdecl ControlDispatcher(
    DWORD dwControlMessage,
    DWORD EBX,
    DWORD EDX,
    DWORD ESI,
    DWORD EDI,
    DWORD ECX)
{
    START_CONTROL_DISPATCH
        ON_W32_DEVICEIOCONTROL(OnW32Deviceiocontrol);
    END_CONTROL_DISPATCH

    return TRUE;
}

// Device Mapping Function:
PVOID MapDevice(PVOID PhysAddress, DWORD SizeInBytes)
{
    PVOID Linear;

    //calc number of mem pages required:
    ULONG nPages = _NPAGES_(PhysAddress, SizeInBytes);

    //set aside linear mem space:
    Linear = PageReserve(PR_PRIVATE,nPages,PR_FIXED);

    //do physical to linear translation (map into user space):
    PageCommitPhys(PAGENUM(Linear),nPages,PAGENUM(PhysAddress),
                   PC_INCR | PC_USER |PC_WRITEABLE);

    return (PVOID) ((ULONG)Linear+PAGEOFF(PhysAddress));
}

// Device Unmapping Function:
// Input: Base of linear memory block
// Output: None
VOID UnmapDevice(PVOID LinearAddress)
{
    //free linear memory:
    PageFree((MEMHANDLE)PAGEBASE(LinearAddress),0);
}

// W32_DEVICEIOCONTROL Handler for Win32 Applications:
// Input: The dioc_InBuf field of the parameter structure
// contains a pointer to the request structure
// Output: Fields of the request structure are updated
// according to the request.
DWORD OnW32Deviceiocontrol(PIOCTLPARAMS p)
{
    PMAPDEVREQUEST pReq;

    switch (p->dioc_IOCtlCode)
    {
    case DIOC_OPEN:
    case DIOC_CLOSEHANDLE:
        break;
    //map request:
```

```
    case MDR_SERVICE_MAP:
        pReq = *(PMAPDEVREQUEST*)p->dioc_InBuf;
        pReq->mdr_LinearAddress =
                          MapDevice(pReq->mdr_PhysicalAddress,
                          pReq->mdr_SizeInBytes);
        if  (pReq->mdr_LinearAddress == NULL)
            pReq->mdr_Status = MDR_STATUS_ERROR;
        else
            pReq->mdr_Status =  MDR_STATUS_SUCCESS;

        break;
    //unmap request:
    case MDR_SERVICE_UNMAP:
        pReq = *(PMAPDEVREQUEST*)p->dioc_InBuf;
        UnmapDevice(pReq->mdr_LinearAddress);
        pReq->mdr_Status = MDR_STATUS_SUCCESS;
        break;
    default:
        return ERROR_INVALID_FUNCTION;
    }

    return DEVIOCTL_NOERROR;
}
```

An appropriate makefile DTMap01.mak needs to invoke special tools from the VxD development environment :

```
# DTMAP01.mak - makefile for VxD DTMap01

DEVICENAME = DTMAP01
DYNAMIC = 1
FRAMEWORK = C
DEBUG = 1
OBJECTS = DTMap01.OBJ

!include $(VTOOLSD)\include\vtoolsd.mak
!include $(VTOOLSD)\include\vxdtarg.mak

DTMap01.OBJ: DTMap01.c DTMap01.h
```

Appendix B: Traceboard API

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+         S t a r t   D T 2 0 0 1 . c p p                   +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
#include <windows.h>
#include <winioctl.h>
#include <conio.h>
#include <memory.h>
#include <stdio.h>
#define NotVxD
#include "DT2001.h"
#include "DTMap01.h"

#define BITS48 0x0000FFFFFFFFFFFF

/*//DLL Entry Point
Note:   The DLL entry point functoin is disables at the moment because Visual
        Basic doesn't seem to be able to call it; (no loadlib function provided).
  BOOL WINAPI DllEntryPoint (HINSTANCE hDLL, DWORD dwReason, LPVOID Reserved)
{
    BOOL Success = TRUE;

    switch (dwReason)
    {
    case DLL_PROCESS_ATTACH:
    {
        //reset_board();
        //set_map(PHY_ADDR);//set physical address of VRAM/SRAM
        //map_RAM();//set linear address of VRAM/SRAM
        break;
    }
    case DLL_PROCESS_DETACH:
    {
        //unmap_RAM();
        //reset_board();
        break;
    }
    case DLL_THREAD_ATTACH:  break;
    case DLL_THREAD_DETACH:  break;
    }
    r eturn Success;
}*/

//Global Declarations:
HANDLE hDevice;       //handle for VxD

//resets board
__declspec( dllexport ) void __stdcall reset_board(void)
{
    _outpd(BASE_ADDR | REG_RESET, ANYVALUE);
}

//Writes to ADDRMAP register
//input: physical address the VRAM/SRAM is to appear at
__declspec( dllexport ) void __stdcall set_map(unsigned long address)
{
    _outpd(BASE_ADDR | REG_ADDRMAP, address);
}

//Read the current physical address mapping
__declspec( dllexport ) unsigned long __stdcall get_map(void)
{
    return (_inpd(BASE_ADDR | REG_ADDRMAP) & 0xFF000000);
}

//Writes to the MODE register
__declspec( dllexport ) void __stdcall set_mode(unsigned long flags)
{
    _outp(BASE_ADDR | REG_MODE, (int)flags);
}

//Read the current mode
__declspec( dllexport ) unsigned long __stdcall get_mode(void)
{
    return (((unsigned long) _inp(BASE_ADDR | REG_MODE)) & 0x000000FF);
}

//Write to the trigger config register
```

```
__declspec( dllexport ) void __stdcall set_trig_config(unsigned long flags)
{
    _outp(BASE_ADDR | REG_TRIGCONFIG, (int)flags);
}

//Read from the trigger config register
__declspec( dllexport ) unsigned long __stdcall get_trig_config(void)
{
    return (((unsigned long) _inp(BASE_ADDR | REG_TRIGCONFIG)) & 0x0000003F);
}

//Reset the timestamp counter
__declspec( dllexport ) void __stdcall reset_timestamp(void)
{
    _outpd(BASE_ADDR | REG_TIMERESET, ANYVALUE);
}

//Write to the STOPCOUNT register
__declspec( dllexport ) void __stdcall set_stop_count(unsigned long flags)
{
    _outpd(BASE_ADDR | REG_STOPCOUNT, flags);
}

//Read from the STOPCOUNT register
__declspec( dllexport ) unsigned long __stdcall get_stop_count(void)
{
    return (_inpd(BASE_ADDR | REG_STOPCOUNT) & 0x00FF8000);
}

//Write to the HEADPTR register
__declspec( dllexport ) void __stdcall set_head_ptr(long int head_ptr)
{
    _outpd(BASE_ADDR | REG_HEADPTR, head_ptr);
}

//Read from the HEADPTR register
__declspec( dllexport ) unsigned long __stdcall get_head_ptr(void)
{
    return (_inpd(BASE_ADDR | REG_HEADPTR) & 0x00FF8000);
}

//Write to the OUTPUTENA register
__declspec( dllexport ) void __stdcall set_sbuf_output_enas(unsigned long flags)
{
    _outp(BASE_ADDR | REG_OUTPUTENA, (int)flags);
}

//Read from the OUTPUTENA register
__declspec( dllexport ) unsigned long __stdcall get_sbuf_output_enas(void)
{
    return (((unsigned long)_inp(BASE_ADDR | REG_OUTPUTENA)) & 0x0000003F);
}

//Read from the STATUS buffer
__declspec( dllexport ) unsigned long __stdcall get_trace_status(void)
{
    return (_inpd(BASE_ADDR | REG_STATUS) & 0x00008000);
}

//Read from the EISA ID register
__declspec( dllexport ) unsigned long __stdcall get_EISA_ID(void)
{
    return _inpd(BASE_ADDR | REG_EISAID);
}

//Read from EISA ID register and decode string
//input: pointer to a char array (min 7 chars long)
//M.Manzke 4th November '98
__declspec( dllexport ) unsigned long __stdcall get_EISAID(char *text_ID)
{
    unsigned int chars;
    unsigned int digits;
    unsigned long reg = _inpd(BASE_ADDR | REG_EISAID);
    chars = reg;
    digits = reg >> 16;

    text_ID[0] = (char) ( ( chars & 0x007C) >>  2) + 0x40;
    text_ID[1] = (char) ((( chars & 0x0003) <<  3) |
                        (( chars & 0xE000) >> 13))+ 0x40;
    text_ID[2] = (char) ( ( chars & 0x1F00) >>  8) + 0x40;
    text_ID[3] = (char) ( (digits & 0x00F0) >>  4) + '0';
    text_ID[4] = (char) ( (digits & 0x000F) >>  0) + '0';
    text_ID[5] = (char) ( (digits & 0xF000) >> 12) + '0';
```

```
    text_ID[6] = (char) ( (digits & 0x0F00) >>  8) + '0';
    text_ID[7] = 0;
    return 1;
}

//Stop trace
__declspec( dllexport ) void __stdcall stop_trace(void)
{
    set_mode(PAUSE);
}

//Map VRAM and SRAM into linear memory, returns a pointer to base (32bit)
__declspec( dllexport ) unsigned long __stdcall map_RAM(void)
{
    DWORD cbBytesReturned;//count of bytes returned from VxD
    MAPDEVREQUEST req; //VxD request structure
    PVOID inBuf[1];       //buffer for DevIOCrtl pointer to req structure

    const PCHAR VxDName = "\\\\.\\DTMAP01.VXD";
    const PCHAR VxDNameAlreadyLoaded = "\\\\.\\DTMAP01";


    hDevice = CreateFile(VxDName, 0,0,0, CREATE_NEW, FILE_FLAG_DELETE_ON_CLOSE, 0);
    if (hDevice == INVALID_HANDLE_VALUE)
        hDevice = CreateFile(VxDNameAlreadyLoaded, 0,0,0, CREATE_NEW,
FILE_FLAG_DELETE_ON_CLOSE,
0);
    if (hDevice == INVALID_HANDLE_VALUE)
        return (unsigned long) GetLastError();//error!
    else
    {
        //set up request structure:
        req.mdr_ServiceID = MDR_SERVICE_MAP;
        req.mdr_PhysicalAddress = (PVOID) get_map();
        req.mdr_SizeInBytes = VRAM_SIZE;
        inBuf[0] = &req;

        //call Win32 API Message DeviceIOControl:
        if(! DeviceIoControl(hDevice, MDR_SERVICE_MAP, inBuf, sizeof(PVOID), NULL, 0,
&cbBytesReturned, NULL))
            return 0;
        else
            return (unsigned long) req.mdr_LinearAddress;//Success!
    }
}

//Unmap VRAM and SRAM
//uses the VxD, frees the linear address space
__declspec ( dllexport ) int __stdcall unmap_RAM(void)
{
    DWORD cbBytesReturned;//count of bytes returned from VxD
    MAPDEVREQUEST req; //VxD request structure
    PVOID inBuf[1];
    const PCHAR VxDName = "\\\\.\\DTMAP01.VXD";
    const PCHAR VxDNameAlreadyLoaded = "\\\\.\\DTMAP01";
    if (hDevice == INVALID_HANDLE_VALUE)
        hDevice = CreateFile(VxDNameAlreadyLoaded, 0,0,0, CREATE_NEW,
FILE_FLAG_DELETE_ON_CLOSE,
0);
    if (hDevice == INVALID_HANDLE_VALUE)
    {
        return 0;//error! can't get a handle on VxD
    }
    else
    {
        //set up request structure:
        req.mdr_ServiceID = MDR_SERVICE_UNMAP;
        req.mdr_PhysicalAddress = (void *) get_map();
        req.mdr_SizeInBytes = VRAM_SIZE;
        inBuf[0]=&req;
        //call Win32 API Message DeviceIOControl:
        if (! DeviceIoControl(hDevice, MDR_SERVICE_UNMAP, inBuf, sizeof(PVOID), NULL, 0,
&cbBytesReturned, NULL))
        {
            return 0;//error! call to VxD failed
        }
        else
        {
            return 1; //success!
        }
    }
}
```

```
//read out SRAM or VRAM and store in array
//input: array of 32bit storage locations, offset address into the VRAM/SRAM and the #bytes to
read out
//output: TURE if success, FALSE if fail
__declspec( dllexport ) int __stdcall get_RAM(PBYTE Map_Address, unsigned long *array,unsigned
long start_address_offset,unsigned long size_in_bytes)
{
    int success = TRUE;
    //boudary checking:
    if (get_mode() & SRAMLD)
    {
        //access SRAM
        if ((start_address_offset + size_in_bytes) > SRAM_SIZE)
            success = FALSE;
    }
    else
    {
        //access VRAM
        if ((start_address_offset + size_in_bytes) > VRAM_SIZE)
            success = FALSE;
        else
            memcpy(array,Map_Address + start_address_offset,size_in_bytes);//read out RAM and
store in array
    }
    return success;
}

/* JD */
__declspec( dllexport ) signed __stdcall Log_RAM(char *filename, PBYTE baseaddr, unsigned long
startaddress, unsigned long endaddress)
{
    FILE *fptr=fopen(filename,"at");
    if (!fptr) {
        return -1;
    }

    fprintf(fptr, "Address  Sample   Contents     Sub-Bank Bank  Row \n");
    fprintf(fptr, "[HEX]    [Dec]    [Hex]          [Dec]    [DEC] [DEC] \n");
    fprintf(fptr, "-------  ------   ------------ -------- ----- -----\n\n");

    PBYTE address;
    PBYTE endaddr = baseaddr + endaddress;
    unsigned long sampleaddr;
    unsigned long subbank=0;
    unsigned long bank=0;
    unsigned long row=0;


    for (address = baseaddr + startaddress, sampleaddr = 0; address < endaddr; address += 8,
sampleaddr += 8) {

        subbank = ((address - baseaddr) / 8) % 4;
        bank = (((address - baseaddr) / 8) /  2048) % 2;
        row = (address - baseaddr) / 32768;

        fprintf(fptr,"%08lx %08li %012I64x %1li %8li %7li\n",
                address - baseaddr,
                (address - baseaddr) / 8,
                (*((LONGLONG*)address)) & BITS48,
                subbank,
                bank,
                row);
    }

    //for (address = baseaddr + startaddress, sampleaddr = 0; address < endaddr; address += 8,
sampleaddr += 8) {
    //  fprintf(fptr,"%08lx %08lx %012I64x\n",sampleaddr, sampleaddr >> 3,
(*((LONGLONG*)address)) & BITS48);
    //}

    fclose(fptr);
    return 0;
}

/* DEBUG CODE STARTS HERE */
/* RC */
__declspec( dllexport ) signed __stdcall save_RAM(char *filename,PBYTE base,
                        unsigned long start_sample,
                       unsigned long numsamples)            //Raw VRAM dump to disk
{
  FILE *fptr=fopen(filename,"wb");
  if (!fptr) {return -1;} //-1 is UNIXish error code.
  base+=(start_sample<<3); //each sample is 8 bytes
```

```
  unsigned long samples_left=numsamples;

  while (samples_left>8192)
  {
       fwrite(base,(8192<<3),1,fptr);
       base+=(8192<<3);
       samples_left-=8192;
  }
  fwrite(base,samples_left,8,fptr);
  fclose(fptr);
  return 0;
}


/* RC */
__declspec( dllexport ) void __stdcall wipe_RAM(PBYTE *baseaddr,
                                unsigned long startsample,
                                unsigned long endsample)
{
  unsigned long *base=(unsigned long *)baseaddr;
  base+=startsample<<1;
  for (unsigned long idx=(endsample-startsample); idx>0; idx--)
  {
    (*base)=0; base++;
    (*base)=0; base++;
  }
}


/* JD */
__declspec( dllexport ) signed __stdcall PRNS_Check(char *filename, PBYTE baseaddr, unsigned
long startvalue)
{
    FILE *fptr=fopen(filename,"wt");
    if (!fptr) {
        return -1;
    }

    fprintf(fptr, "PRNS Log File\n");
    fprintf(fptr, "Offset    Is           Should Be    EXOR\n");
    fprintf(fptr, "--------- ------------ ------------ ------------\n");

    LONGLONG PRNStart = (LONGLONG)startvalue & BITS48;
    LONGLONG PRN = PRNStart;

    PBYTE address;
    PBYTE endaddr;

    endaddr = baseaddr + 0x01000000;

    for (address = baseaddr; address < endaddr; address += 8) {
        if ((*(LONGLONG *)address & BITS48) != PRN) {
            fprintf(fptr,"+%08lx %012I64x %012I64x %012I64\n",address - baseaddr, *(LONGLONG
*)address & BITS48, PRN & BITS48, (*(LONGLONG *)address & BITS48) ^ (PRN & BITS48));
        }
        PRN = ((PRN << 1) | (((PRN & 0x000000800000) >> 23) ^ ((PRN & 800000000000) >> 47))) &
BITS48;
    }

    fprintf(fptr, "\n\n");

    fclose(fptr);
    return 0;
}


__declspec( dllexport ) signed __stdcall AIA_Check(char *filename, PBYTE baseaddr)
{
    FILE *fptr=fopen(filename,"wt");
    if (!fptr) {
        return -1;
    }

    fprintf(fptr, "Address in address Log File\n");
    fprintf(fptr, "Offset    Is           Should Be    EXOR\n");
    fprintf(fptr, "--------- ------------ ------------ ------------\n");

    PBYTE address;
    PBYTE endaddr;
    LONGLONG ADR;

    endaddr = baseaddr + 0x01000000;
```

```
    for (address = baseaddr, ADR = 0; address < endaddr; address += 8, ADR++) {
        //ADR = (((LONGLONG)address) | (((LONGLONG)address) << 32)) & 0x0000FFFFFFFFFFFF;
        if ((*(LONGLONG *)address & BITS48) != ADR) {
            fprintf(fptr,"%08lx %012I64x %012I64x %012I64x\n",address - baseaddr, *(LONGLONG
*)address & BITS48, ADR, (*(LONGLONG *)address & BITS48) ^ ADR);
        }
    }

    fprintf(fptr, "\n\n");

    fclose(fptr);
    return 0;
}


__declspec( dllexport ) signed __stdcall AIA_Check_Bar(char *filename, PBYTE baseaddr)
{
    FILE *fptr=fopen(filename,"wt");
    if (!fptr) {
        return -1;
    }

    fprintf(fptr, "Address in address complement Log File\n");
    fprintf(fptr, "Offset    Is           Should Be    EXOR\n");
    fprintf(fptr, "--------- ------------ ------------ ------------\n");

    PBYTE address;
    PBYTE endaddr;
    LONGLONG ADR;

    endaddr = baseaddr + 0x01000000;

    for (address = baseaddr; address < endaddr; address += 8) {
        //ADR = (~(((LONGLONG)address) | (((LONGLONG)address) << 32))) & 0x0000FFFFFFFFFFFF;
        if ((*(LONGLONG *)address & BITS48) != ADR) {
            fprintf(fptr,"+%08lx %012I64x %012I64x %012I64x\n",address - baseaddr, *(LONGLONG
*)address & BITS48, ADR, (*(LONGLONG *)address & BITS48) ^ ADR);
        }
    }

    fprintf(fptr, "\n\n");

    fclose(fptr);
    return 0;
}


__declspec( dllexport ) void __stdcall PRNS_Fill(PBYTE baseaddr, unsigned long startvalue)
{
    LONGLONG PRNStart = (LONGLONG)startvalue & 0x0000FFFFFFFFFFFF;
    LONGLONG PRN = PRNStart;

    PBYTE address;
    PBYTE endaddr;

    endaddr = baseaddr + 0x01000000;

    for (address = baseaddr; address < endaddr; address += 8) {
        *(LONGLONG *)address = PRN;
        PRN = (PRN << 1) | (((PRN & 0x000000800000) >> 23) ^ ((PRN & 800000000000) >> 47));
    }

}


__declspec( dllexport ) void __stdcall AIA_Fill(PBYTE baseaddr)
{
    LONGLONG ADR;

    PBYTE address;
    PBYTE endaddr;

    endaddr = baseaddr + 0x01000000;

    for (address = baseaddr, ADR = 0; address < endaddr; address += 8, ADR++) {
        //ADR = (((LONGLONG)address) | (((LONGLONG)address) << 32)) & 0x0000FFFFFFFFFFFF;
        *(LONGLONG *)address = ADR;
    }
}


__declspec( dllexport ) void __stdcall AIA_Fill_Bar(PBYTE baseaddr)
{
```

```
    LONGLONG ADR;

    PBYTE address;
    PBYTE endaddr;

    endaddr = baseaddr + 0x01000000;

    for (address = baseaddr, ADR = 0; address < endaddr; address += 8, ADR++) {
        //ADR = (~(((LONGLONG)address) | (((LONGLONG)address) << 32))) & 0x0000FFFFFFFFFFFF;
        *(LONGLONG *)address = ~ADR;
    }
}


// write trigger or trace filter to SRAM
__declspec( dllexport ) unsigned long __stdcall set_SRAM(PBYTE baseaddr, unsigned long
w_sram_address, unsigned long t_t_channel)
{

    LONGLONG srambuffer;
    PBYTE address;

    address = baseaddr + w_sram_address;

    //debug only -start ++++++++++++++++++++++++++++++++++++++++++++++++++

    FILE *fptr=fopen("Debug.log","at");

    if (!fptr) {
        return -1;
    }

    //fprintf(fptr, "opened file for set_SRAM function \n\n");

    //fprintf(fptr, "PBYTE baseaddr : set_SRAM \n");
    //fprintf(fptr, "%d\n", baseaddr);

    //fprintf(fptr, "unsigned long w_sram_address : set_SRAM \n");
    //fprintf(fptr, "%d\n", w_sram_address);

    //fprintf(fptr, "unsigned long t_t_channel : set_SRAM \n");
    //fprintf(fptr, "%d\n", t_t_channel);


    //debug only - end ++++++++++++++++++++++++++++++++++++++++++++++++


    //select the SRAM and the trigger/trace-filter channel(s)

    //           0x0000,0000,0000,0000
    srambuffer = 0x0000000000000000;

    if ((t_t_channel & SRAM0_SELECT)>0) // SRAM0 0x00000010
    {
        if ((t_t_channel & CHANNEL1_SELECT)>0) //channnel # 1 0x00000001
        {
            srambuffer = srambuffer | SRAM0D0;       //0x0000000000000001
        }
        if ((t_t_channel & CHANNEL2_SELECT)>0) //channnel # 2 0x00000002
        {
            srambuffer = srambuffer | SRAM0D1;       //0x0000000000000002
        }
        if ((t_t_channel & CHANNEL3_SELECT)>0) //channnel # 3 0x00000004
        {
            srambuffer = srambuffer | SRAM0D2;       //0x0000000000000004
        }
        if ((t_t_channel & CHANNEL4_SELECT)>0) //channnel # 4 0x00000008
        {
            srambuffer = srambuffer | SRAM0D3;       //0x0000000000000008
        }
        *(LONGLONG *)address = srambuffer | (*(LONGLONG *)address & 0xFFFFFFFFFFFFFFF0);
    }
    else if ((t_t_channel & SRAM1_SELECT)>0) // SRAM1 0x00000020
    {
        if ((t_t_channel & CHANNEL1_SELECT)>0) //channnel # 1 0x00000001
        {
            srambuffer = srambuffer | SRAM1D0;       //0x0000000000000010
        }
        if ((t_t_channel & CHANNEL2_SELECT)>0) //channnel # 2 0x00000002
        {
            srambuffer = srambuffer | SRAM1D1;       //0x0000000000000020
        }
```

```
        if ((t_t_channel & CHANNEL3_SELECT)>0) //channnel # 3 0x00000004
        {
            srambuffer = srambuffer | SRAM1D2;       //0x0000000000000040
        }
        if ((t_t_channel & CHANNEL4_SELECT)>0) //channnel # 4 0x00000008
        {
            srambuffer = srambuffer | SRAM1D3;       //0x0000000000000080
        }
        *(LONGLONG *)address = srambuffer | (*(LONGLONG *)address & 0xFFFFFFFFFFFFFF0F);
    }
    else if ((t_t_channel & SRAM2_SELECT)>0) // SRAM2 0x00000040
    {
        if ((t_t_channel & CHANNEL1_SELECT)>0) //channnel # 1 0x00000001
        {
            srambuffer = srambuffer | SRAM2D0;       //0x0000000000000100
        }
        if ((t_t_channel & CHANNEL2_SELECT)>0) //channnel # 2 0x00000002
        {
            srambuffer = srambuffer | SRAM2D1;       //0x0000000000000200
        }
        if ((t_t_channel & CHANNEL3_SELECT)>0) //channnel # 3 0x00000004
        {
            srambuffer = srambuffer | SRAM2D2;       //0x0000000000000400
        }
        if ((t_t_channel & CHANNEL4_SELECT)>0) //channnel # 4 0x00000008
        {
            srambuffer = srambuffer | SRAM2D3;       //0x0000000000000800
        }
         *(LONGLONG *)address = srambuffer | (*(LONGLONG *)address & 0xFFFFFFFFFFFFF0FF);
    }


    //fprintf(fptr, "sram DATA before write: \n");
    //fprintf(fptr, "%d\n", srambuffer);


    fclose(fptr);

    return (unsigned long)srambuffer;
}



// read trigger or trace filter to SRAM
__declspec( dllexport ) unsigned long __stdcall get_SRAM(PBYTE baseaddr, unsigned long
r_sram_address)
{


    //unsigned long srambuffer;
    PBYTE address = baseaddr + r_sram_address;


    //FILE *fptr=fopen("Debug.log","at");

    //if (!fptr) {
    //  return -1;
    //}

    //fprintf(fptr, "opened file for get_SRAM function \n\n");

    //fprintf(fptr, "PBYTE baseaddr : get_SRAM \n");
    //fprintf(fptr, "%d\n", baseaddr);

    //fprintf(fptr, "unsigned long r_sram_address : get_SRAM \n");
    //fprintf(fptr, "%d\n", r_sram_address);



    //srambuffer = *(unsigned long *)address;

    //fprintf(fptr, "sram DATA before read : get_SRAM \n");
    //fprintf(fptr, "%u\n", srambuffer);

    //fclose(fptr);

    return (*((unsigned long*)address));
}


//Verify VRAM data
```

```
__declspec( dllexport ) signed __stdcall verify_VRAM(PBYTE baseaddr, unsigned long
startaddress, unsigned long endaddress)
{

    PBYTE address;
    PBYTE endaddr = baseaddr + endaddress;
    unsigned long sampleaddr;
    unsigned long errornumber = 0;
    unsigned long synchcount = 0;
    unsigned int  sychflag = 0;
    unsigned int  resynch_req_counter = 0;

    LONGLONG counter00_03 = 0;
    LONGLONG counterout = 0;


FILE *fptr=fopen("VRAM_Error.log","at");
    if (!fptr) {
        return 2;
    }


    fprintf(fptr,"%08lx %08lx\n", startaddress, endaddress);
    fprintf(fptr, "Address  Sample   Contents     Sub-Bank Bank  Row \n");
    fprintf(fptr, "[HEX]    [Dec]    [Hex]          [Dec]    [DEC] [DEC] \n");
    fprintf(fptr, "-------  ------   ------------ -------- ----- -----\n\n");


    for (address = baseaddr + startaddress, sampleaddr = 0; address < endaddr; address += 8,
sampleaddr += 8) {


        if (sychflag == 0)
        {

            sychflag = 1;

            //Synchronies counter with first VRAM data
            while ((counterout != ((*((LONGLONG*)address)) & BITS48)) && (synchcount < 100)) {

                ++synchcount;

                if (counter00_03 >= 15)
                {
                    counter00_03 =0;
                }
                else
                {
                    ++counter00_03;
                }

                counterout = ((counterout << 4) | counter00_03) & BITS48;

                if (synchcount == 100)
                {
                    fprintf(fptr, " Aborted synchronization after 100 iterations.\n");
                    fprintf(fptr,"%08lx %08li %012I64x %1li %8li %7li\n\n",
                            address - baseaddr,
                            (address - baseaddr) / 8,
                            (*((LONGLONG*)address)) & BITS48,
                            ((address - baseaddr) / 8) % 4,           //subbank
                            (((address - baseaddr) / 8) /  2048) % 2,  //bank
                            (address - baseaddr) / 32768);            //row
                    return 2; //Error
                }

            } // End of synchronization loop

        }
        else // compair VRAM against the counter
        {

            if (counter00_03 >= 15)
            {
                counter00_03 =0;
            }
            else
            {
                ++counter00_03;
            }

            counterout = ((counterout << 4) | counter00_03) & BITS48;
```

```
            if (counterout != ((*((LONGLONG*)address)) & BITS48))
            {
                errornumber = 1;
                fprintf(fptr,"%08lx %08li %012I64x %1li %8li %7li\n",
                        address - baseaddr,
                        (address - baseaddr) / 8,
                        (*((LONGLONG*)address)) & BITS48,
                        ((address - baseaddr) / 8) % 4,           //subbank
                        (((address - baseaddr) / 8) /  2048) % 2,  //bank
                        (address - baseaddr) / 32768);            //row

                fprintf(fptr,"%012I64x\n", counterout);

                ++resynch_req_counter;
                if (resynch_req_counter > 10)
                {
                    resynch_req_counter = 0;
                    sychflag = 0;
                    fprintf(fptr, "Initialize resynchronization after 10 corrupted
                            readings.\n");
                }

            }
        }
    }


    fclose(fptr); // close the error-log file

    if (errornumber == 1)
    {
        return 2; // Error
    }
    else
    {
        return 1; // Complete
    }
}


// write trigger or trace filter to SRAM
__declspec( dllexport ) unsigned long __stdcall wipe_SRAM(PBYTE baseaddr)
{

    LONGLONG address;
    PBYTE addressNew;

    for (address = 0; address < 65536; address += 1) {

        addressNew = (PBYTE)((LONGLONG)baseaddr + (address << 2));
        *(LONGLONG *)addressNew = *(LONGLONG *)addressNew & 0xFFFFFFFFFFFFF000;

    }

    return 1;
}


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+           E n d   D T 2 0 0 1 . c p p                      +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
+------------------------------------------------------------+
+            S t a r t   D T 2 0 0 1 . d e f                 +
+------------------------------------------------------------+
;DT2001 def DLL definition file
LIBRARY DT2001
CODE    PRELOAD MAVEABLE DISCARDABLE
DATA    PRELOAD SINGLE

EXPORTS
    ;The names of the exported DLL functions
    reset_board
    set_map
    get_map
    set_mode
    get_mode
    set_trig_config
    get_trig_config
    reset_timestamp
    set_stop_count
    get_stop_count
    set_head_ptr
    get_head_ptr
    set_sbuf_output_enas
    get_sbuf_output_enas
    get_trace_status
    get_EISA_ID
    get_EISAID
    stop_trace
    map_RAM
    get_RAM
    unmap_RAM
    save_RAM
    Log_RAM
    wipe_RAM
    PRNS_Check
    PRNS_Fill
    AIA_Check
    AIA_Check_Bar
    AIA_Fill
    AIA_Fill_Bar
    set_SRAM
    get_SRAM
    verify_VRAM
    wipe_SRAM
+------------------------------------------------------------+
+            E n d   D T 2 0 0 1 . d e f                     +
+------------------------------------------------------------+
```

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+               S t a r t   D T 2 0 0 1 . h                 +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//flag definitions for set_mode function:
#define SRAMLD     0x00000001
#define PAUSE      0x00000002
#define RECORD     0x00000004
#define WRAP       0x00000008
#define TIMECFG1   0x00000000
#define TIMECFG2   0x00000010
#define TIMECFG4   0x00000020
#define TIMECFG8   0x00000030
#define DMAENA     0x00000040
#define IRQENA     0x00000080

//flag definitions for set_trig_config function:
#define CHANNEL1_TRACE 0x00000000
#define CHANNEL1_TRIGG 0x00000001
#define CHANNEL2_TRACE 0x00000000
#define CHANNEL2_TRIGG 0x00000002
#define CHANNEL3_TRACE 0x00000000
#define CHANNEL3_TRIGG 0x00000004
#define CHANNEL4_TRACE 0x00000000
#define CHANNEL4_TRIGG 0x00000008
#define TRACE_CONTINUOUS 0x00000010
#define TRIG_IMMEDIATE 0x00000020

//flag definitions for set_stop_count function:
#define TRIG_AT_END       0x00000000
#define TRIG_AT_START  0x00FF8000

//flag definitions for set_sbuf_output_enas function:
#define BUFFER0ENA     0x00000001
#define BUFFER1ENA     0x00000002
#define BUFFER2ENA     0x00000004
#define BUFFER3ENA     0x00000008
#define BUFFER4ENA     0x00000010
#define BUFFER5ENA     0x00000020

// Deep Trace register offsets
#define BASE_ADDR      0x2000
#define REG_RESET      0x0
#define REG_ADDRMAP       0x4
#define REG_MODE       0x8
#define REG_TRIGCONFIG 0xC
#define REG_TIMERESET  0x10
#define REG_STOPCOUNT  0x14
#define REG_HEADPTR       0x18
#define REG_OUTPUTENA  0x1C
#define REG_STATUS     0x20
#define REG_EISAID     0x80

//SRAM buffer flags
#define SRAM0D0           0x0000000000000001
#define SRAM0D1           0x0000000000000002
#define SRAM0D2           0x0000000000000004
#define SRAM0D3           0x0000000000000008

#define SRAM1D0           0x0000000000000010
#define SRAM1D1           0x0000000000000020
#define SRAM1D2           0x0000000000000040
#define SRAM1D3           0x0000000000000080

#define SRAM2D0           0x0000000000000100
#define SRAM2D1           0x0000000000000200
#define SRAM2D2           0x0000000000000400
#define SRAM2D3           0x0000000000000800

//SRAM channel select
#define CHANNEL1_SELECT   0x00000001
#define CHANNEL2_SELECT   0x00000002
#define CHANNEL3_SELECT   0x00000004
#define CHANNEL4_SELECT   0x00000008

//SRAM chip select
#define SRAM0_SELECT   0x00000010
#define SRAM1_SELECT   0x00000020
#define SRAM2_SELECT   0x00000040


#define PHY_ADDR       0x07000000 //set physical address map to >256 MB, i.e. outside Win95
physical memory range
```

```
#define VRAM_SIZE      0x01000000
#define SRAM_SIZE      0x00080000

#define ANYVALUE 1
#define TRUE 1
#define FALSE 0

//Funtion prototypes
__declspec(dllexport) void __stdcall reset_board(void);
    //Reset board
__declspec( dllexport ) void __stdcall set_map(unsigned long address);            //Writes to
ADDRMAP register
__declspec( dllexport ) unsigned long __stdcall get_map(void);
    //Read the current address mapping
__declspec( dllexport ) void __stdcall set_mode(unsigned long flags);             //Writes to
the MODE register
__declspec( dllexport ) unsigned long __stdcall get_mode(void);
    //Read the current mode
__declspec( dllexport ) void __stdcall set_trig_config(unsigned long flags);//Write to the
trigger config register
__declspec( dllexport ) unsigned long __stdcall get_trig_config(void);            //Read from
the trigger config register
__declspec( dllexport ) void __stdcall reset_timestamp(void);
    //Reset the timestamp counter
__declspec( dllexport ) void __stdcall set_stop_count(unsigned long flags);       //Write to
the STOPCOUNT register
__declspec( dllexport ) unsigned long __stdcall get_stop_count(void);             //Read from
the STOPCOUNT register
__declspec( dllexport ) void __stdcall set_head_ptr(unsigned long head_ptr);//Write to the
HEADPTR register
__declspec( dllexport ) unsigned long __stdcall get_head_ptr(void);               //Read from
the HEADPTR register
__declspec( dllexport ) void __stdcall set_sbuf_output_enas(unsigned long flags);//Write to
the OUTPUTENA register
__declspec( dllexport ) unsigned long __stdcall get_sbuf_output_enas(void);       //Read from
the OUTPUTENA register
__declspec( dllexport ) unsigned long __stdcall get_trace_status(void);           //Read from
the STATUS buffer
__declspec( dllexport ) unsigned long __stdcall get_EISA_ID(void);                //Read from
the EISA ID register
__declspec( dllexport ) unsigned long __stdcall get_EISAID(char *text_ID); //Read from EISA ID
register amd decode
__declspec( dllexport ) void __stdcall stop_trace(void);
    //Stop trace
__declspec( dllexport ) unsigned long __stdcall map_RAM(void);
    //Map VRAM and SRAM into linear memory
__declspec( dllexport ) int __stdcall unmap_RAM(void);
    //Unmap VRAM and SRAM
//the get_RAM finction fills an array with the contents of V/SRAM specified:
__declspec( dllexport ) int __stdcall get_RAM(PBYTE Map_Address, unsigned long *array,unsigned
long start_address_offset,unsigned long size_in_bytes);
__declspec( dllexport ) signed __stdcall Log_RAM(char *filename, PBYTE baseaddr, unsigned long
startaddress, unsigned long endaddress);

/* DEBUG CODE */
//Raw VRAM dump to disk
__declspec( dllexport ) signed __stdcall save_RAM(char *filename, PBYTE *baseaddr, unsigned
long start_sample, unsigned long numsamples);
//Write zeros into VRAM
__declspec( dllexport ) void __stdcall wipe_RAM(BYTE *baseaddr,unsigned long startsample,
unsigned long endsample);
//Verify memory read/write
__declspec( dllexport ) int __stdcall PRNS_Check(char *filename, PBYTE *baseaddr, unsigned
long startvalue);
__declspec( dllexport ) void __stdcall PRNS_Fill(PBYTE *baseaddr, unsigned long startvalue);
__declspec( dllexport ) int __stdcall AIA_Check(char *filename, PBYTE *baseaddr);
__declspec( dllexport ) void __stdcall AIA_Fill(PBYTE *baseaddr);
__declspec( dllexport ) int __stdcall AIA_Check_Bar(char *filename, PBYTE *baseaddr);
__declspec( dllexport ) void __stdcall AIA_Fill_Bar(PBYTE *baseaddr);

// write trigger or trace filter to SRAM
__declspec( dllexport ) unsigned long __stdcall set_SRAM(PBYTE *baseaddr, unsigned long
w_sram_address, unsigned long t_t_channel);

// read trigger or trace filter to SRAM
__declspec( dllexport ) unsigned long __stdcall get_SRAM(PBYTE *baseaddr, unsigned long
r_sram_address);

// verfiy VRAM data
__declspec( dllexport ) signed __stdcall verify_VRAM(PBYTE baseaddr, unsigned long
startaddress, unsigned long endaddress);

// wipe the SRAM
```

```
__declspec( dllexport ) unsigned long __stdcall wipe_SRAM(PBYTE baseaddr);

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                E n d   D T 2 0 0 1 . h                     +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Appendix C: Visual Basic GUI source code

```
' ************************************
' DEEP TRACE TEST PROGRAM
' SCIEurope
'
' Version      : 2.1.0
' Last Updated : 15-04-1999
'
' Components   : DTDiag02.BAS
'                FrmMain.FRM  (This file)
'                FrmDumpOptions.FRM
'                FrmLogOptions.FRM
'                FrmPRNSOptions.FRM
' ************************************

' Force compiler to warn about undeclared variables
Option Explicit

Dim NumIterations As Long



' NOTE: ALL the code for the check boxes which appears
'       below follows the same pattern

' User sets/resets the DMA enable bit using the check box
Private Sub chkDmaena_Click()
    ' If Click event occured as a result of some other code
    ' changing the Value of the check box, do not update
    ' the registers. The RegLock is set by cmdRegUpdate_Click.
    If RegLock Then
        Exit Sub
    End If

    If chkDmaena.Value = 0 Then
        set_mode (get_mode And Not DT_DMAENA)
    Else
        set_mode (get_mode Or DT_DMAENA)
    End If
End Sub

' See above
Private Sub chkEna_Click(Index As Integer)
    If RegLock Then
        Exit Sub
    End If

    ' tmp_val is build up by ORing appropriate values
    ' depending on the contents of the check boxes
    Dim tmp_val As Long

    tmp_val = 0
    If chkEna(0).Value = 1 Then
        tmp_val = tmp_val Or DT_OUTPUTENA0
    End If
    If chkEna(1).Value = 1 Then
        tmp_val = tmp_val Or DT_OUTPUTENA1
    End If
    If chkEna(2).Value = 1 Then
        tmp_val = tmp_val Or DT_OUTPUTENA2
    End If
    If chkEna(3).Value = 1 Then
        tmp_val = tmp_val Or DT_OUTPUTENA3
    End If
    If chkEna(4).Value = 1 Then
        tmp_val = tmp_val Or DT_OUTPUTENA4
    End If
    If chkEna(5).Value = 1 Then
        tmp_val = tmp_val Or DT_OUTPUTENA5
    End If

    ' Write tmp_val to the output enable register
    Call set_sbuf_output_enas(tmp_val)

End Sub

' See above
Private Sub chkIrqena_Click()
    If RegLock Then
```

```
        Exit Sub
    End If

    If chkIrqena.Value = 0 Then
        set_mode (get_mode And Not DT_IRQENA)
    Else
        set_mode (get_mode Or DT_IRQENA)
    End If

End Sub

' See above
Private Sub chkPause_Click()
    If RegLock Then
        Exit Sub
    End If

    If chkPause.Value = 0 Then
        set_mode (get_mode And Not DT_PAUSE)
    Else
        set_mode (get_mode Or DT_PAUSE)
    End If
End Sub

' See above
Private Sub chkRecord_Click()
    If RegLock Then
        Exit Sub
    End If

    If chkRecord.Value = 0 Then
        set_mode (get_mode And Not DT_RECORD)
    Else
        set_mode (get_mode Or DT_RECORD)
    End If

End Sub

' See above
Private Sub chkSramld_Click()
    If RegLock Then
        Exit Sub
    End If

    If chkSramld.Value = 0 Then
        set_mode (get_mode And Not DT_SRAMLD)
    Else
        set_mode (get_mode Or DT_SRAMLD)
    End If
End Sub

' See above
Private Sub chkTrigConf_Click(Index As Integer)
    If RegLock Then
        Exit Sub
    End If

    ' tmp_val is build up by ORing appropriate values
    ' depending on the contents of the check boxes
    Dim tmp_val As Long

    tmp_val = 0
    If chkTrigConf(0).Value = 1 Then
        tmp_val = tmp_val Or DT_TRIGCONFIG0
    End If
    If chkTrigConf(1).Value = 1 Then
        tmp_val = tmp_val Or DT_TRIGCONFIG1
    End If
    If chkTrigConf(2).Value = 1 Then
        tmp_val = tmp_val Or DT_TRIGCONFIG2
    End If
    If chkTrigConf(3).Value = 1 Then
        tmp_val = tmp_val Or DT_TRIGCONFIG3
    End If
    If chkTrigConf(4).Value = 1 Then
        tmp_val = tmp_val Or DT_TRACE_CONTINUOUSLY
    End If
    If chkTrigConf(5).Value = 1 Then
        tmp_val = tmp_val Or DT_TRIGGER_IMMEDIATELY
    End If

    ' Write the contents of tmp_val to the Trigger
    ' Config register
```

```
    Call set_trig_config(tmp_val)

End Sub

' See above
Private Sub chkWrap_Click()
    If RegLock Then
        Exit Sub
    End If
    If chkWrap.Value = 0 Then
        set_mode (get_mode And Not DT_WRAP)
    Else
        set_mode (get_mode Or DT_WRAP)
    End If
End Sub

' See above
Private Sub cmbTimecfg_Click()
    If RegLock Then
        Exit Sub
    End If

    ' The value of the dropdown list is decoded and written to
    ' the mode register
    If cmbTimecfg.ListIndex = 0 Then set_mode ((get_mode And Not DT_TIMECFG) Or
DT_TIMECFG20ns)
    If cmbTimecfg.ListIndex = 1 Then set_mode ((get_mode And Not DT_TIMECFG) Or
DT_TIMECFG40ns)
    If cmbTimecfg.ListIndex = 2 Then set_mode ((get_mode And Not DT_TIMECFG) Or
DT_TIMECFG80ns)
    If cmbTimecfg.ListIndex = 3 Then set_mode ((get_mode And Not DT_TIMECFG) Or
DT_TIMECFG160ns)
End Sub

' Command button writes contents of address map text box
' to the address map register
Private Sub cmdAddrWrite_Click()
    Call set_map(CLng("&H" & txtAddr_Map))
End Sub

' Command button resets the board
Private Sub cmdBoardReset_Click()

    ' Manually reset the board
    Call reset_board
End Sub


Private Sub cmdDump_Click()
    ' Display Dump form
    frmDumpOptions.Show
End Sub

Private Sub cmdExitApp_Click()
    ' Terminate the program
    End
End Sub

Private Sub cmdFill_Click()
    Call PRNS_Fill(Linear_Mapping, &H12345678)
End Sub

' Command button writes contents of head pointer text box
' to the address map register
Private Sub cmdHeadWrite_Click()
    Call set_head_ptr(CLng("&H" + txtHead_Ptr))
End Sub


' Command button Maps/Remaps the Deep Trace on-board memory
Private Sub cmdMemRemap_Click()
    Linear_Mapping = map_RAM
    If Linear_Mapping = 0 Then
        Call Error001
    End If
End Sub


' 'Macro' to set up and execute a simle trace from an
' external sourse - used to develop 'Stress Test' macros
Private Sub cmdQuick_Click()

    If TIP = False Then
```

```
        ' Perform a quick test by tracing from some source

        cmdQuick.Caption = "Wait..."

        ' Reset the board
        Call reset_board

        ' Set address map register
        Call set_map(DT_RAM_Base)

        ' Wipe RAM
        Call wipe_RAM(Linear_Mapping, &H0, &H1FFFFF)

        ' Set PAUSE and RECORD
        Call set_mode(DT_PAUSE)

        ' Set Stop Count to max number of samples
        Call set_stop_count(&HFF8000)

        ' Set trigger config to trace continuously
        Call set_trig_config(DT_TRACE_CONTINUOUSLY Or DT_TRIGGER_IMMEDIATELY)

        ' Set RECORD
        Call set_mode(get_mode Or DT_RECORD)

        ' Set Head Pointer to 0x00000000
        Call set_head_ptr(&H0)

        ' Trace from external source
        Call set_sbuf_output_enas(&H3F)

        ' Release PAUSE bit in mode register
        Call set_mode(get_mode And Not DT_PAUSE)

        ' Change the caption
        cmdQuick.Caption = "Running"
        TIP = True
        Call cmdRegUpdate_Click
    Else
        TIP = False
        Call set_mode(get_mode Or DT_PAUSE)
        Call cmdRegUpdate_Click
        cmdQuick.Caption = "Quick Test"
    End If
End Sub

Private Sub cmdReadfromSRAM_Click()

        'Michael Manzke 24th February 1998

        Dim SRAM2_Value, SRAM1_Value, SRAM0_Value As Long

        ' Set address map register
        Call set_map(DT_RAM_Base)

        'Set SRAMLB bit of the Mode register
        Call set_mode(get_mode Or DT_SRAMLD)

        'Reset Record bit of the Mode register
        Call set_mode(get_mode And Not DT_RECORD)

        'Set Pause bit of the Mode register
        Call set_mode(get_mode Or DT_PAUSE)

        'Read trigger or trace values from the SRAM

        SRAM2_Value = 0
        SRAM2_Value = get_SRAM(Linear_Mapping, CLng("&H" & Sram_ms_bits.Text) * 4)
        If (SRAM2_Value And &H800) > 0 Then Sram2d3.Value = 1 Else Sram2d3.Value = 0
        If (SRAM2_Value And &H400) > 0 Then Sram2d2.Value = 1 Else Sram2d2.Value = 0
        If (SRAM2_Value And &H200) > 0 Then Sram2d1.Value = 1 Else Sram2d1.Value = 0
        If (SRAM2_Value And &H100) > 0 Then Sram2d0.Value = 1 Else Sram2d0.Value = 0

        SRAM1_Value = 0
        SRAM1_Value = get_SRAM(Linear_Mapping, CLng("&H" & Sram_bits.Text) * 4)
        If (SRAM1_Value And &H80) > 0 Then Sram1d3.Value = 1 Else Sram1d3.Value = 0
        If (SRAM1_Value And &H40) > 0 Then Sram1d2.Value = 1 Else Sram1d2.Value = 0
        If (SRAM1_Value And &H20) > 0 Then Sram1d1.Value = 1 Else Sram1d1.Value = 0
        If (SRAM1_Value And &H10) > 0 Then Sram1d0.Value = 1 Else Sram1d0.Value = 0

        SRAM0_Value = 0
        SRAM0_Value = get_SRAM(Linear_Mapping, CLng("&H" & Sram_ls_bits.Text) * 4)
```

```
        If (SRAM0_Value And &H8) > 0 Then Sram0d3.Value = 1 Else Sram0d3.Value = 0
        If (SRAM0_Value And &H4) > 0 Then Sram0d2.Value = 1 Else Sram0d2.Value = 0
        If (SRAM0_Value And &H2) > 0 Then Sram0d1.Value = 1 Else Sram0d1.Value = 0
        If (SRAM0_Value And &H1) > 0 Then Sram0d0.Value = 1 Else Sram0d0.Value = 0


End Sub

' The contents of all of the registers is loaded, decoded
' and written to the register summary window
Private Sub cmdRegUpdate_Click()

    ' Set RegLock to prevent Click events on check boxes
    RegLock = True

    ' Temp variables so you don't have to keep re-reading
    ' registers from the board
    Dim current_status, current_mode, current_trig_config As Long
    Dim current_stop_count, current_head_ptr As Long
    Dim current_address_map, current_EISA_ID, current_output_enas As Long

    ' Address map
    current_address_map = get_map
    txtAddr_Map.Text = pad_hex(Hex(current_address_map))

    ' Stop Count
    current_stop_count = get_stop_count
    txtStop_Count.Text = pad_hex(Hex(current_stop_count))

    ' Head Pointer
    current_head_ptr = get_head_ptr
    txtHead_Ptr.Text = pad_hex(Hex(current_head_ptr))

    ' EISA ID
    Dim FormattedID As String * 10
    current_EISA_ID = get_EISA_ID
    txtEISA_ID.Text = pad_hex(Hex(current_EISA_ID))
    get_EISAID (FormattedID)
    txtFormatted_ID.Text = FormattedID

    ' Mode Register
    current_mode = get_mode
    If ((current_mode And DT_SRAMLD) = DT_SRAMLD) Then chkSramld.Value = 1 Else
chkSramld.Value = 0
    If ((current_mode And DT_PAUSE) = DT_PAUSE) Then chkPause.Value = 1 Else chkPause.Value =
0
    If ((current_mode And DT_RECORD) = DT_RECORD) Then chkRecord.Value = 1 Else
chkRecord.Value = 0
    If ((current_mode And DT_WRAP) = DT_WRAP) Then chkWrap.Value = 1 Else chkWrap.Value = 0
    If ((current_mode And DT_DMAENA) = DT_DMAENA) Then chkDmaena.Value = 1 Else
chkDmaena.Value = 0
    If ((current_mode And DT_IRQENA) = DT_IRQENA) Then chkIrqena.Value = 1 Else
chkIrqena.Value = 0
    If ((current_mode And DT_TIMECFG) = DT_TIMECFG20ns) Then cmbTimecfg.ListIndex = 0
    If ((current_mode And DT_TIMECFG) = DT_TIMECFG40ns) Then cmbTimecfg.ListIndex = 1
    If ((current_mode And DT_TIMECFG) = DT_TIMECFG80ns) Then cmbTimecfg.ListIndex = 2
    If ((current_mode And DT_TIMECFG) = DT_TIMECFG160ns) Then cmbTimecfg.ListIndex = 3

    ' Trigger/Trace Config
    current_trig_config = get_trig_config
    If ((current_trig_config And DT_TRIGCONFIG0) = DT_TRIGCONFIG0) Then chkTrigConf(0).Value =
1 Else chkTrigConf(0).Value = 0
    If ((current_trig_config And DT_TRIGCONFIG1) = DT_TRIGCONFIG1) Then chkTrigConf(1).Value =
1 Else chkTrigConf(1).Value = 0
    If ((current_trig_config And DT_TRIGCONFIG2) = DT_TRIGCONFIG2) Then chkTrigConf(2).Value =
1 Else chkTrigConf(2).Value = 0
    If ((current_trig_config And DT_TRIGCONFIG3) = DT_TRIGCONFIG3) Then chkTrigConf(3).Value =
1 Else chkTrigConf(3).Value = 0
    If ((current_trig_config And DT_TRACE_CONTINUOUSLY) = DT_TRACE_CONTINUOUSLY) Then
chkTrigConf(4).Value = 1 Else chkTrigConf(4).Value = 0
    If ((current_trig_config And DT_TRIGGER_IMMEDIATELY) = DT_TRIGGER_IMMEDIATELY) Then
chkTrigConf(5).Value = 1 Else chkTrigConf(5).Value = 0

    ' Serial Buffer Enables
    current_output_enas = get_sbuf_output_enas
    If ((current_output_enas And DT_OUTPUTENA0) = DT_OUTPUTENA0) Then chkEna(0).Value = 1 Else
chkEna(0).Value = 0
    If ((current_output_enas And DT_OUTPUTENA1) = DT_OUTPUTENA1) Then chkEna(1).Value = 1 Else
chkEna(1).Value = 0
    If ((current_output_enas And DT_OUTPUTENA2) = DT_OUTPUTENA2) Then chkEna(2).Value = 1 Else
chkEna(2).Value = 0
    If ((current_output_enas And DT_OUTPUTENA3) = DT_OUTPUTENA3) Then chkEna(3).Value = 1 Else
chkEna(3).Value = 0
```

```
    If ((current_output_enas And DT_OUTPUTENA4) = DT_OUTPUTENA4) Then chkEna(4).Value = 1 Else
chkEna(4).Value = 0
    If ((current_output_enas And DT_OUTPUTENA5) = DT_OUTPUTENA5) Then chkEna(5).Value = 1 Else
chkEna(5).Value = 0

    ' Status
    current_status = get_trace_status
    If ((current_status And DT_DONE) = DT_DONE) Then chkDone.Value = 1 Else chkDone.Value = 0

    RegLock = False

End Sub

Private Sub cmdStartTest_Click()

    NumIterations = CLng(txtNumIt.Text)

    Print #FileNum, "=== START ============================="
    Print #FileNum, "Deep Trace Stress Test Log"
    Print #FileNum, "Date:        ", Format(Now, "dd-mmm-yyyy")
    Print #FileNum, "Start Time:  ", Format(Now, "hh:mm:ss")
    Print #FileNum, "Iterations:  ", NumIterations
    Print #FileNum, "----------------------------------------"

    Dim i As Integer

    ' Set up the board
    Call set_mode(DT_WRAP Or DT_PAUSE)  ' Paused ready for cont. trace
    Call set_stop_count(&H800000)
    Call set_head_ptr(&H0)
    Call set_trig_config(DT_TRACE_CONTINUOUSLY Or DT_TRIGGER_IMMEDIATELY)
    Call set_mode(get_mode Or DT_RECORD)
    Call set_sbuf_output_enas(&H3F)

    Call StartContinuousTracing
    Print #FileNum, "Continuous tracing started at ", Format(Now, "hh:mm:ss")

    Call StartTracing
    Print #FileNum, "Manually triggered at ", Format(Now, "hh:mm:ss")

    TraceTimer.Enabled = True

End Sub

' Command button writes contents of stop count text box
' to the address map register
Private Sub cmdStopWrite_Click()
    Call set_stop_count(CLng("&H" & txtStop_Count))
End Sub

' Command button resets the time stamp counter
Private Sub cmdTimeReset_Click()
    ' Reset the timestamp counter
    reset_timestamp
End Sub

Private Sub cmdVerrifyVRAM_Click()

ContTest = Not ContTest

    If ContTest Then

        cmdVerrifyVRAM.Caption = "Stop VRAM Verification"
        ' Reset the board
        Call reset_board
        ' Set address map register
        Call set_map(DT_RAM_Base)

        Testindex = 0
        TimerCycle = 0

        secerrors(0).Caption = Str(0)
        secerrors(1).Caption = Str(0)
        secerrors(2).Caption = Str(0)
        secerrors(3).Caption = Str(0)
        secerrors(4).Caption = Str(0)
        secerrors(5).Caption = Str(0)
        secerrors(6).Caption = Str(0)
        secerrors(7).Caption = Str(0)
        secerrors(8).Caption = Str(0)
        secerrors(9).Caption = Str(0)
        secerrors(10).Caption = Str(0)
        secerrors(11).Caption = Str(0)
```

```
        secerrors(12).Caption = Str(0)
        secerrors(13).Caption = Str(0)
        secerrors(14).Caption = Str(0)
        secerrors(15).Caption = Str(0)

    Else
        cmdVerrifyVRAM.Caption = "Start VRAM Verification"
    End If

End Sub

' Decode user input (start & end addresses) and call
' get_RAM with an empty array to read a memory range
Private Sub cmdVRAMUpdate_Click()
    Dim from_addr, to_addr As Long

    from_addr = CLng("&H" & txtVRAMFrom)
    to_addr = CLng("&H" & txtVRAMTo)

    If Log_RAM("C:\WINDOWS\TEMP\DT.LOG", Linear_Mapping, from_addr, to_addr) = -1 Then
        Call Error002
    End If

    txtVRAM.LoadFile "C:\WINDOWS\TEMP\DT.LOG", rtfText

    Kill ("C:\WINDOWS\TEMP\DT.LOG")
End Sub


Private Sub cmdWipe_Click()
    Call wipe_RAM(Linear_Mapping, &H0, &H1FFFFF)
End Sub

Private Sub cmdWipeSRAM_Click()

        SRAM2_Channel4.BackColor = &HFF&
        SRAM2_Channel3.BackColor = &HFF&
        SRAM2_Channel2.BackColor = &HFF&
        SRAM2_Channel1.BackColor = &HFF&

        SRAM1_Channel4.BackColor = &HFF&
        SRAM1_Channel3.BackColor = &HFF&
        SRAM1_Channel2.BackColor = &HFF&
        SRAM1_Channel1.BackColor = &HFF&

        SRAM0_Channel4.BackColor = &HFF&
        SRAM0_Channel3.BackColor = &HFF&
        SRAM0_Channel2.BackColor = &HFF&
        SRAM0_Channel1.BackColor = &HFF&

        SRAM2_Channel4.Refresh
        SRAM2_Channel3.Refresh
        SRAM2_Channel2.Refresh
        SRAM2_Channel1.Refresh

        SRAM1_Channel4.Refresh
        SRAM1_Channel3.Refresh
        SRAM1_Channel2.Refresh
        SRAM1_Channel1.Refresh

        SRAM0_Channel4.Refresh
        SRAM0_Channel3.Refresh
        SRAM0_Channel2.Refresh
        SRAM0_Channel1.Refresh


        ' Set address map register
         Call set_map(DT_RAM_Base)

        'Set SRAMLB bit of the Mode register
        Call set_mode(get_mode Or DT_SRAMLD)

        'Reset Record bit of the Mode register
        Call set_mode(get_mode And Not DT_RECORD)

        'Set Pause bit of the Mode register
        Call set_mode(get_mode Or DT_PAUSE)

        'Wipe the SRAM
        Call wipe_SRAM(Linear_Mapping)

        SRAM2_Channel4.BackColor = &HFFFFFF
        SRAM2_Channel3.BackColor = &HFFFFFF
```

```
        SRAM2_Channel2.BackColor = &HFFFFFF
        SRAM2_Channel1.BackColor = &HFFFFFF

        SRAM1_Channel4.BackColor = &HFFFFFF
        SRAM1_Channel3.BackColor = &HFFFFFF
        SRAM1_Channel2.BackColor = &HFFFFFF
        SRAM1_Channel1.BackColor = &HFFFFFF

        SRAM0_Channel4.BackColor = &HFFFFFF
        SRAM0_Channel3.BackColor = &HFFFFFF
        SRAM0_Channel2.BackColor = &HFFFFFF
        SRAM0_Channel1.BackColor = &HFFFFFF

        If SRAM2_Channel4.ListCount > 0 Then SRAM2_Channel4.ListIndex = 0
        If SRAM2_Channel3.ListCount > 0 Then SRAM2_Channel3.ListIndex = 0
        If SRAM2_Channel2.ListCount > 0 Then SRAM2_Channel2.ListIndex = 0
        If SRAM2_Channel1.ListCount > 0 Then SRAM2_Channel1.ListIndex = 0

        If SRAM1_Channel4.ListCount > 0 Then SRAM1_Channel4.ListIndex = 0
        If SRAM1_Channel3.ListCount > 0 Then SRAM1_Channel3.ListIndex = 0
        If SRAM1_Channel2.ListCount > 0 Then SRAM1_Channel2.ListIndex = 0
        If SRAM1_Channel1.ListCount > 0 Then SRAM1_Channel1.ListIndex = 0

        If SRAM0_Channel4.ListCount > 0 Then SRAM0_Channel4.ListIndex = 0
        If SRAM0_Channel3.ListCount > 0 Then SRAM0_Channel3.ListIndex = 0
        If SRAM0_Channel2.ListCount > 0 Then SRAM0_Channel2.ListIndex = 0
        If SRAM0_Channel1.ListCount > 0 Then SRAM0_Channel1.ListIndex = 0

        SRAM2_Channel4.Refresh
        SRAM2_Channel3.Refresh
        SRAM2_Channel2.Refresh
        SRAM2_Channel1.Refresh

        SRAM1_Channel4.Refresh
        SRAM1_Channel3.Refresh
        SRAM1_Channel2.Refresh
        SRAM1_Channel1.Refresh

        SRAM0_Channel4.Refresh
        SRAM0_Channel3.Refresh
        SRAM0_Channel2.Refresh
        SRAM0_Channel1.Refresh


End Sub

Private Sub cmdWriteToSram_Click()

        'Michael Manzke 24th February 1998

        Dim Trigger_Trace_Channel As Long
        Dim Trigger_Trace_Settings1 As Long
        Dim Trigger_Trace_Settings2 As Long
        Dim Trigger_Trace_Settings3 As Long


        'Set SRAMLB bit of the Mode register
        Call set_mode(get_mode Or DT_SRAMLD)

        'Reset Record bit of the Mode register
        Call set_mode(get_mode And Not DT_RECORD)

        'Set Pause bit of the Mode register
        Call set_mode(get_mode Or DT_PAUSE)

        Trigger_Trace_Channel = &H0
        If check_Channel1.Value Then 'Trigger or Trace-Filter Channel # 1
            Trigger_Trace_Channel = Trigger_Trace_Channel Or &H1
        End If

        If check_Channel2.Value Then 'Trigger or Trace-Filter Channel # 2
            Trigger_Trace_Channel = Trigger_Trace_Channel Or &H2
        End If

        If Check_Channel3.Value Then 'Trigger or Trace-Filter Channel # 3
            Trigger_Trace_Channel = Trigger_Trace_Channel Or &H4
        End If

        If Check_Channel4.Value Then 'Trigger or Trace-Filter Channel # 4
            Trigger_Trace_Channel = Trigger_Trace_Channel Or &H8
        End If

        'Write triger or trace values into the SRAM
```

```
        Trigger_Trace_Channel = Trigger_Trace_Channel And &HF  'resets SRAM flags
        Trigger_Trace_Channel = Trigger_Trace_Channel Or &H10 'sets the SRAM0 flag
        Trigger_Trace_Settings1 = set_SRAM(Linear_Mapping, CLng("&H" & Sram_ls_bits.Text) * 4,
Trigger_Trace_Channel)

        Trigger_Trace_Channel = Trigger_Trace_Channel And &HF  'resets SRAM flags
        Trigger_Trace_Channel = Trigger_Trace_Channel Or &H20 'sets the SRAM1 flag
        Trigger_Trace_Settings2 = set_SRAM(Linear_Mapping, CLng("&H" & Sram_bits.Text) * 4,
Trigger_Trace_Channel)

        Trigger_Trace_Channel = Trigger_Trace_Channel And &HF  'resets SRAM flags
        Trigger_Trace_Channel = Trigger_Trace_Channel Or &H40 'sets the SRAM2 flag
        Trigger_Trace_Settings3 = set_SRAM(Linear_Mapping, CLng("&H" & Sram_ms_bits.Text) * 4,
Trigger_Trace_Channel)

        'Read trigger or trace values from teh SRAM
        'temp disabled: Call cmdReadfromSRAM_Click
        'the following code is for test porpose only =>
        '++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

        'If (Trigger_Trace_Settings1 And &H8) > 0 Then Sram0d3.Value = 1 Else Sram0d3.Value =
0
        'If (Trigger_Trace_Settings1 And &H4) > 0 Then Sram0d2.Value = 1 Else Sram0d2.Value =
0
        'If (Trigger_Trace_Settings1 And &H2) > 0 Then Sram0d1.Value = 1 Else Sram0d1.Value =
0
        'If (Trigger_Trace_Settings1 And &H1) > 0 Then Sram0d0.Value = 1 Else Sram0d0.Value =
0

        'If (Trigger_Trace_Settings2 And &H80) > 0 Then Sram1d3.Value = 1 Else Sram1d3.Value =
0
        'If (Trigger_Trace_Settings2 And &H40) > 0 Then Sram1d2.Value = 1 Else Sram1d2.Value =
0
        'If (Trigger_Trace_Settings2 And &H20) > 0 Then Sram1d1.Value = 1 Else Sram1d1.Value =
0
        'If (Trigger_Trace_Settings2 And &H10) > 0 Then Sram1d0.Value = 1 Else Sram1d0.Value =
0

        'If (Trigger_Trace_Settings3 And &H800) > 0 Then Sram2d3.Value = 1 Else Sram2d3.Value
= 0
        'If (Trigger_Trace_Settings3 And &H400) > 0 Then Sram2d2.Value = 1 Else Sram2d2.Value
= 0
        'If (Trigger_Trace_Settings3 And &H200) > 0 Then Sram2d1.Value = 1 Else Sram2d1.Value
= 0
        'If (Trigger_Trace_Settings3 And &H100) > 0 Then Sram2d0.Value = 1 Else Sram2d0.Value
= 0

        '++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
End Sub

Private Sub Command1_Click()
    frmLogOptions.Show
End Sub


Private Sub cmdStampTest_Click()
    Call set_mode(&H2)
    Call set_trig_config(&H10)
    Call set_stop_count(&H0)
    Call set_head_ptr(&H0)
    Call set_mode(get_mode Or DT_RECORD)
End Sub

Private Sub Command2_Click()
    frmPRNSOptions.Show
End Sub

Private Sub Command3_Click()
    Call AIA_Fill(Linear_Mapping)
End Sub

Private Sub Command4_Click()
    Call AIA_Fill_Bar(Linear_Mapping)
End Sub

Private Sub Command5_Click()

    Dim ms_bits, bits, ls_bits  As Long

    Sram_data.Text = String(12 - Len(Trim(Sram_data.Text)), "0") & Trim(Sram_data.Text)

    Sram_ms_bits.Text = Left(Sram_data.Text, 4)
    Sram_bits.Text = Mid(Sram_data.Text, 5, 4)
```

```
    Sram_ls_bits.Text = Right(Sram_data.Text, 4)

    ms_bits = CLng("&H" & Sram_ms_bits.Text) * 4
    bits = CLng("&H" & Sram_bits.Text) * 4
    ls_bits = CLng("&H" & Sram_ls_bits.Text) * 4

    shift_ms_bits.Text = Hex(ms_bits)
    shift_bits.Text = Hex(bits)
    shift_ls_bits.Text = Hex(ls_bits)

End Sub


' Called when program starts
Private Sub Form_Load()
    ' Initialise regLock variable
    RegLock = False

    ' Set the Deep Trace RAM base address
    ' Eventually we will read this from the registry
    ' of something
    DT_RAM_Base = &H9000000

    ' Set the address map register with the above value
    set_map (DT_RAM_Base)

    'Stop the continious test
    ContTest = False

    ' Map the Deep Trace RAM to a linear address
    Linear_Mapping = map_RAM

    ' Check for map errors
    If Linear_Mapping = 0 Then
        Call Error001
        End
    End If

    ' Update the registers
    Call cmdRegUpdate_Click

    FileNum = FreeFile
    Open "C:\WINDOWS\DESKTOP\DT2001.LOG" For Append As FileNum

    ProgressBar1.Visible = False

End Sub

Private Sub Form_Unload(Cancel As Integer)
    ' Terminate the application
    Call unmap_RAM
End Sub


Private Sub Scan_SRAM_Click()

        SRAM2_Channel4.Clear
        SRAM2_Channel3.Clear
        SRAM2_Channel2.Clear
        SRAM2_Channel1.Clear

        SRAM1_Channel4.Clear
        SRAM1_Channel3.Clear
        SRAM1_Channel2.Clear
        SRAM1_Channel1.Clear

        SRAM0_Channel4.Clear
        SRAM0_Channel3.Clear
        SRAM0_Channel2.Clear
        SRAM0_Channel1.Clear

        Dim SRAM_Value, SRAMIndex As Long

        ' Set address map register
        Call set_map(DT_RAM_Base)

        'Set SRAMLB bit of the Mode register
        Call set_mode(get_mode Or DT_SRAMLD)

        'Reset Record bit of the Mode register
        Call set_mode(get_mode And Not DT_RECORD)

        'Set Pause bit of the Mode register
```

```
        Call set_mode(get_mode Or DT_PAUSE)

        'Read trigger or trace values from the SRAM

        SRAM2_Channel4.BackColor = &HFF&
        SRAM2_Channel3.BackColor = &HFF&
        SRAM2_Channel2.BackColor = &HFF&
        SRAM2_Channel1.BackColor = &HFF&

        SRAM1_Channel4.BackColor = &HFF&
        SRAM1_Channel3.BackColor = &HFF&
        SRAM1_Channel2.BackColor = &HFF&
        SRAM1_Channel1.BackColor = &HFF&

        SRAM0_Channel4.BackColor = &HFF&
        SRAM0_Channel3.BackColor = &HFF&
        SRAM0_Channel2.BackColor = &HFF&
        SRAM0_Channel1.BackColor = &HFF&

        SRAM2_Channel4.Refresh
        SRAM2_Channel3.Refresh
        SRAM2_Channel2.Refresh
        SRAM2_Channel1.Refresh

        SRAM1_Channel4.Refresh
        SRAM1_Channel3.Refresh
        SRAM1_Channel2.Refresh
        SRAM1_Channel1.Refresh

        SRAM0_Channel4.Refresh
        SRAM0_Channel3.Refresh
        SRAM0_Channel2.Refresh
        SRAM0_Channel1.Refresh

        ProgressBar1.Min = 0
        ProgressBar1.Max = 65535
        ProgressBar1.Value = ProgressBar1.Min
        ProgressBar1.Visible = True

        For SRAMIndex = 0 To 65535 '65535

            ProgressBar1.Value = SRAMIndex
            SRAM_Value = 0
            SRAM_Value = get_SRAM(Linear_Mapping, (SRAMIndex * 4))

            If (SRAM_Value And &H800) > 0 Then SRAM2_Channel4.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))
            If (SRAM_Value And &H400) > 0 Then SRAM2_Channel3.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))
            If (SRAM_Value And &H200) > 0 Then SRAM2_Channel2.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))
            If (SRAM_Value And &H100) > 0 Then SRAM2_Channel1.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))

            If (SRAM_Value And &H80) > 0 Then SRAM1_Channel4.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))
            If (SRAM_Value And &H40) > 0 Then SRAM1_Channel3.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))
            If (SRAM_Value And &H20) > 0 Then SRAM1_Channel2.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))
            If (SRAM_Value And &H10) > 0 Then SRAM1_Channel1.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))

            If (SRAM_Value And &H8) > 0 Then SRAM0_Channel4.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))
            If (SRAM_Value And &H4) > 0 Then SRAM0_Channel3.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))
            If (SRAM_Value And &H2) > 0 Then SRAM0_Channel2.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))
            If (SRAM_Value And &H1) > 0 Then SRAM0_Channel1.AddItem (String(4 -
Len(Hex(SRAMIndex)), "0") & Hex(SRAMIndex))

        Next SRAMIndex

        ProgressBar1.Visible = False

        SRAM2_Channel4.BackColor = &HFFFFFF
        SRAM2_Channel3.BackColor = &HFFFFFF
        SRAM2_Channel2.BackColor = &HFFFFFF
        SRAM2_Channel1.BackColor = &HFFFFFF

        SRAM1_Channel4.BackColor = &HFFFFFF
        SRAM1_Channel3.BackColor = &HFFFFFF
```

```
        SRAM1_Channel2.BackColor = &HFFFFFF
        SRAM1_Channel1.BackColor = &HFFFFFF

        SRAM0_Channel4.BackColor = &HFFFFFF
        SRAM0_Channel3.BackColor = &HFFFFFF
        SRAM0_Channel2.BackColor = &HFFFFFF
        SRAM0_Channel1.BackColor = &HFFFFFF

        If SRAM2_Channel4.ListCount > 0 Then SRAM2_Channel4.ListIndex = 0
        If SRAM2_Channel3.ListCount > 0 Then SRAM2_Channel3.ListIndex = 0
        If SRAM2_Channel2.ListCount > 0 Then SRAM2_Channel2.ListIndex = 0
        If SRAM2_Channel1.ListCount > 0 Then SRAM2_Channel1.ListIndex = 0

        If SRAM1_Channel4.ListCount > 0 Then SRAM1_Channel4.ListIndex = 0
        If SRAM1_Channel3.ListCount > 0 Then SRAM1_Channel3.ListIndex = 0
        If SRAM1_Channel2.ListCount > 0 Then SRAM1_Channel2.ListIndex = 0
        If SRAM1_Channel1.ListCount > 0 Then SRAM1_Channel1.ListIndex = 0

        If SRAM0_Channel4.ListCount > 0 Then SRAM0_Channel4.ListIndex = 0
        If SRAM0_Channel3.ListCount > 0 Then SRAM0_Channel3.ListIndex = 0
        If SRAM0_Channel2.ListCount > 0 Then SRAM0_Channel2.ListIndex = 0
        If SRAM0_Channel1.ListCount > 0 Then SRAM0_Channel1.ListIndex = 0


        SRAM2_Channel4.Refresh
        SRAM2_Channel3.Refresh
        SRAM2_Channel2.Refresh
        SRAM2_Channel1.Refresh

        SRAM1_Channel4.Refresh
        SRAM1_Channel3.Refresh
        SRAM1_Channel2.Refresh
        SRAM1_Channel1.Refresh

        SRAM0_Channel4.Refresh
        SRAM0_Channel3.Refresh
        SRAM0_Channel2.Refresh
        SRAM0_Channel1.Refresh

End Sub


Private Sub TraceTimer_Timer()

    If ContTest = True Then

        TimerCycle = TimerCycle + 1

        Select Case TimerCycle

        Case 1

            Testindex = Testindex + 1

            frmMain.txtverificationcount.Text = Int(Testindex)

            txtVerificationdis.ForeColor = &HFF&
            txtVerificationdis.Text = "STARTED NEW TEST"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Time
            txtVerificationdis.Refresh

            lbVRAM01.BackColor = &HFFFFFF
            lbVRAM02.BackColor = &HFFFFFF
            lbVRAM03.BackColor = &HFFFFFF
            lbVRAM04.BackColor = &HFFFFFF
            lbVRAM05.BackColor = &HFFFFFF
            lbVRAM06.BackColor = &HFFFFFF
            lbVRAM07.BackColor = &HFFFFFF
            lbVRAM08.BackColor = &HFFFFFF
            lbVRAM09.BackColor = &HFFFFFF
            lbVRAM10.BackColor = &HFFFFFF
            lbVRAM11.BackColor = &HFFFFFF
            lbVRAM12.BackColor = &HFFFFFF
            lbVRAM13.BackColor = &HFFFFFF
            lbVRAM14.BackColor = &HFFFFFF
            lbVRAM15.BackColor = &HFFFFFF
            lbVRAM16.BackColor = &HFFFFFF

        Case 2
```

```
            ' Wipe RAM
            txtVerificationdis.ForeColor = &H80000008
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 01"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "WIPE
VRAM"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "FROM: " &
Hex(&H0)
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "TO: " &
Hex(&H1FFFFF)
            Where = InStr(txtVerificationdis.Text, Hex(&H1FFFFF))
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh
            Call wipe_RAM(Linear_Mapping, &H0, &H1FFFFF)

        Case 3

            ' Set PAUSE and RECORD
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 02"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "SET
PAUSE"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "SET
RECORD"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "SET RECORD")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh
            Call set_mode(DT_PAUSE)

        Case 4

            ' Set Stop Count to max number of samples
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 03"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "SET STOP
COUNT TO:"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) &
Hex(&HFFF800)
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, Hex(&HFFF800))
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh
            Call set_stop_count(&HFFF800)

        Case 5

            ' Set trigger config to trace continuously
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 04"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "SET
TRIGGER IMMEDIATELY"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "SET TRACE
CONTINUOUSLY"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "CONTINUOUSLY")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh
            Call set_trig_config(DT_TRACE_CONTINUOUSLY Or DT_TRIGGER_IMMEDIATELY)

        Case 6

            ' Set RECORD
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 05"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "SET
RECORD"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "SET RECORD")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh
            Call set_mode(get_mode Or DT_RECORD)

        Case 7

            ' Set Head Pointer to 0x00000000
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
```

```
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 06"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "SET HEAD
POINTER"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "TO: " &
Hex(&H0)
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "TO: " & Hex(&H0))
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh
            Call set_head_ptr(&H0)

        Case 8

            ' Trace from external source
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 07"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "ENABLE
SERIAL BUFFER"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "ENABLE SERIAL BUFFER")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh
            Call set_sbuf_output_enas(&H3F)

        Case 9

            ' Release PAUSE bit in mode register
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 08"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "RESET
PAUSE"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "RESET PAUSE")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh
            Call set_mode(get_mode And Not DT_PAUSE)

        Case 10

            ' update the GUI
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 09"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "UPDATE
GUI"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "UPDATE GUI")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh
            Call cmdRegUpdate_Click

       Case 11

            ' Trace Board reads SCI data
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 10"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "TRACE
BOARD READS SCI DATA"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "TRACE BOARD READS SCI DATA")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

        Case 12

            ' SET PAUSE bit in mode register
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 11"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "SET
PAUSE"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "SET PAUSE")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh
            Call set_mode(get_mode Or DT_PAUSE)


       Case 13
```

```
            'VRAM verrification section 1 - 000 000 to 0FF FFF [Hex]
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 12"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 1"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "000 000
TO 0FF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "0FF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &H0, &HFFFFF) = 1 Then
                lbVRAM01.BackColor = &HFFFF00   ' completed
            Else
                lbVRAM01.BackColor = &HFF&      ' error
                secerrors(0).Caption = Str(Val(secerrors(0).Caption) + 1)
            End If

       Case 14

            'VRAM verrification section 2 - 100 000 to 1FF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 13"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 2"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "100 000
TO 1FF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "1FF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &H100000, &H1FFFFF) = 1 Then
                lbVRAM02.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM02.BackColor = &HFF&    ' error
                secerrors(1).Caption = Str(Val(secerrors(1).Caption) + 1)
            End If

       Case 15

            'VRAM verrification section 3 - 200 000 to 2FF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 14"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 3"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "200 000
TO 2FF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "2FF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &H200000, &H2FFFFF) = 1 Then
                lbVRAM03.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM03.BackColor = &HFF&    ' error
                secerrors(2).Caption = Str(Val(secerrors(2).Caption) + 1)
            End If

       Case 16

            'VRAM verrification section 4 - 300 000 to 3FF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 15"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 4"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "300 000
TO 3FF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "3FF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh
```

```
            If verify_VRAM(Linear_Mapping, &H300000, &H3FFFFF) = 1 Then
                lbVRAM04.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM04.BackColor = &HFF&    ' error
                secerrors(3).Caption = Str(Val(secerrors(3).Caption) + 1)
            End If

       Case 17

            'VRAM verrification section 5 - 400 000 to 4FF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 16"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 5"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "400 000
TO 4FF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "4FF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &H400000, &H4FFFFF) = 1 Then
                lbVRAM05.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM05.BackColor = &HFF&    ' error
                secerrors(4).Caption = Str(Val(secerrors(4).Caption) + 1)
            End If

       Case 18

            'VRAM verrification section 6 - 500 000 to 5FF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 17"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 6"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "500 000
TO 5FF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "5FF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &H500000, &H5FFFFF) = 1 Then
                lbVRAM06.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM06.BackColor = &HFF&    ' error
                secerrors(5).Caption = Str(Val(secerrors(5).Caption) + 1)
            End If

       Case 19

            'VRAM verrification section 7 - 600 000 to 6FF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 18"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 7"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "600 000
TO 6FF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "6FF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &H600000, &H6FFFFF) = 1 Then
                lbVRAM07.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM07.BackColor = &HFF&    ' error
                secerrors(6).Caption = Str(Val(secerrors(6).Caption) + 1)
            End If


       Case 20

            'VRAM verrification section 8 - 700 000 to 7FF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
```

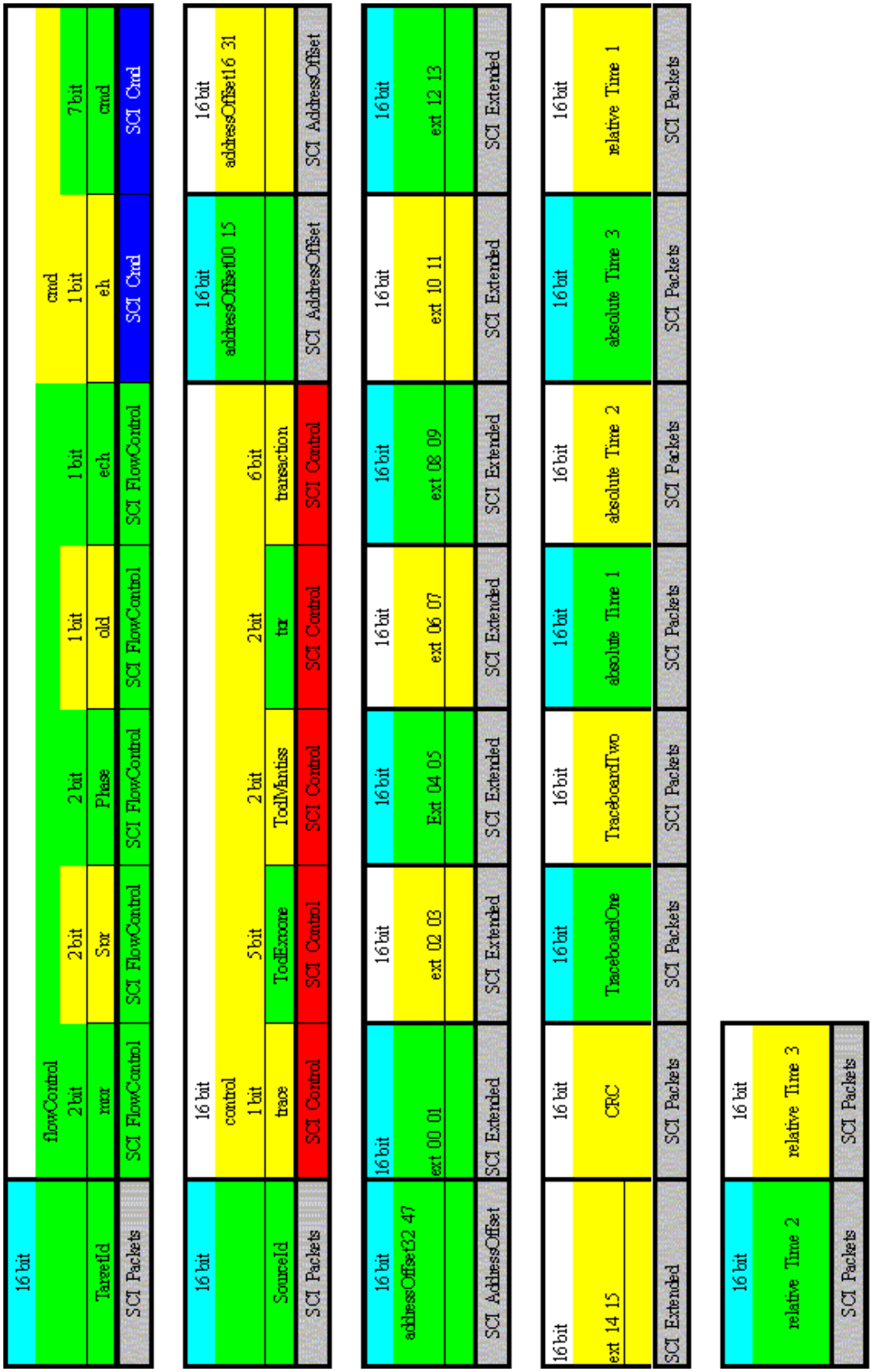
```
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 19"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 8"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "700 000
TO 7FF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "7FF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &H700000, &H7FFFFF) = 1 Then
                lbVRAM08.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM08.BackColor = &HFF&    ' error
                secerrors(7).Caption = Str(Val(secerrors(7).Caption) + 1)
            End If

       Case 21

            'VRAM verrification section 9 - 800 000 to 8FF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 20"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 9"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "800 000
TO 8FF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "8FF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &H800000, &H8FFFFF) = 1 Then
                lbVRAM09.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM09.BackColor = &HFF&    ' error
                secerrors(8).Caption = Str(Val(secerrors(8).Caption) + 1)
            End If

       Case 22

            'VRAM verrification section 10 - 900 000 to 9FF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 21"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 10"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "900 000
TO 9FF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "9FF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &H900000, &H9FFFFF) = 1 Then
                lbVRAM10.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM10.BackColor = &HFF&    ' error
                secerrors(9).Caption = Str(Val(secerrors(9).Caption) + 1)
            End If


       Case 23

            'VRAM verrification section 11 - A00 000 to AFF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 22"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 11"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "A00 000
TO AFF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "AFF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &HA00000, &HAFFFFF) = 1 Then
                lbVRAM11.BackColor = &HFFFF00 ' completed
            Else
```

```
                lbVRAM11.BackColor = &HFF&    ' error
                secerrors(10).Caption = Str(Val(secerrors(10).Caption) + 1)
            End If

        Case 24

            'VRAM verrification section 12 - B00 000 to BFF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 23"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 12"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "B00 000
TO BFF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "BFF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &HB00000, &HBFFFFF) = 1 Then
                lbVRAM12.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM12.BackColor = &HFF&    ' error
                secerrors(11).Caption = Str(Val(secerrors(11).Caption) + 1)
            End If

        Case 25

            'VRAM verrification section 13 - C00 000 to CFF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 24"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 13"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "C00 000
TO CFF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "CFF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &HC00000, &HCFFFFF) = 1 Then
                lbVRAM13.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM13.BackColor = &HFF&    ' error
                secerrors(12).Caption = Str(Val(secerrors(12).Caption) + 1)
            End If

        Case 26

            'VRAM verrification section 14 - D00 000 to DFF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 24"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 14"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "D00 000
TO DFF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "DFF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &HD00000, &HDFFFFF) = 1 Then
                lbVRAM14.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM14.BackColor = &HFF&    ' error
                secerrors(13).Caption = Str(Val(secerrors(13).Caption) + 1)
            End If


        Case 27

            'VRAM verrification section 15 - E00 000 to EFF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 25"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 15"
```

```
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "E00 000
TO EFF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "EFF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &HE00000, &HEFFFFF) = 1 Then
                lbVRAM15.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM15.BackColor = &HFF&    ' error
                secerrors(14).Caption = Str(Val(secerrors(14).Caption) + 1)
            End If


       Case 28

            'VRAM verrification section 16 - F00 000 to FFF FFF [Hex]

            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & Chr(13) &
Chr(10) & Time
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "STEP 26"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "VRAM
VERIFICATION SECTION 16"
            txtVerificationdis.Text = txtVerificationdis.Text & Chr(13) & Chr(10) & "F00 000
TO FFF FFF [Hex]"
            Where = Where + 10
            Where = InStr(Where, txtVerificationdis.Text, "FFF FFF [Hex]")
            txtVerificationdis.SelStart = Where - 1
            txtVerificationdis.Refresh

            If verify_VRAM(Linear_Mapping, &HF00000, &HFFFFFF) = 1 Then
                lbVRAM16.BackColor = &HFFFF00 ' completed
            Else
                lbVRAM16.BackColor = &HFF&    ' error
                secerrors(15).Caption = Str(Val(secerrors(15).Caption) + 1)
            End If


            TimerCycle = 0

        End Select

        'Call set_mode(get_mode Or DT_PAUSE)
        'Call frmMain.cmdRegUpdate


    End If


    'Call Beep

    'TraceTimer.Enabled = False

    'NumIterations = NumIterations - 1

    ' Stop Tracing
    'Call StopTracing
    'Print #FileNum, "Tracing stopped at ", Format(Now, "hh:mm:ss")
    'Print #FileNum, ""

    ' Record data

    ' Restart continuous tracing
    'Call StartContinuousTracing
    'Print #FileNum, "Continuous tracing started at ", Format(Now, "hh:mm:ss")

    ' Analyse data
    ' Code to be added here

    ' Start tracing
    'Call StartTracing
    'Print #FileNum, "Manually triggered at ", Format(Now, "hh:mm:ss")

    'TraceTimer.Enabled = True

    'If NumIterations = 0 Then
    '    TraceTimer.Enabled = False
    '    Print #FileNum, "----------------------------------------"
    '    Print #FileNum, "Stop Time:   ", Format(Now, "hh:mm:ss")
    '    Print #FileNum, "========================================"
    'End If
  End Sub
```

Appendix D: Request-send-packet with extended header and 0 bytes data

Packet Type 1

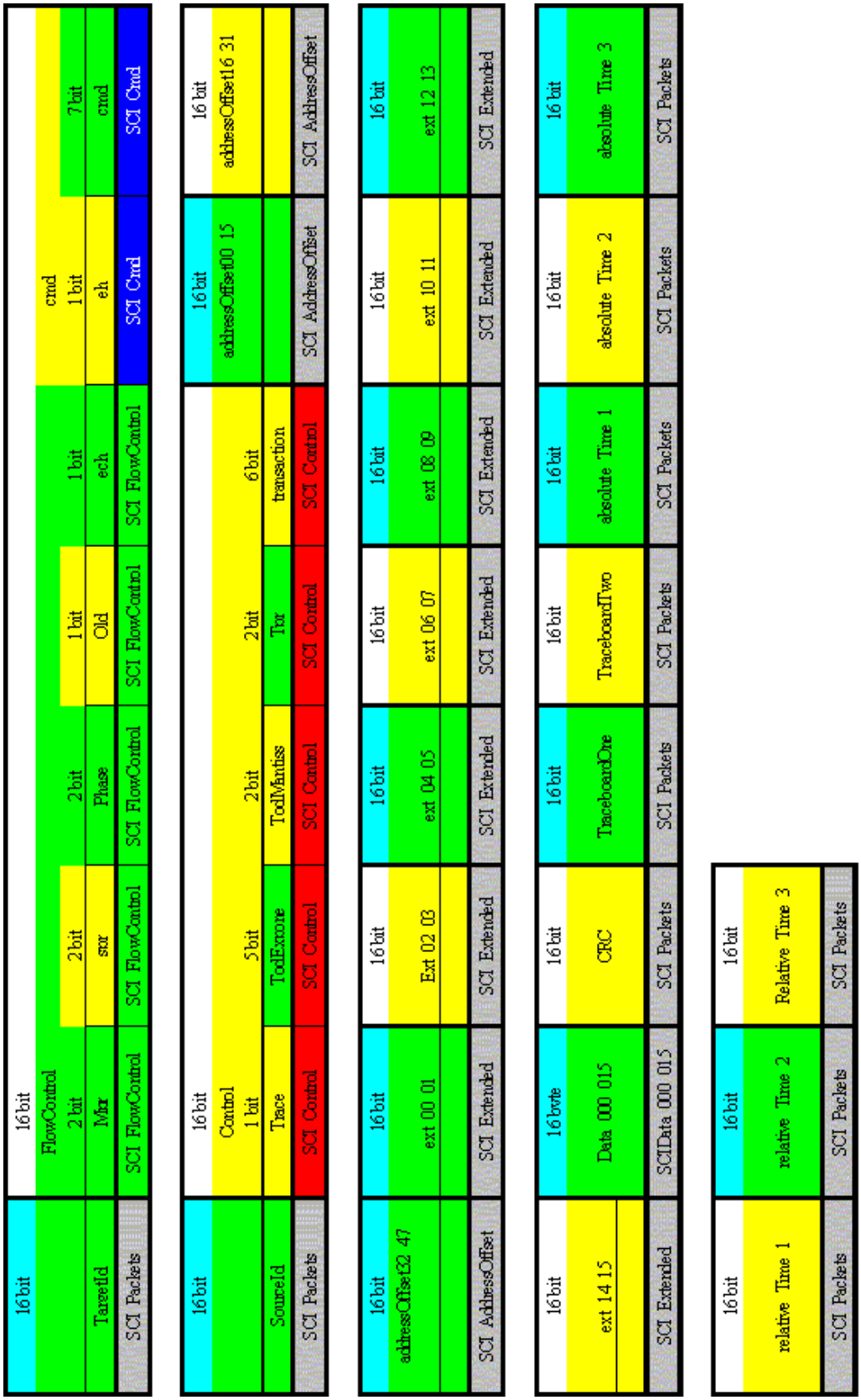


The following SQL Query retrieves all information related to a ***SCI packet type 1 Request-send-packet with extended header and 0 bytes data***. In this instance form a SCI packet with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3

FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId = SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId = SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)

WHERE (((SCI_Packets.TraceId) Like 1)
         AND ((SCI_Packets.PacketId) Like 3)
         AND ((SCI_Packets.Packet_Type_Id) Like 1));
```

Appendix D: Request-send-packet with extended header and 16 bytes data

Packet Type 2


16 bit
FlowControl
2 bit
2 bit
2 bit
1 bit
1 bit
cmd
1 bit
7 bit
TargetId
Mxr
sxr
Phase
Old
ech
eh
cmd
SCI Packets
SCI FlowControl
SCI FlowControl
SCI FlowControl
SCI FlowControl
SCI FlowControl
SCI Cmd
SCI Cmd
16 bit
Control
1 bit
5 bit
2 bit
2 bit
6 bit
16 bit
addressOffset16 31
SourceId
Trace
TodExcone
TodMantiss
Tsr
transaction
SCI Packets
SCI Control
SCI Control
SCI Control
SCI Control
SCI Control
SCI AddressOffset
SCI AddressOffset
16 bit
16 bit
16 bit
16 bit
16 bit
16 bit
addressOffset32 47
ext 00 01
Ext 02 03
ext 04 05
ext 06 07
ext 08 09
ext 10 11
ext 12 13
SCI AddressOffset
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCI Extended
16 bit
16 byte
16 bit
16 bit
16 bit
16 bit
16 bit
ext 14 15
Data 000 015
CRC
TraceboardOne
TraceboardTwo
absolute Time 1
absolute Time 2
absolute Time 3
SCI Extended
SCIData 000 015
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets
16 bit
16 bit
16 bit
relative Time 1
relative Time 2
Relative Time 3
SCI Packets
SCI Packets
SCI Packets

The following SQL Query retrieves all information related to a ***SCI packet type 2 Request-send-packet with extended header and 16 bytes data***. In this instance form a SCI packet with a TraceId =1 and a PacketId = 3.

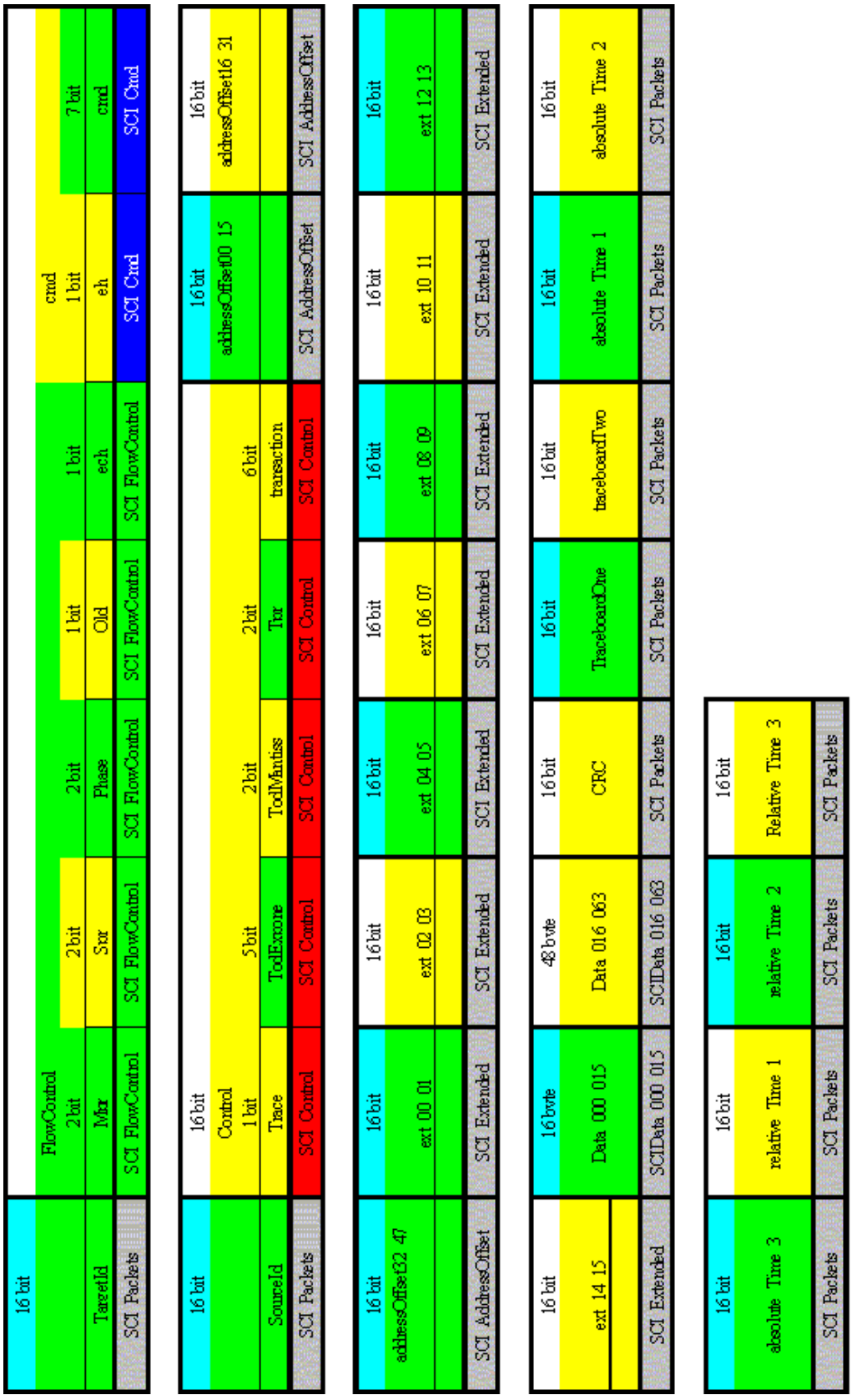
```
SELECT SCI_Packets.TraceId,
              SCI_Packets.PacketId,
              SCI_Packets.Packet_Type_Id,
              SCI_Packet_Type_Id.Packet_Type_Description,
              SCI_Packets.targetId,
              SCI_FlowControl.mpr,
              SCI_FlowControl.spr,
              SCI_FlowControl.phase,
              SCI_FlowControl.old,
              SCI_FlowControl.ech,
              SCI_Cmd.cmd_eh,
              SCI_Cmd.cmd_cmd,
              SCI_Packets.sourceId,
              SCI_Control.control_trace,
              SCI_Control.control_todExpone,
              SCI_Control.control_todMantiss,
              SCI_Control.control_tpr,
              SCI_Control.control_transaction,
              SCI_AddressOffset.addressOffset00_15,
              SCI_AddressOffset.addressOffset16_31,
              SCI_AddressOffset.addressOffset32_47,
              SCI_Extended.ext_00_01,
              SCI_Extended.ext_02_03,
              SCI_Extended.ext_04_05,
              SCI_Extended.ext_06_07,
              SCI_Extended.ext_08_09,
              SCI_Extended.ext_10_11,
              SCI_Extended.ext_12_13,
              SCI_Extended.ext_14_15,
              SCIData_000_015.SCIData_000_001,
              SCIData_000_015.SCIData_002_003,
              SCIData_000_015.SCIData_004_005,
              SCIData_000_015.SCIData_006_007,
              SCIData_000_015.SCIData_008_009,
              SCIData_000_015.SCIData_010_011,
              SCIData_000_015.SCIData_012_013,
              SCIData_000_015.SCIData_014_015,
              SCI_Packets.CRC,
              SCI_Packets.traceboardOne,
              SCI_Packets.traceboardTwo,
              SCI_Packets.absolute_Time_1,
              SCI_Packets.absolute_Time_2,
              SCI_Packets.absolute_Time_3,
              SCI_Packets.relative_Time_1,
              SCI_Packets.relative_Time_2,
              SCI_Packets.relative_Time_3
```

See next page for SQL Query continuation

```
FROM ((((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
        ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId = SCI_FlowControl.PacketId)
        AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
        AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
        AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
        AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
        AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCIData_000_015 ON (SCI_Packets.PacketId = SCIData_000_015.PacketId)
        AND (SCI_Packets.TraceId = SCIData_000_015.TraceId)

WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3)
       AND ((SCI_Packets.Packet_Type_Id) Like 2));
```

Appendix D: Request-send-packet with extended header and 64 bytes data

Packet Type 3

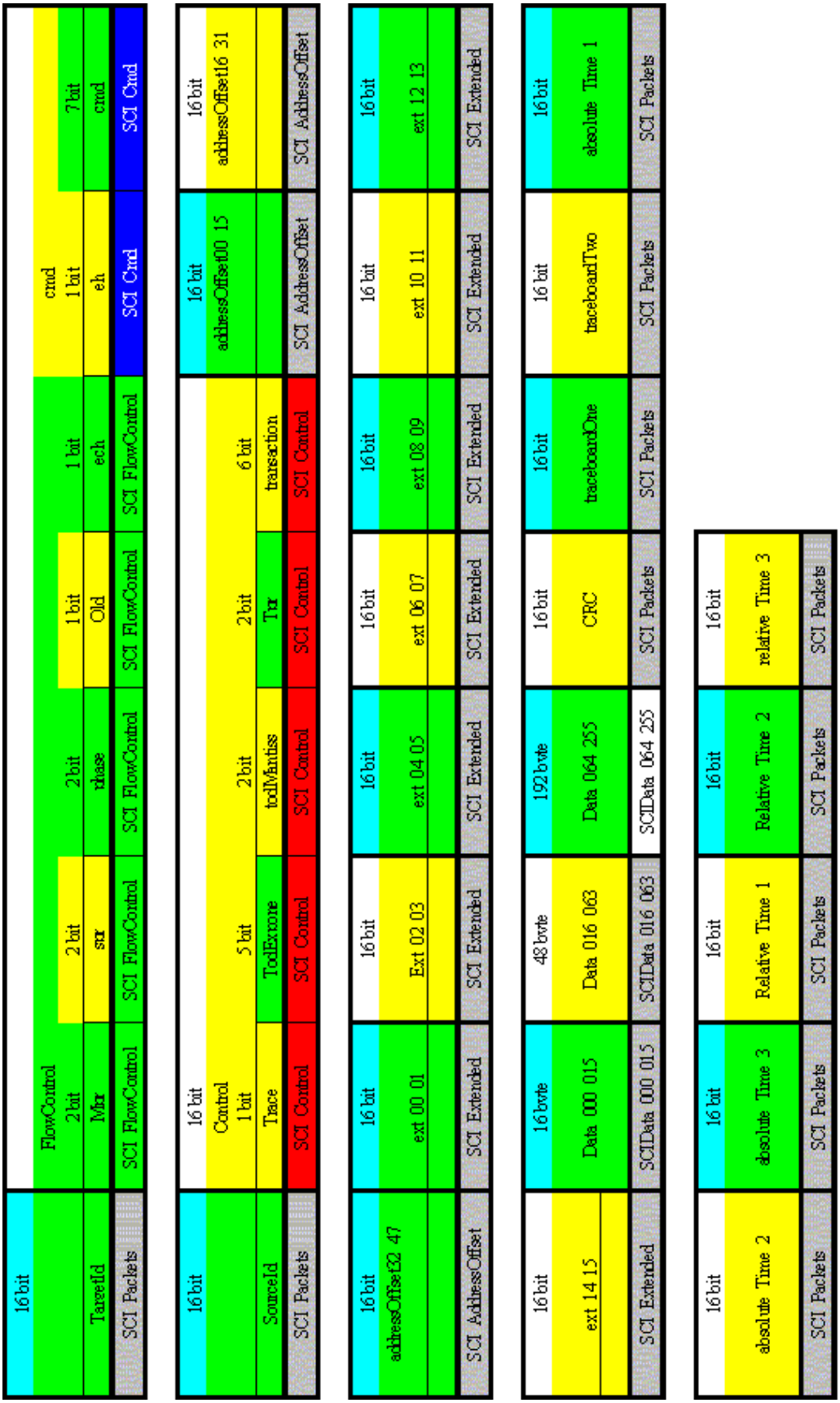


The following SQL Query retrieves all information related to a ***SCI packet type 3 Request-send-packet with extended header and 64 bytes data***. In this instance form a SCI packet with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old, SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh, SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01, SCI_Extended.ext_02_03, SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07, SCI_Extended.ext_08_09, SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13, SCI_Extended.ext_14_15,
             SCIData_000_015.SCIData_000_001, SCIData_000_015.SCIData_002_003,
             SCIData_000_015.SCIData_004_005, SCIData_000_015.SCIData_006_007,
             SCIData_000_015.SCIData_008_009, SCIData_000_015.SCIData_010_011,
             SCIData_000_015.SCIData_012_013, SCIData_000_015.SCIData_014_015,
             SCIData_016_063.SCIData_016_017, SCIData_016_063.SCIData_018_019,
             SCIData_016_063.SCIData_020_021, SCIData_016_063.SCIData_022_023,
             SCIData_016_063.SCIData_024_025, SCIData_016_063.SCIData_026_027,
             SCIData_016_063.SCIData_028_029, SCIData_016_063.SCIData_030_031,
             SCIData_016_063.SCIData_032_033, SCIData_016_063.SCIData_034_035,
             SCIData_016_063.SCIData_036_037, SCIData_016_063.SCIData_038_039,
             SCIData_016_063.SCIData_040_041, SCIData_016_063.SCIData_042_043,
             SCIData_016_063.SCIData_044_045, SCIData_016_063.SCIData_046_047,
             SCIData_016_063.SCIData_048_049, SCIData_016_063.SCIData_050_051,
             SCIData_016_063.SCIData_052_053, SCIData_016_063.SCIData_054_055,
             SCIData_016_063.SCIData_056_057, SCIData_016_063.SCIData_058_059,
             SCIData_016_063.SCIData_060_061, SCIData_016_063.SCIData_062_063,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2, SCI_Packets.relative_Time_3
FROM (((((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
        ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
              LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId = SCI_FlowControl.PacketId)
        AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId))
              LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
        AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
              LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
        AND (SCI_Packets.TraceId = SCI_Control.TraceId))
              LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
        AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
              LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
        AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
              LEFT JOIN SCIData_000_015 ON (SCI_Packets.PacketId = SCIData_000_015.PacketId)
        AND (SCI_Packets.TraceId = SCIData_000_015.TraceId))
              LEFT JOIN SCIData_016_063 ON (SCI_Packets.PacketId = SCIData_016_063.PacketId)
        AND (SCI_Packets.TraceId = SCIData_016_063.TraceId)

WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3)
       AND ((SCI_Packets.Packet_Type_Id) Like 3));
```

Appendix D: Request-send-packet with extended header and 256 bytes data

Packet Type 4


16 bit
TargetId
SCI Packets
FlowControl
2 bit
Mstr
SCI FlowControl
2 bit
str
SCI FlowControl
2 bit
rbase
SCI FlowControl
1 bit
Old
SCI FlowControl
1 bit
ech
SCI FlowControl
cmd
1 bit
eh
SCI Cmd
7 bit
cmd
SCI Cmd
16 bit
SourceId
SCI Packets
Control
1 bit
Trace
SCI Control
5 bit
ToolExtone
SCI Control
2 bit
todMantiss
SCI Control
2 bit
Tsr
SCI Control
6 bit
transaction
SCI Control
16 bit
addressOffset00 15
SCI AddressOffset
16 bit
addressOffset16 31
SCI AddressOffset
16 bit
addressOffset32 47
SCI AddressOffset
16 bit
ext 00 01
SCI Extended
16 bit
Ext 02 03
SCI Extended
16 bit
ext 04 05
SCI Extended
16 bit
ext 06 07
SCI Extended
16 bit
ext 08 09
SCI Extended
16 bit
ext 10 11
SCI Extended
16 bit
ext 12 13
SCI Extended
16 bit
ext 14 15
SCI Extended
16 byte
Data 000 015
SCIData 000 015
48 byte
Data 016 063
SCIData 016 063
192 byte
Data 064 255
SCIData 064 255
16 bit
CRC
SCI Packets
16 bit
traceboardOne
SCI Packets
16 bit
traceboardTwo
SCI Packets
16 bit
absolute Time 1
SCI Packets
16 bit
absolute Time 2
SCI Packets
16 bit
absolute Time 3
SCI Packets
16 bit
Relative Time 1
SCI Packets
16 bit
Relative Time 2
SCI Packets
16 bit
relative Time 3
SCI Packets

The following SQL Query retrieves all information related to a ***SCI packet type 4 Request-send-packet with extended header and 256 bytes data***. In this instance form a SCI packet with a TraceId =1 and a PacketId = 3.

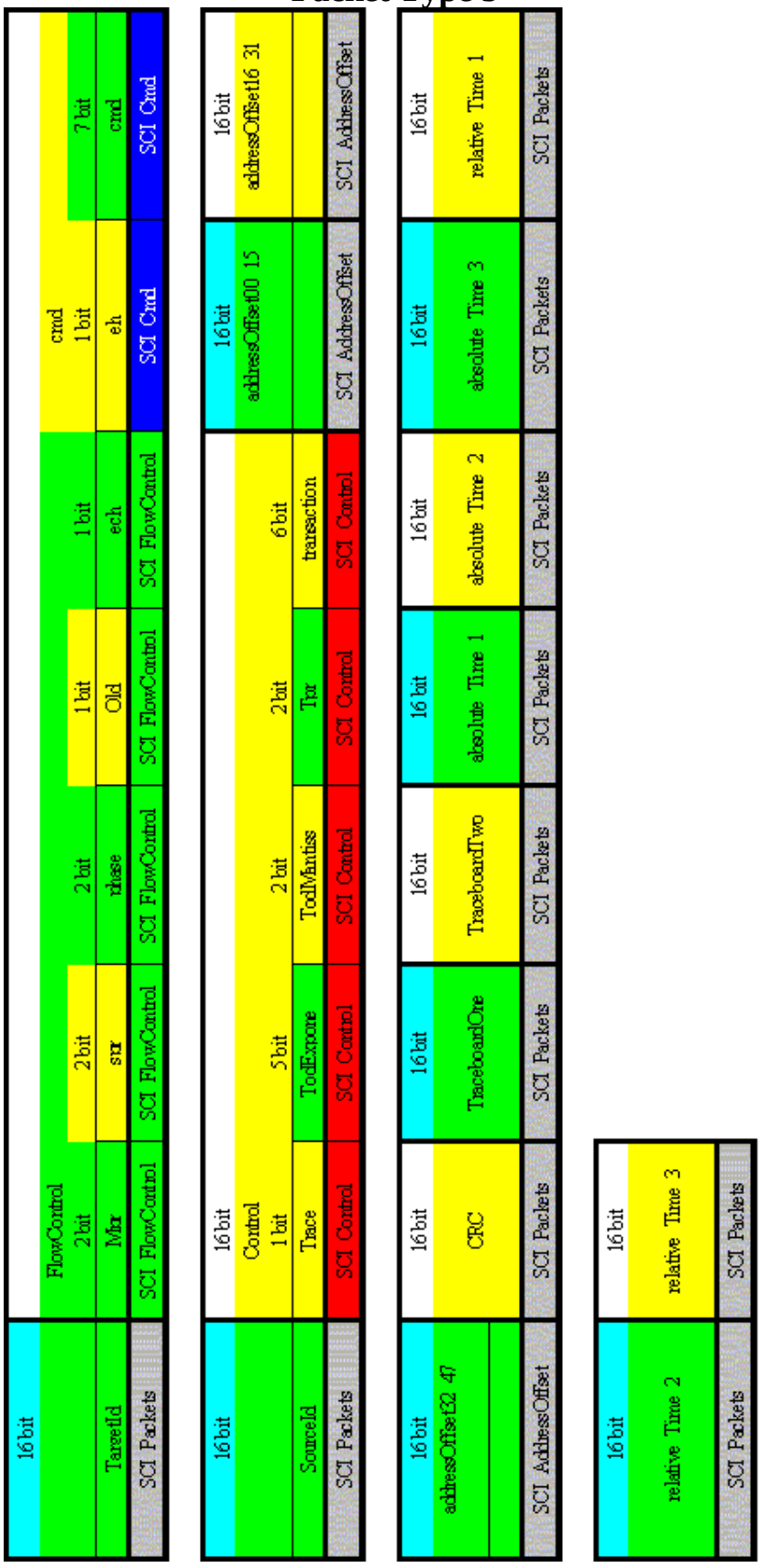
```
SELECT SCI_Packets.TraceId,
              SCI_Packets.PacketId,
              SCI_Packets.Packet_Type_Id,
              SCI_Packet_Type_Id.Packet_Type_Description,
              SCI_FlowControl.mpr,
              SCI_FlowControl.spr,
              SCI_FlowControl.phase,
              SCI_FlowControl.old,
              SCI_FlowControl.ech,
              SCI_Cmd.cmd_eh,
              SCI_Cmd.cmd_cmd,
              SCI_Packets.sourceId,
              SCI_Control.control_trace,
              SCI_Control.control_todExpone,
              SCI_Control.control_todMantiss,
              SCI_Control.control_tpr,
              SCI_Control.control_transaction,
              SCI_AddressOffset.addressOffset00_15,
              SCI_AddressOffset.addressOffset16_31,
              SCI_AddressOffset.addressOffset32_47,
              SCI_Extended.ext_00_01, SCI_Extended.ext_02_03,
              SCI_Extended.ext_04_05, SCI_Extended.ext_06_07,
              SCI_Extended.ext_08_09, SCI_Extended.ext_10_11,
              SCI_Extended.ext_12_13, SCI_Extended.ext_14_15,
              SCIData_000_015.SCIData_000_001, SCIData_000_015.SCIData_002_003,
              SCIData_000_015.SCIData_004_005, SCIData_000_015.SCIData_006_007,
              SCIData_000_015.SCIData_008_009, SCIData_000_015.SCIData_010_011,
              SCIData_000_015.SCIData_012_013, SCIData_000_015.SCIData_014_015,
              SCIData_016_063.SCIData_016_017, SCIData_016_063.SCIData_018_019,
              SCIData_016_063.SCIData_020_021, SCIData_016_063.SCIData_022_023,
              SCIData_016_063.SCIData_024_025, SCIData_016_063.SCIData_026_027,
              SCIData_016_063.SCIData_028_029, SCIData_016_063.SCIData_030_031,
              SCIData_016_063.SCIData_032_033, SCIData_016_063.SCIData_034_035,
              SCIData_016_063.SCIData_036_037, SCIData_016_063.SCIData_038_039,
              SCIData_016_063.SCIData_040_041, SCIData_016_063.SCIData_042_043,
              SCIData_016_063.SCIData_044_045, SCIData_016_063.SCIData_046_047,
              SCIData_016_063.SCIData_048_049, SCIData_016_063.SCIData_050_051,
              SCIData_016_063.SCIData_052_053, SCIData_016_063.SCIData_054_055,
              SCIData_016_063.SCIData_056_057, SCIData_016_063.SCIData_058_059,
              SCIData_016_063.SCIData_060_061, SCIData_016_063.SCIData_062_063,
              SCIData_064_255.SCIData_064_065, SCIData_064_255.SCIData_066_067,
              SCIData_064_255.SCIData_068_069, SCIData_064_255.SCIData_070_071,
              SCIData_064_255.SCIData_072_073, SCIData_064_255.SCIData_074_075,
              SCIData_064_255.SCIData_076_077, SCIData_064_255.SCIData_078_079,
              SCIData_064_255.SCIData_080_081, SCIData_064_255.SCIData_082_083,
              SCIData_064_255.SCIData_084_085, SCIData_064_255.SCIData_086_087,
              SCIData_064_255.SCIData_088_089, SCIData_064_255.SCIData_090_091,
              SCIData_064_255.SCIData_092_093, SCIData_064_255.SCIData_094_095,
              SCIData_064_255.SCIData_096_097, SCIData_064_255.SCIData_098_099,
              SCIData_064_255.SCIData_100_101, SCIData_064_255.SCIData_102_103,
              SCIData_064_255.SCIData_104_105, SCIData_064_255.SCIData_106_107,
              SCIData_064_255.SCIData_108_109, SCIData_064_255.SCIData_110_111,
              SCIData_064_255.SCIData_112_113, SCIData_064_255.SCIData_114_115,
              SCIData_064_255.SCIData_116_117, SCIData_064_255.SCIData_118_119,
              SCIData_064_255.SCIData_120_121, SCIData_064_255.SCIData_122_123,
              SCIData_064_255.SCIData_124_125, SCIData_064_255.SCIData_126_127,
              SCIData_064_255.SCIData_128_129, SCIData_064_255.SCIData_130_131,
              SCIData_064_255.SCIData_132_133, SCIData_064_255.SCIData_134_135,
              SCIData_064_255.SCIData_136_137, SCIData_064_255.SCIData_138_139,
              SCIData_064_255.SCIData_140_141, SCIData_064_255.SCIData_142_143,
              SCIData_064_255.SCIData_144_145, SCIData_064_255.SCIData_146_147,
              SCIData_064_255.SCIData_148_149, SCIData_064_255.SCIData_150_151,
              SCIData_064_255.SCIData_152_153, SCIData_064_255.SCIData_154_155,
              SCIData_064_255.SCIData_156_157, SCIData_064_255.SCIData_158_159,
              SCIData_064_255.SCIData_160_161, SCIData_064_255.SCIData_162_163,
              SCIData_064_255.SCIData_164_165, SCIData_064_255.SCIData_166_167,
              SCIData_064_255.SCIData_168_169, SCIData_064_255.SCIData_170_171,
              SCIData_064_255.SCIData_172_173, SCIData_064_255.SCIData_174_175,
              SCIData_064_255.SCIData_176_177, SCIData_064_255.SCIData_178_179,
              SCIData_064_255.SCIData_180_181, SCIData_064_255.SCIData_182_183,
              SCIData_064_255.SCIData_184_185, SCIData_064_255.SCIData_186_187,
              SCIData_064_255.SCIData_188_189, SCIData_064_255.SCIData_190_191,
              SCIData_064_255.SCIData_192_193, SCIData_064_255.SCIData_194_195,
```

See next page for SQL Query continuation

```
              SCIData_064_255.SCIData_196_197, SCIData_064_255.SCIData_198_199,
              SCIData_064_255.SCIData_200_201, SCIData_064_255.SCIData_202_203,
              SCIData_064_255.SCIData_204_205, SCIData_064_255.SCIData_206_207,
              SCIData_064_255.SCIData_208_209, SCIData_064_255.SCIData_210_211,
              SCIData_064_255.SCIData_212_213, SCIData_064_255.SCIData_214_215,
              SCIData_064_255.SCIData_216_217, SCIData_064_255.SCIData_218_219,
              SCIData_064_255.SCIData_220_221, SCIData_064_255.SCIData_222_223,
              SCIData_064_255.SCIData_224_225, SCIData_064_255.SCIData_226_227,
              SCIData_064_255.SCIData_228_229, SCIData_064_255.SCIData_230_231,
              SCIData_064_255.SCIData_232_233, SCIData_064_255.SCIData_234_235,
              SCIData_064_255.SCIData_236_237, SCIData_064_255.SCIData_238_239,
              SCIData_064_255.SCIData_240_241, SCIData_064_255.SCIData_242_243,
              SCIData_064_255.SCIData_244_245, SCIData_064_255.SCIData_246_247,
              SCIData_064_255.SCIData_248_249, SCIData_064_255.SCIData_250_251,
              SCIData_064_255.SCIData_252_253, SCIData_064_255.SCIData_254_255,
              SCI_Packets.CRC,
              SCI_Packets.traceboardOne,
              SCI_Packets.traceboardTwo,
              SCI_Packets.absolute_Time_1,
              SCI_Packets.absolute_Time_2,
              SCI_Packets.absolute_Time_3,
              SCI_Packets.relative_Time_1,
              SCI_Packets.relative_Time_2,
              SCI_Packets.relative_Time_3

FROM (((((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
        ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
              LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId = SCI_FlowControl.PacketId)
        AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId))
              LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
        AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
              LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
        AND (SCI_Packets.TraceId = SCI_Control.TraceId))
              LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId = SCI_AddressOffset.PacketId)
        AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
              LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
        AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
              LEFT JOIN SCIData_000_015 ON (SCI_Packets.PacketId = SCIData_000_015.PacketId)
        AND (SCI_Packets.TraceId = SCIData_000_015.TraceId))
              LEFT JOIN SCIData_016_063 ON (SCI_Packets.PacketId = SCIData_016_063.PacketId)
        AND (SCI_Packets.TraceId = SCIData_016_063.TraceId))
              LEFT JOIN SCIData_064_255 ON (SCI_Packets.PacketId = SCIData_064_255.PacketId)
        AND (SCI_Packets.TraceId = SCIData_064_255.TraceId)

WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3)
       AND ((SCI_Packets.Packet_Type_Id) Like 4));
```

Appendix D: Request-send-packet with 0 bytes data

Packet Type 5

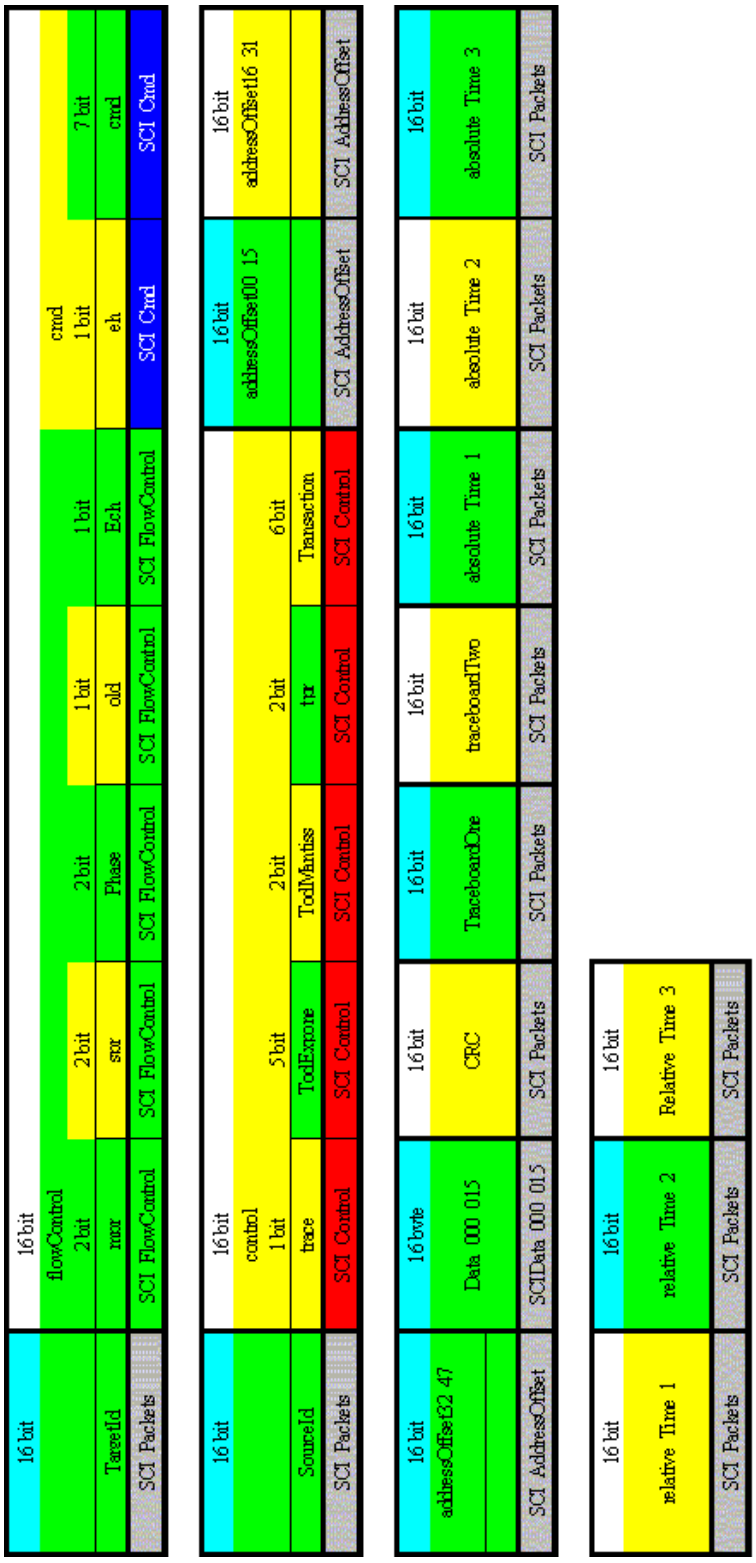


The following SQL Query retrieves all information related to a ***SCI packet type 5 Request-send-packet with 0 bytes data***. In this instance form a SCI packet with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
              SCI_Packets.PacketId,
              SCI_Packets.Packet_Type_Id,
              SCI_Packet_Type_Id.Packet_Type_Description,
              SCI_FlowControl.mpr,
              SCI_FlowControl.spr,
              SCI_FlowControl.phase,
              SCI_FlowControl.old,
              SCI_FlowControl.ech,
              SCI_Cmd.cmd_eh,
              SCI_Cmd.cmd_cmd,
              SCI_Packets.sourceId,
              SCI_Control.control_trace,
              SCI_Control.control_todExpone,
              SCI_Control.control_todMantiss,
              SCI_Control.control_tpr,
              SCI_Control.control_transaction,
              SCI_AddressOffset.addressOffset00_15,
              SCI_AddressOffset.addressOffset16_31,
              SCI_AddressOffset.addressOffset32_47,
              SCI_Packets.CRC,
              SCI_Packets.traceboardOne,
              SCI_Packets.traceboardTwo,
              SCI_Packets.absolute_Time_1,
              SCI_Packets.absolute_Time_2,
              SCI_Packets.absolute_Time_3,
              SCI_Packets.relative_Time_1,
              SCI_Packets.relative_Time_2,
              SCI_Packets.relative_Time_3

FROM ((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
        ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
        AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
        AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
        AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId = SCI_FlowControl.PacketId)
        AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)

WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3)
       AND ((SCI_Packets.Packet_Type_Id) Like 5));
```

Appendix D: Request-send-packet with 16 bytes data

Packet Type 6

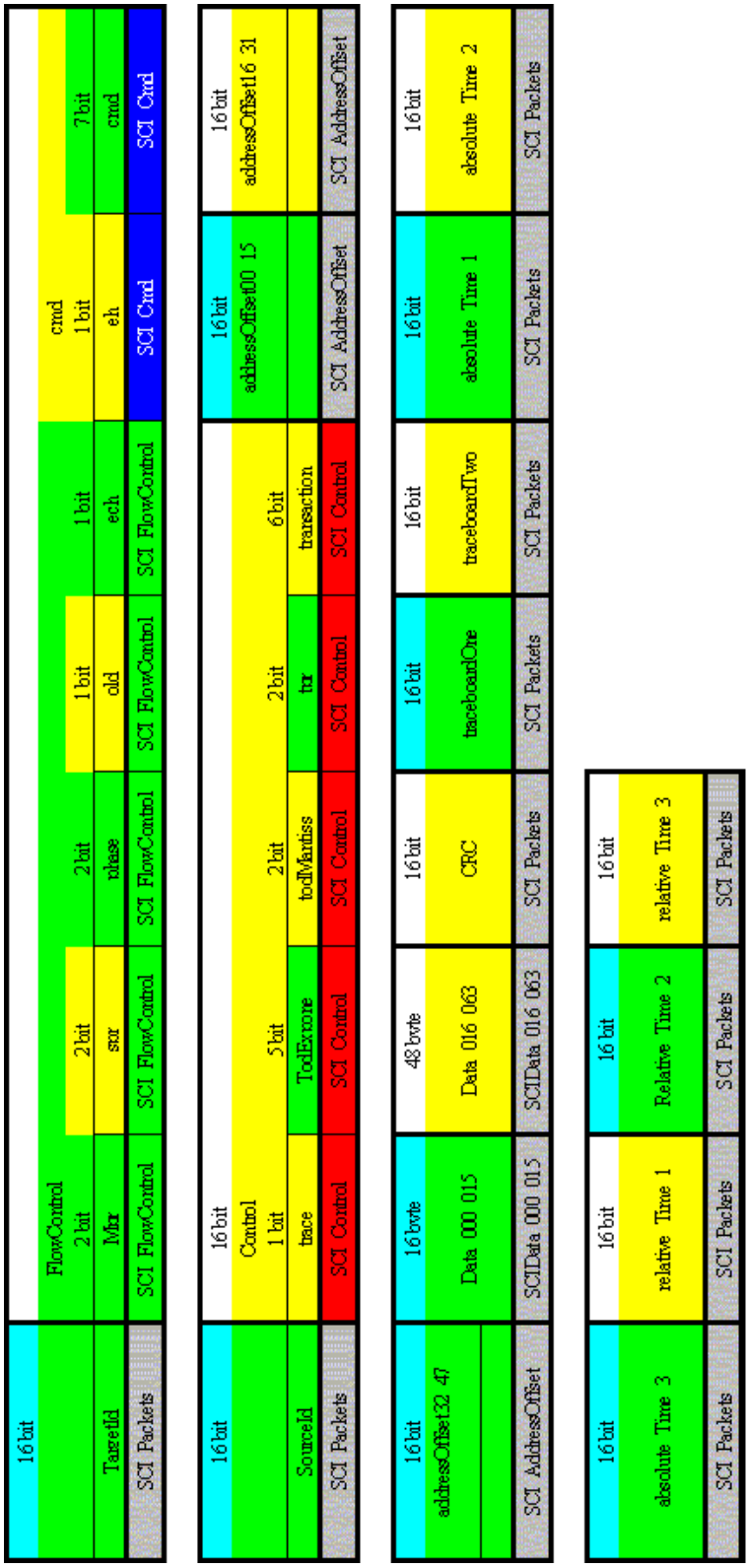


The following SQL Query retrieves all information related to a ***SCI packet type 6 Request-send-packet with 16 bytes data***. In this instance form a SCI packet with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCIData_000_015.SCIData_000_001, SCIData_000_015.SCIData_002_003,
             SCIData_000_015.SCIData_004_005, SCIData_000_015.SCIData_006_007,
             SCIData_000_015.SCIData_008_009, SCIData_000_015.SCIData_010_011,
             SCIData_000_015.SCIData_012_013, SCIData_000_015.SCIData_014_015,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3

FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
        ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
              LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
        AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
              LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
        AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
              LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
        AND (SCI_Packets.TraceId = SCI_Control.TraceId))
              LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId = SCI_FlowControl.PacketId)
        AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId))
              LEFT JOIN SCIData_000_015 ON (SCI_Packets.PacketId = SCIData_000_015.PacketId)
        AND (SCI_Packets.TraceId = SCIData_000_015.TraceId)

WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3)
       AND ((SCI_Packets.Packet_Type_Id) Like 6));
```

Appendix D: Request-send-packet with 64 bytes data

Packet Type 7

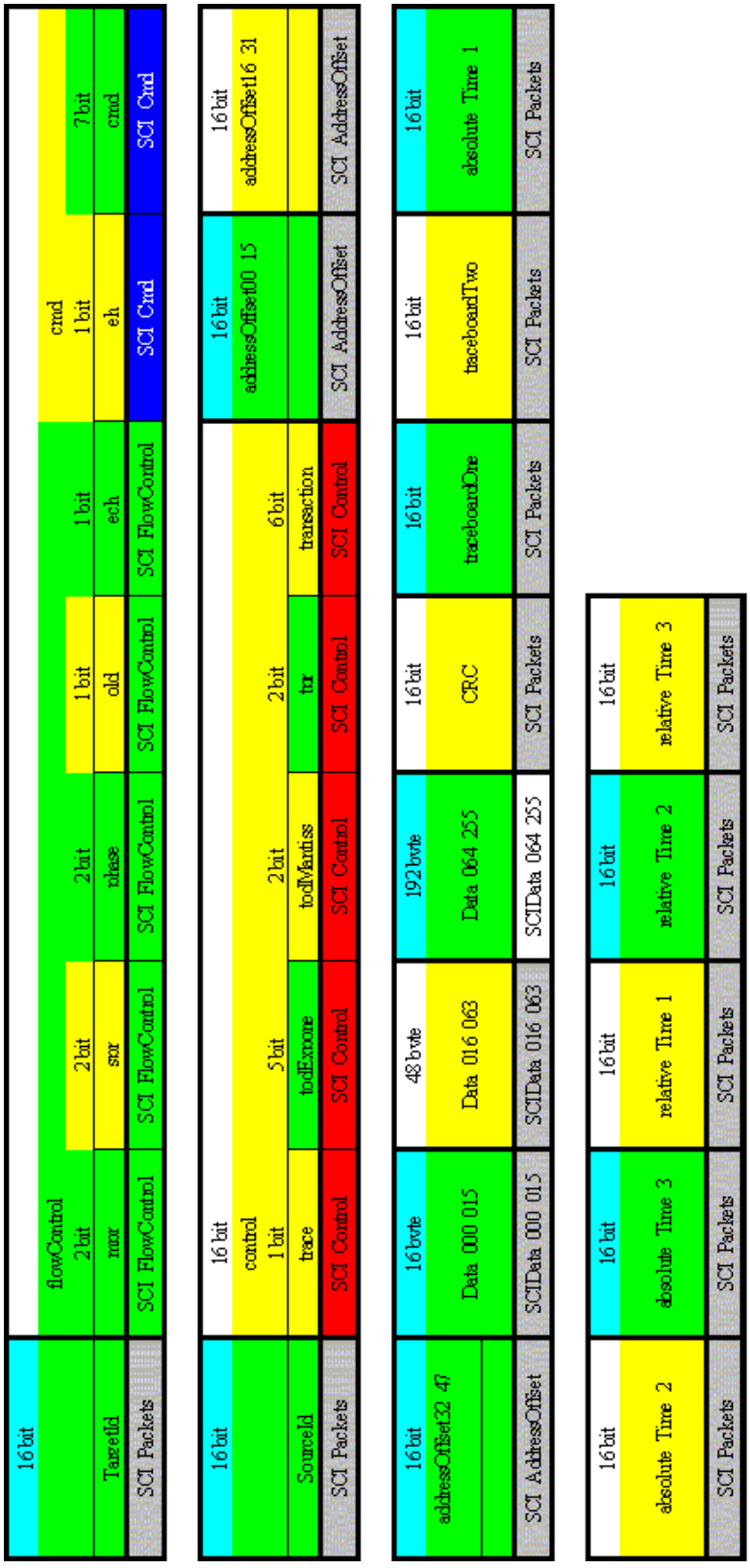


The following SQL Query retrieves all information related to a ***SCI packet type 7 Request-send-packet with 64 bytes data***. In this instance form a SCI packet with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCIData_000_015.SCIData_000_001, SCIData_000_015.SCIData_002_003,
             SCIData_000_015.SCIData_004_005, SCIData_000_015.SCIData_006_007,
             SCIData_000_015.SCIData_008_009, SCIData_000_015.SCIData_010_011,
             SCIData_000_015.SCIData_012_013, SCIData_000_015.SCIData_014_015,
             SCIData_016_063.SCIData_016_017, SCIData_016_063.SCIData_018_019,
             SCIData_016_063.SCIData_020_021, SCIData_016_063.SCIData_022_023,
             SCIData_016_063.SCIData_024_025, SCIData_016_063.SCIData_026_027,
             SCIData_016_063.SCIData_028_029, SCIData_016_063.SCIData_030_031,
             SCIData_016_063.SCIData_032_033, SCIData_016_063.SCIData_034_035,
             SCIData_016_063.SCIData_036_037, SCIData_016_063.SCIData_038_039,
             SCIData_016_063.SCIData_040_041, SCIData_016_063.SCIData_042_043,
             SCIData_016_063.SCIData_044_045, SCIData_016_063.SCIData_046_047,
             SCIData_016_063.SCIData_048_049, SCIData_016_063.SCIData_050_051,
             SCIData_016_063.SCIData_052_053, SCIData_016_063.SCIData_054_055,
             SCIData_016_063.SCIData_056_057, SCIData_016_063.SCIData_058_059,
             SCIData_016_063.SCIData_060_061, SCIData_016_063.SCIData_062_063,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3

FROM ((((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
               LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId = SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
               LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
               LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
               LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId = SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId))
               LEFT JOIN SCIData_000_015 ON (SCI_Packets.PacketId = SCIData_000_015.PacketId)
         AND (SCI_Packets.TraceId = SCIData_000_015.TraceId))
               LEFT JOIN SCIData_016_063 ON (SCI_Packets.PacketId = SCIData_016_063.PacketId)
         AND (SCI_Packets.TraceId = SCIData_016_063.TraceId)

WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3)
       AND ((SCI_Packets.Packet_Type_Id) Like 7));
```

Appendix D: Request-send-packet with 256 bytes data

Packet Type 8

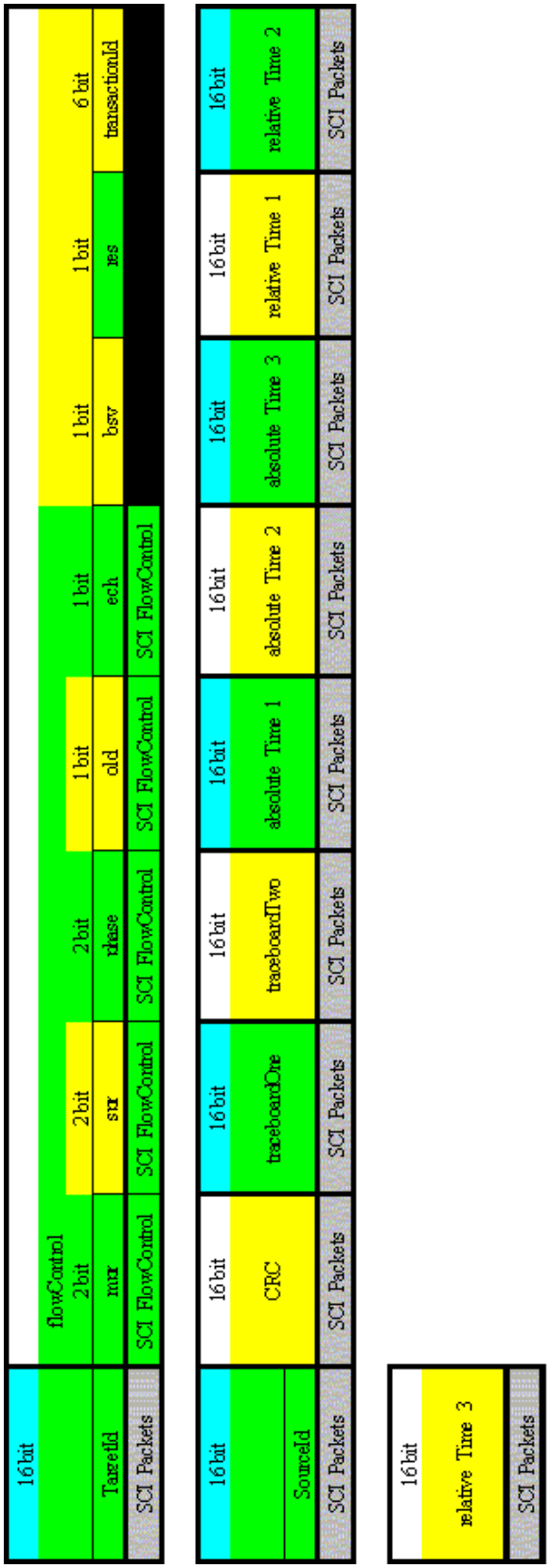


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Request-echo-packet

Packet Type 9

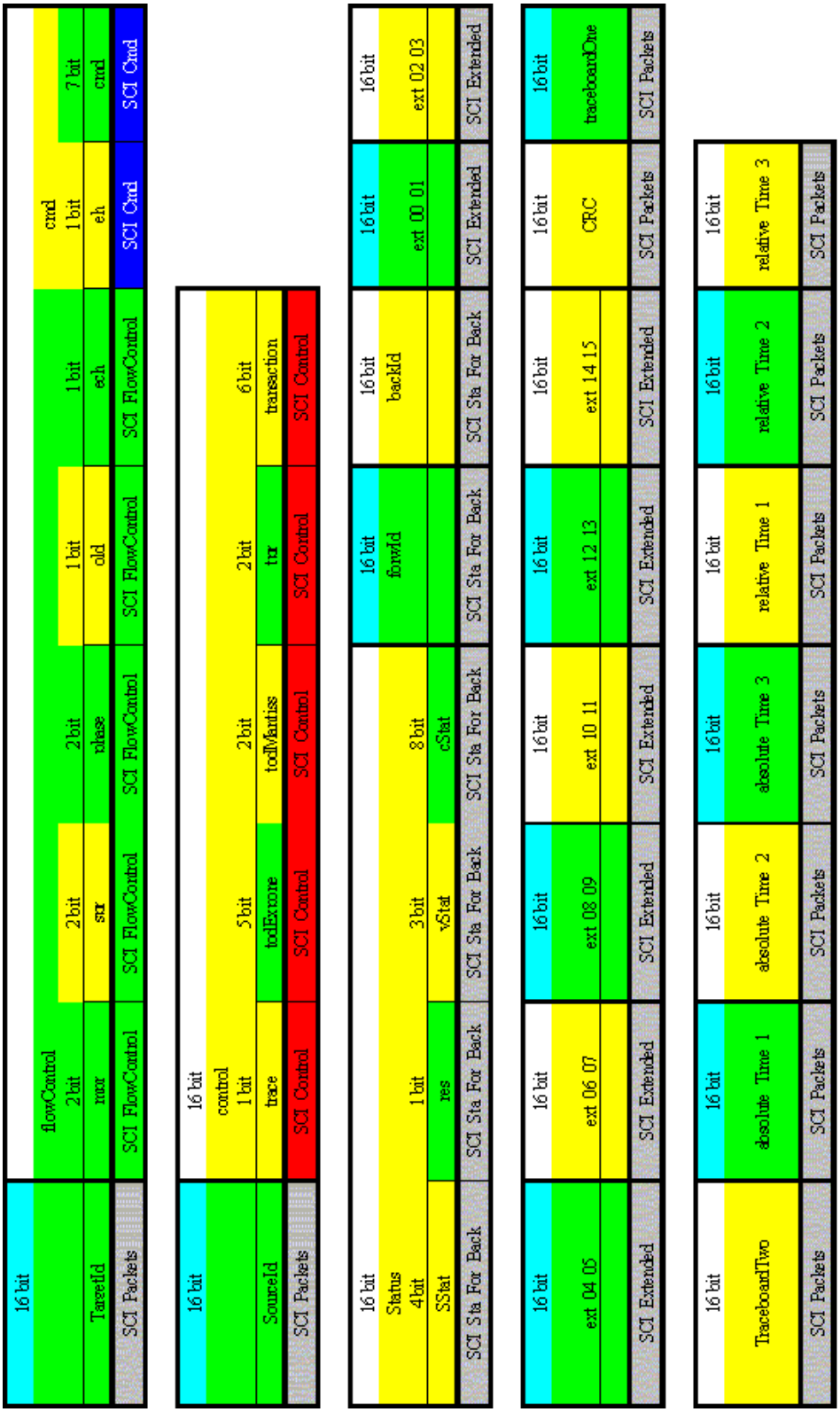

16 bit
TargetId
SCI Packets
flowControl
2 bit
mar
SCI FlowControl
2 bit
str
SCI FlowControl
2 bit
rdata
SCI FlowControl
1 bit
old
SCI FlowControl
1 bit
ech
SCI FlowControl
1 bit
bsy
1 bit
res
6 bit
transactionId
16 bit
SourceId
SCI Packets
16 bit
CRC
SCI Packets
16 bit
traceboardOne
SCI Packets
16 bit
traceboardTwo
SCI Packets
16 bit
absolute Time 1
SCI Packets
16 bit
absolute Time 2
SCI Packets
16 bit
absolute Time 3
SCI Packets
16 bit
relative Time 1
SCI Packets
16 bit
relative Time 2
SCI Packets
16 bit
relative Time 3
SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Response-send-packet with extended header and 0 bytes data

Packet Type 10

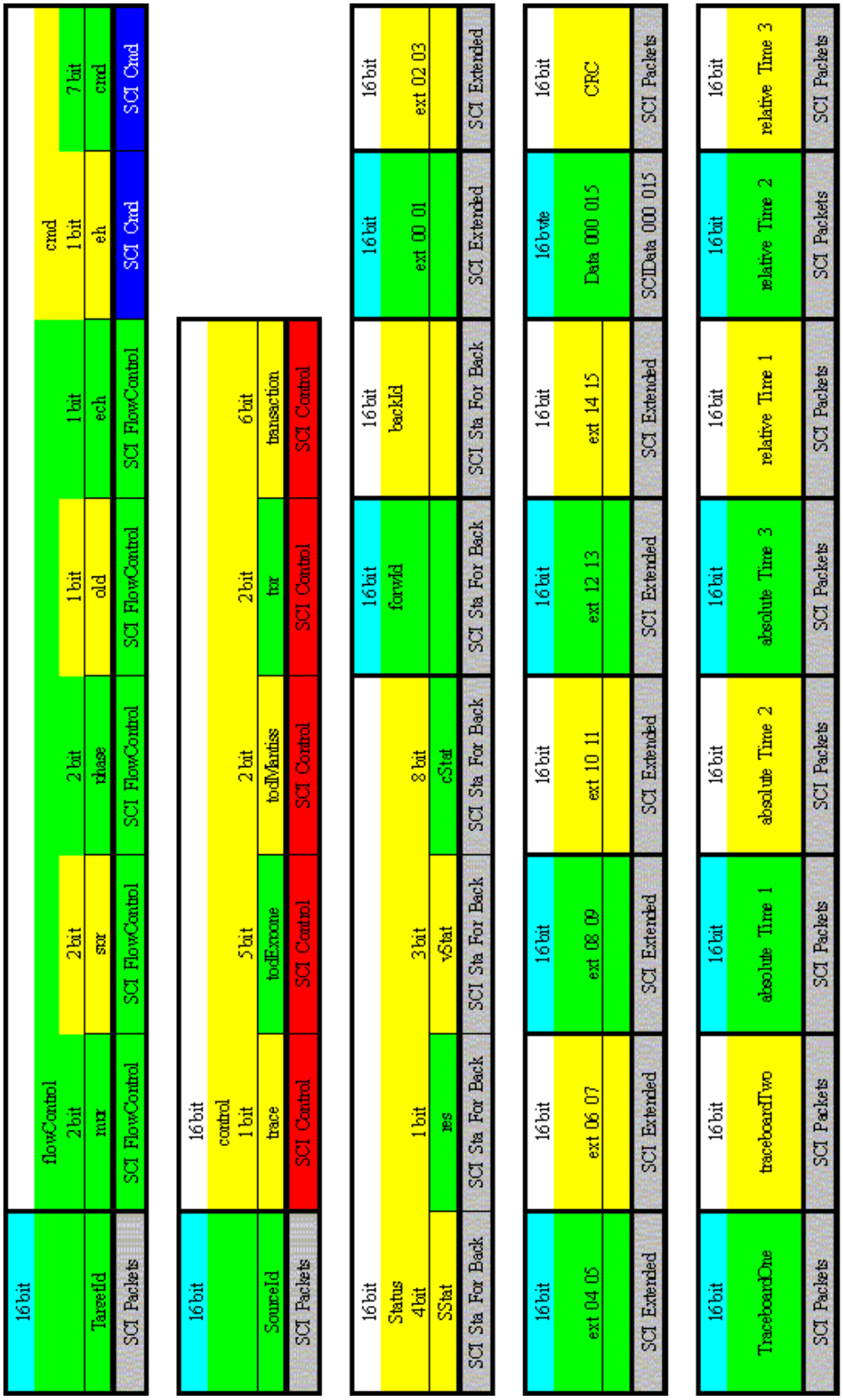


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Response-send-packet with extended header and 16 bytes data

Packet Type 11

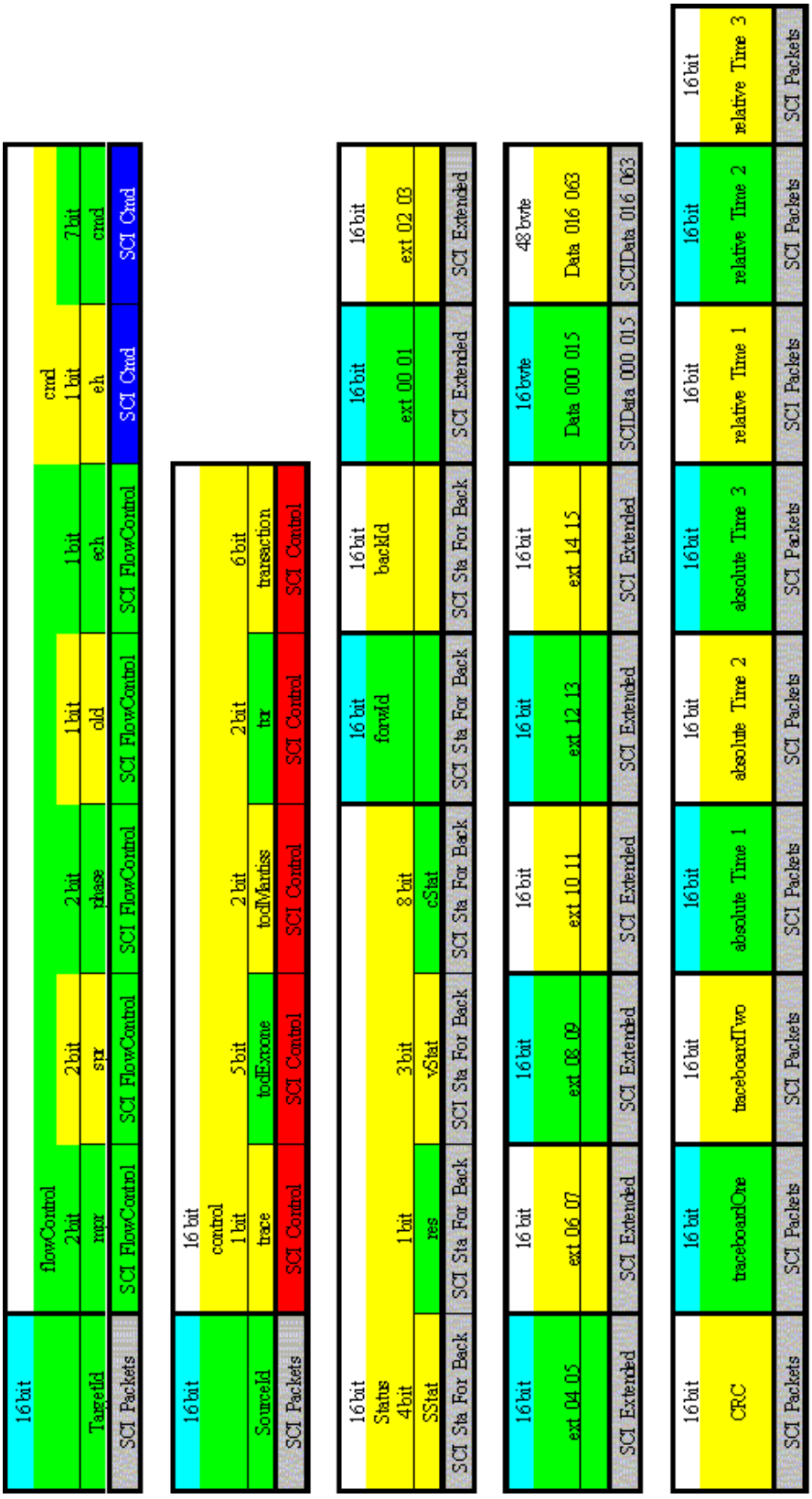

16 bit
flowControl 2 bit
2 bit
2 bit
1 bit
1 bit
cmd 1 bit
7 bit
TaredId
mar
sar
phase
old
ech
eh
cmd
SCI Packets
SCI FlowControl
SCI FlowControl
SCI FlowControl
SCI FlowControl
SCI Cmd
SCI Cmd
16 bit
control 1 bit
5 bit
2 bit
2 bit
6 bit
SourceId
trace
todExcome
todMantiss
bar
transaction
SCI Packets
SCI Control
SCI Control
SCI Control
SCI Control
16 bit
Status 4bit
1 bit
3 bit
8 bit
16 bit
forwId
16 bit
backId
16 bit
ext 00 01
16 bit
ext 02 03
SStat
res
vStat
cStat
SCI Sta For Back
SCI Sta For Back
SCI Sta For Back
SCI Sta For Back
SCI Sta For Back
SCI Extended
SCI Extended
16 bit
16 bit
16 bit
16 bit
16 bit
16 byte
16 bit
CRC
ext 04 05
ext 06 07
ext 08 09
ext 10 11
ext 12 13
ext 14 15
Data 000 015
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCIData 000 015
SCI Packets
16 bit
16 bit
16 bit
16 bit
16 bit
16 bit
16 bit
TraceboardOne
traceboardTwo
absolute Time 1
absolute Time 2
absolute Time 3
relative Time 1
relative Time 2
relative Time 3
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Response-send-packet with extended header and 64 bytes data

Packet Type 12

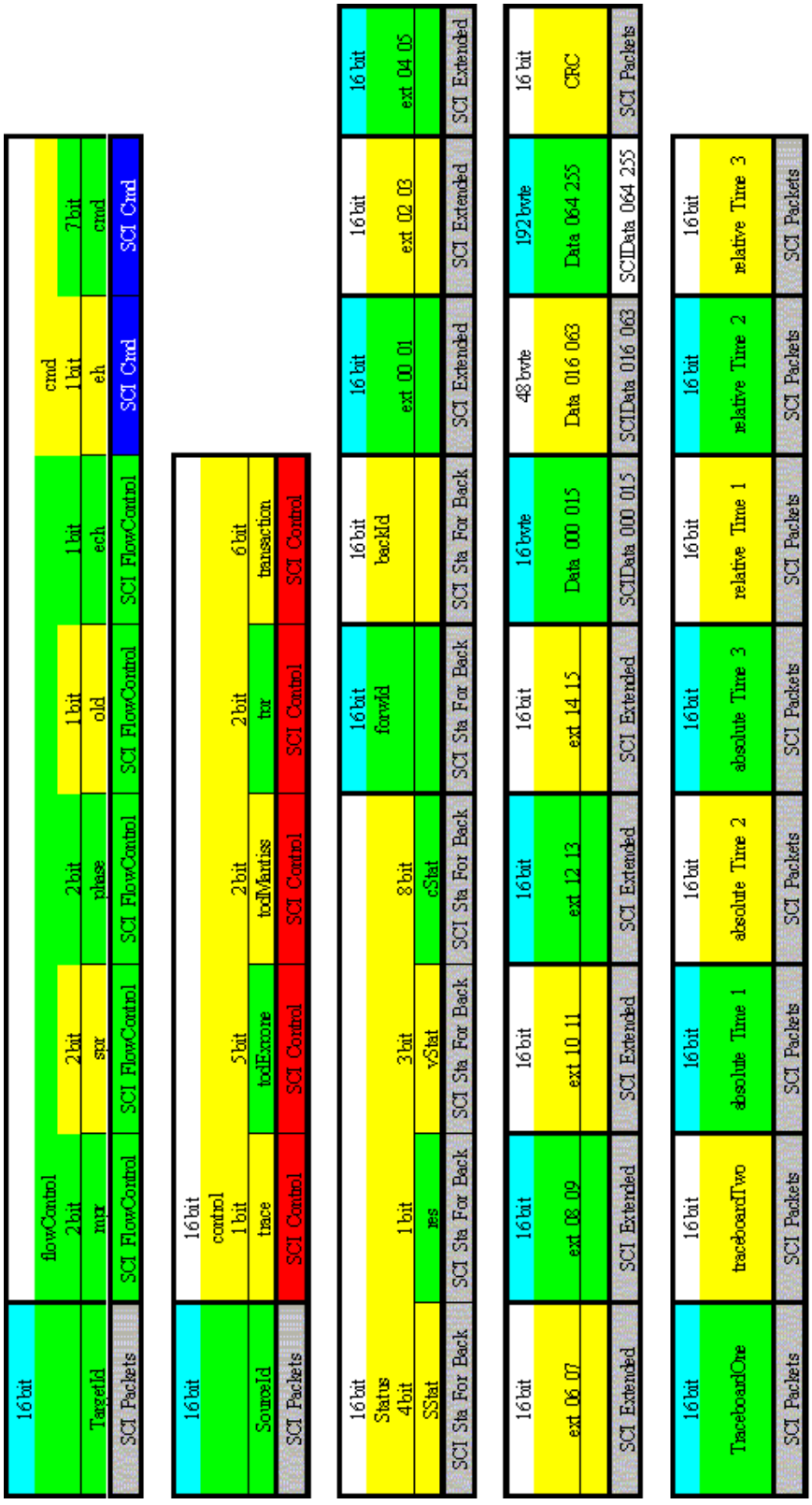

16 bit
flowControl 2 bit
2 bit
2 bit
1 bit
1 bit
cmd 1 bit
7 bit
TargetId
mor
str
phase
old
eh
eh
cmd
SCI Packets
SCI FlowControl
SCI FlowControl
SCI FlowControl
SCI FlowControl
SCI Cmd
SCI Cmd
16 bit
control 1 bit
5 bit
2 bit
2 bit
6 bit
SourceId
trace
bdExcone
bdMantiss
bar
transaction
SCI Packets
SCI Control
SCI Control
SCI Control
SCI Control
16 bit
Status 4 bit
1 bit
3 bit
8 bit
16 bit
forwId
16 bit
backId
16 bit
ext 02 03
SStat
res
vStat
cStat
ext 00 01
SCI Sta For Back
SCI Sta For Back
SCI Sta For Back
SCI Sta For Back
SCI Sta For Back
SCI Extended
SCI Extended
16 bit
16 bit
16 bit
16 bit
16 bit
16 bit
48 byte
ext 04 05
ext 06 07
ext 08 09
ext 10 11
ext 12 13
ext 14 15
Data 016 063
Data 000 015
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCIData 000 015
SCIData 016 063
16 bit
16 bit
16 bit
16 bit
16 bit
16 bit
16 bit
CRC
traceboardOne
traceboardTwo
absolute Time 1
absolute Time 2
absolute Time 3
relative Time 1
relative Time 2
relative Time 3
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Response-send-packet with extended header and 256 bytes data

Packet Type 13

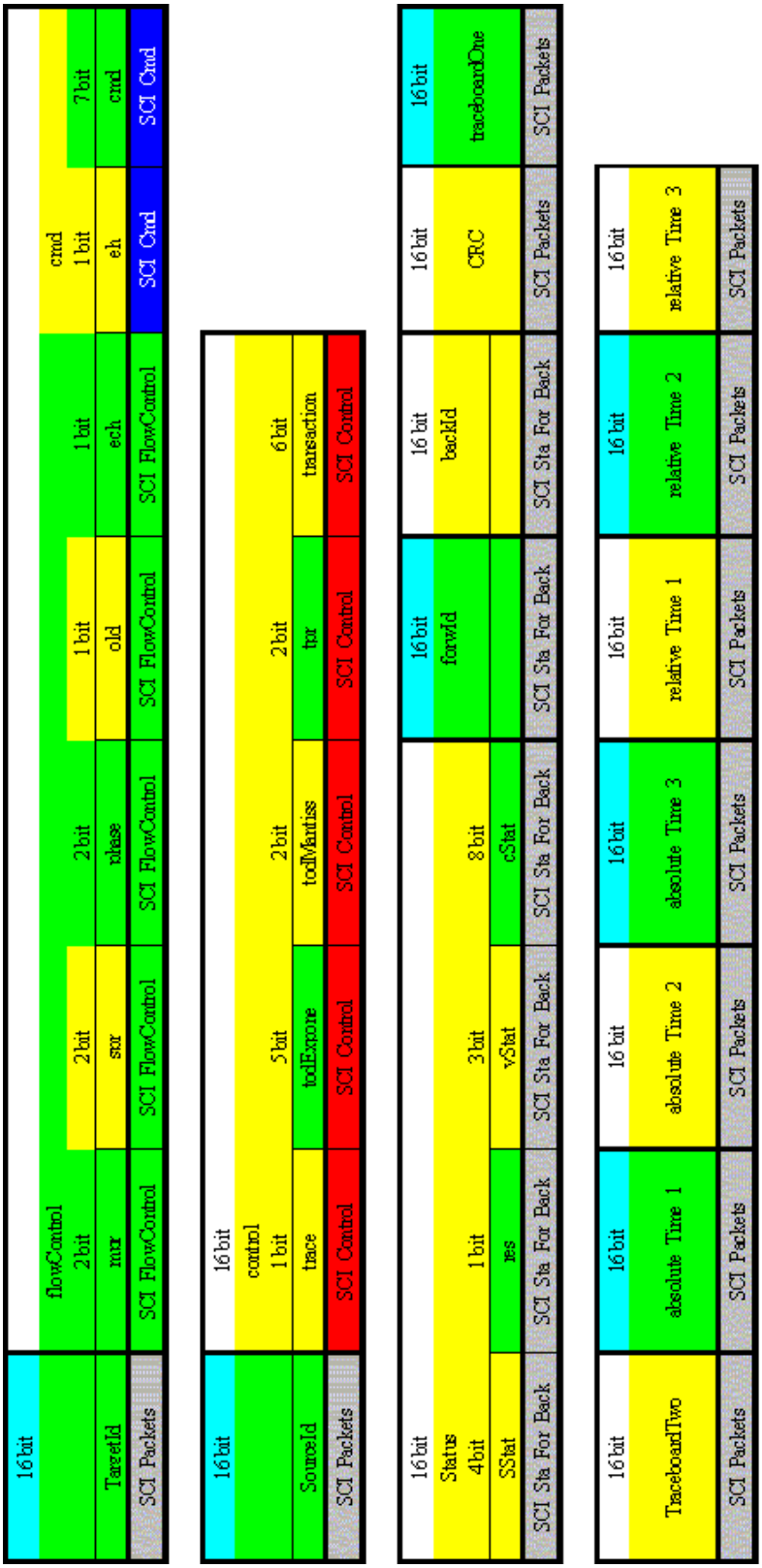


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Response-send-packet with 0 bytes data

Packet Type 14

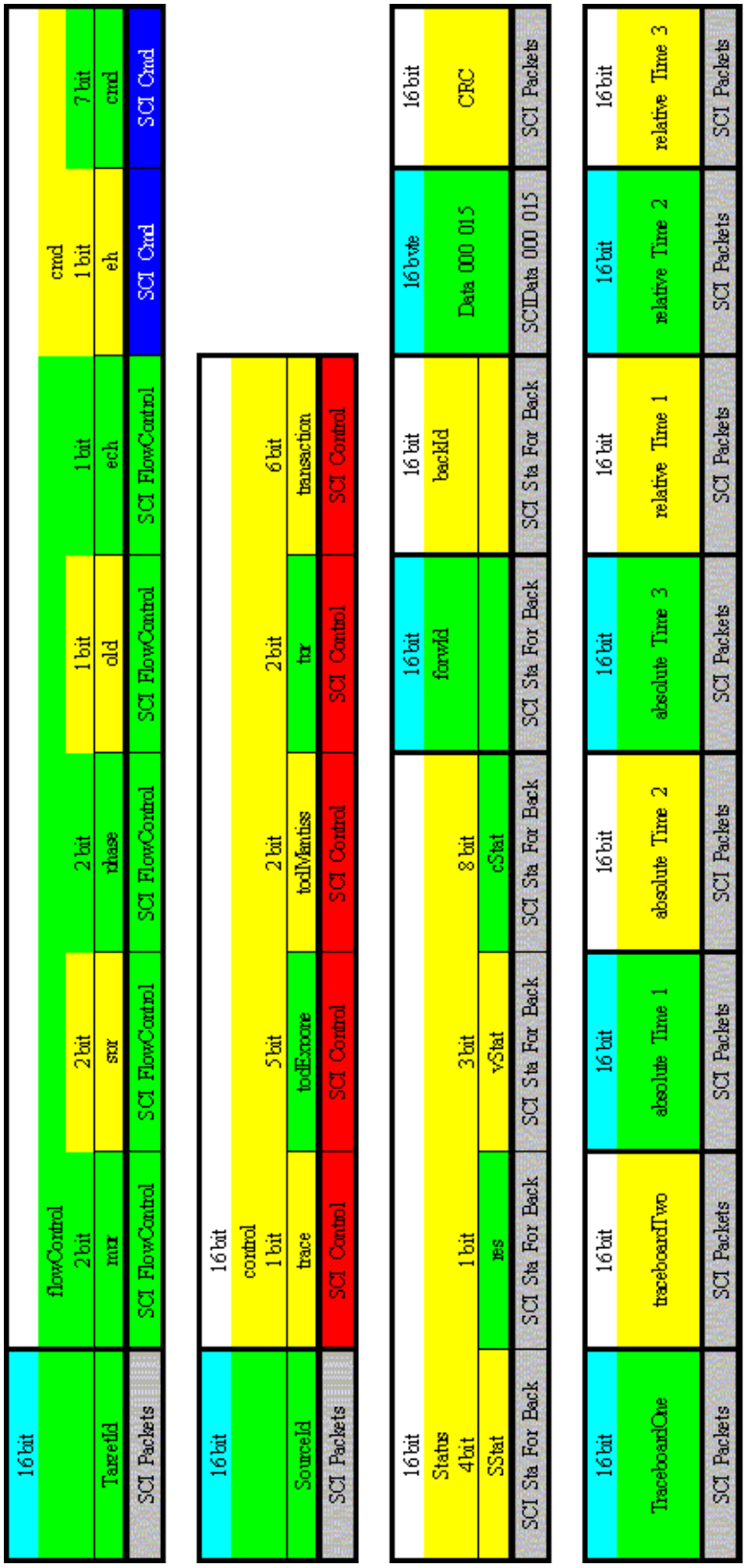


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Response-send-packet with 16 bytes data

Packet Type 15

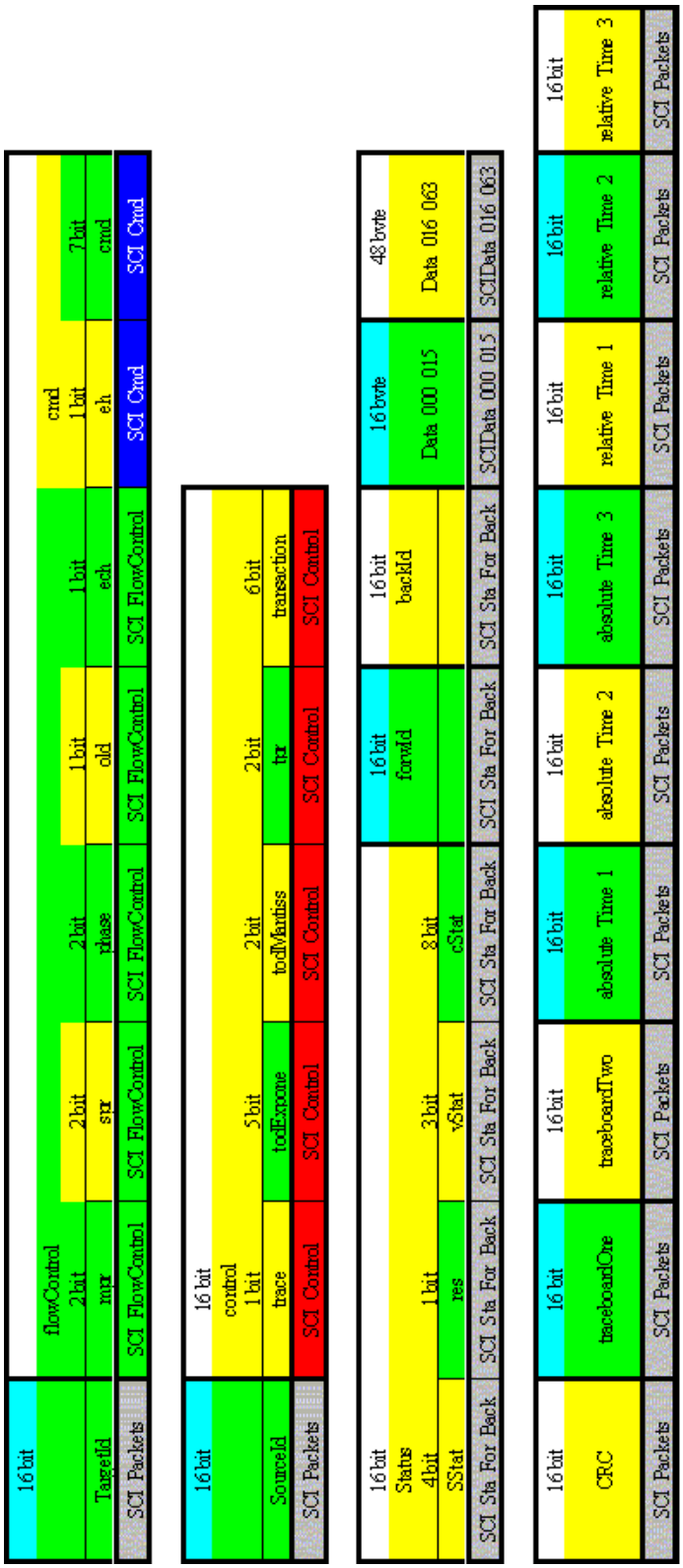


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Response-send-packet with 64 bytes data

Packet Type 16



The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Response-send-packet with 256 bytes data

Packet Type 17

16 bit	16 bit flowControl 2 bit	2 bit	2 bit	1 bit	1 bit	cmd 1 bit	7 bit
TargetId	mmr	spr	phase	old	ech	eh	cmd
SCI Packets	SCI FlowControl	SCI FlowControl	SCI FlowControl	SCI FlowControl	SCI FlowControl	SCI Cmd	SCI Cmd

16 bit	16 bit control 1 bit	5 bit	2 bit	2 bit	6 bit
SourceId	trace	todExpone	todMantiss	tpr	transaction
SCI Packets	SCI Control	SCI Control	SCI Control	SCI Control	SCI Control

16 bit Status 4 bit	1 bit	3 bit	8 bit	16 bit	16 bit	16 byte	48 byte
SStat	res	vStat	cStat	forwId	backId	Data 000 015	Data 016 063
SCI Sta For Back	SCI Sta For Back	SCI Sta For Back	SCI Sta For Back	SCI Sta For Back	SCI Sta For Back	SCIData 000 015	SCIData 016 063

192 byte	16 bit	16 bit	16 bit	16 bit	16 bit	16 bit	16 bit
Data 064 255	CRC	traceboardOne	traceboardTwo	absolute Time 1	absolute Time 2	absolute Time 3	relative Time 1
SCIData 064 255	SCI Packets	SCI Packets	SCI Packets	SCI Packets	SCI Packets	SCI Packets	SCI Packets

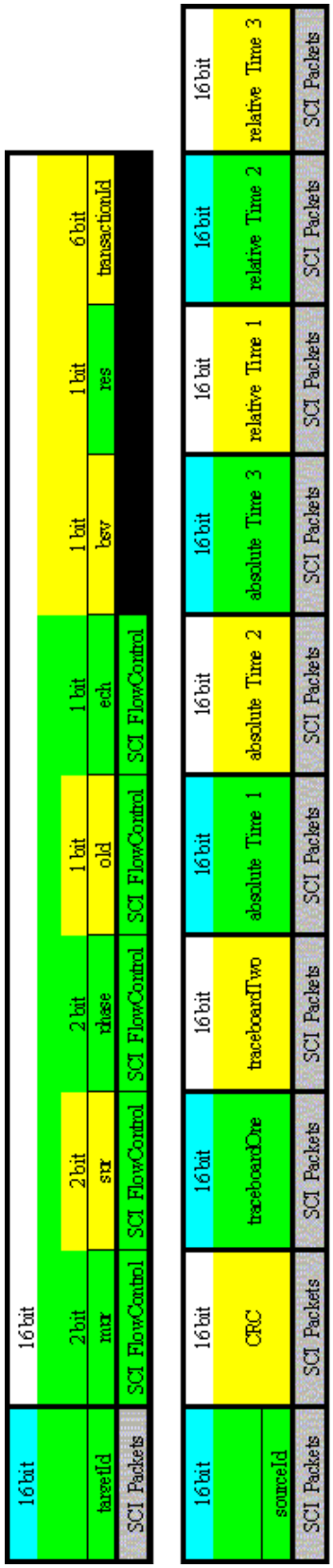
16 bit	16 bit
relative Time 2	relative Time 3
SCI Packets	SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Response-echo-packet

Packet Type 18

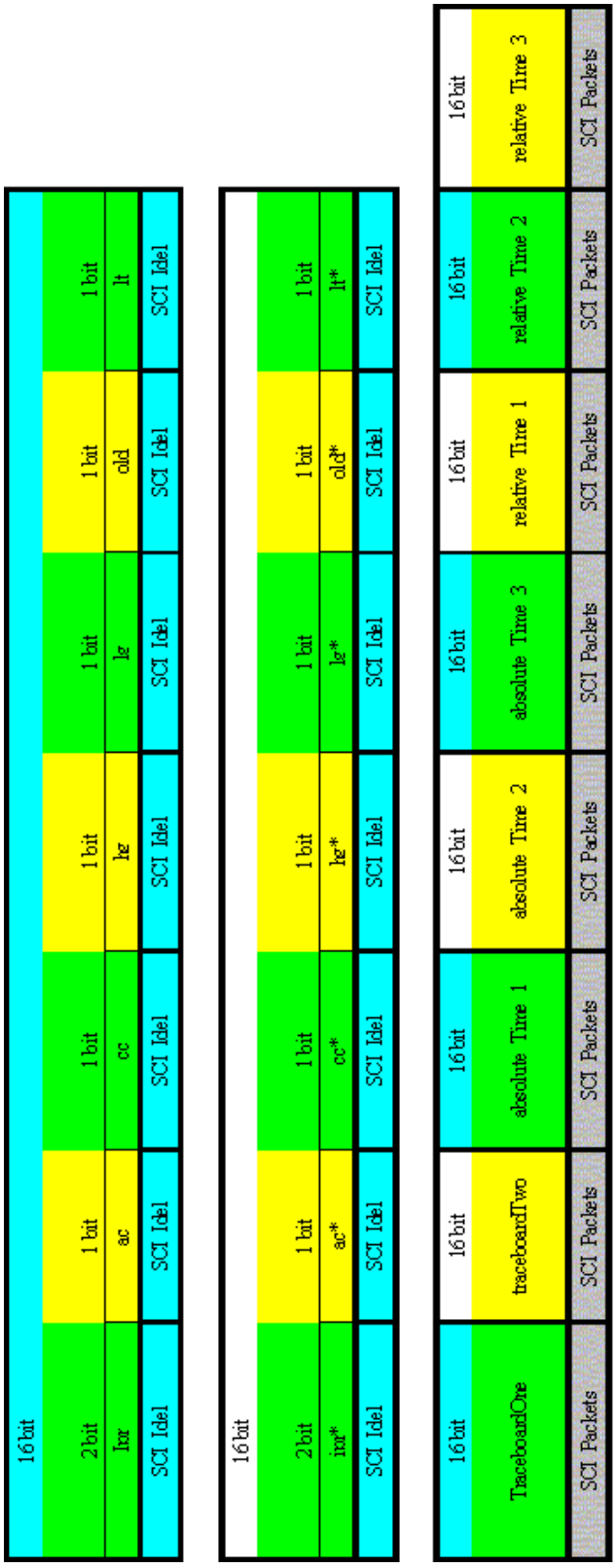

16 bit
2 bit
targetId
mar
SCI Packets
SCI FlowControl
16 bit
2 bit
2 bit
1 bit
1 bit
1 bit
6 bit
sar
phase
old
ech
bsy
res
transactionId
SCI FlowControl
SCI FlowControl
SCI FlowControl
16 bit
16 bit
16 bit
16 bit
16 bit
16 bit
16 bit
sourceId
CRC
traceboardOne
traceboardTwo
absolute Time 1
absolute Time 2
absolute Time 3
relative Time 1
relative Time 2
relative Time 3
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Idle Symbols

Packet Type 19


16 bit
2 bit
Ixr
1 bit
ac
1 bit
cc
1 bit
kr
1 bit
kz
1 bit
old
1 bit
lt
SCI Idel
SCI Idel
SCI Idel
SCI Idel
SCI Idel
SCI Idel
16 bit
2 bit
ixr*
1 bit
ac*
1 bit
cc*
1 bit
kr*
1 bit
kz*
1 bit
old*
1 bit
lt*
SCI Idel
SCI Idel
SCI Idel
SCI Idel
SCI Idel
SCI Idel
16 bit
TraceboardOne
16 bit
traceboardTwo
16 bit
absolute Time 1
16 bit
absolute Time 2
16 bit
absolute Time 3
16 bit
relative Time 1
16 bit
relative Time 2
16 bit
relative Time 3
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Sync packets

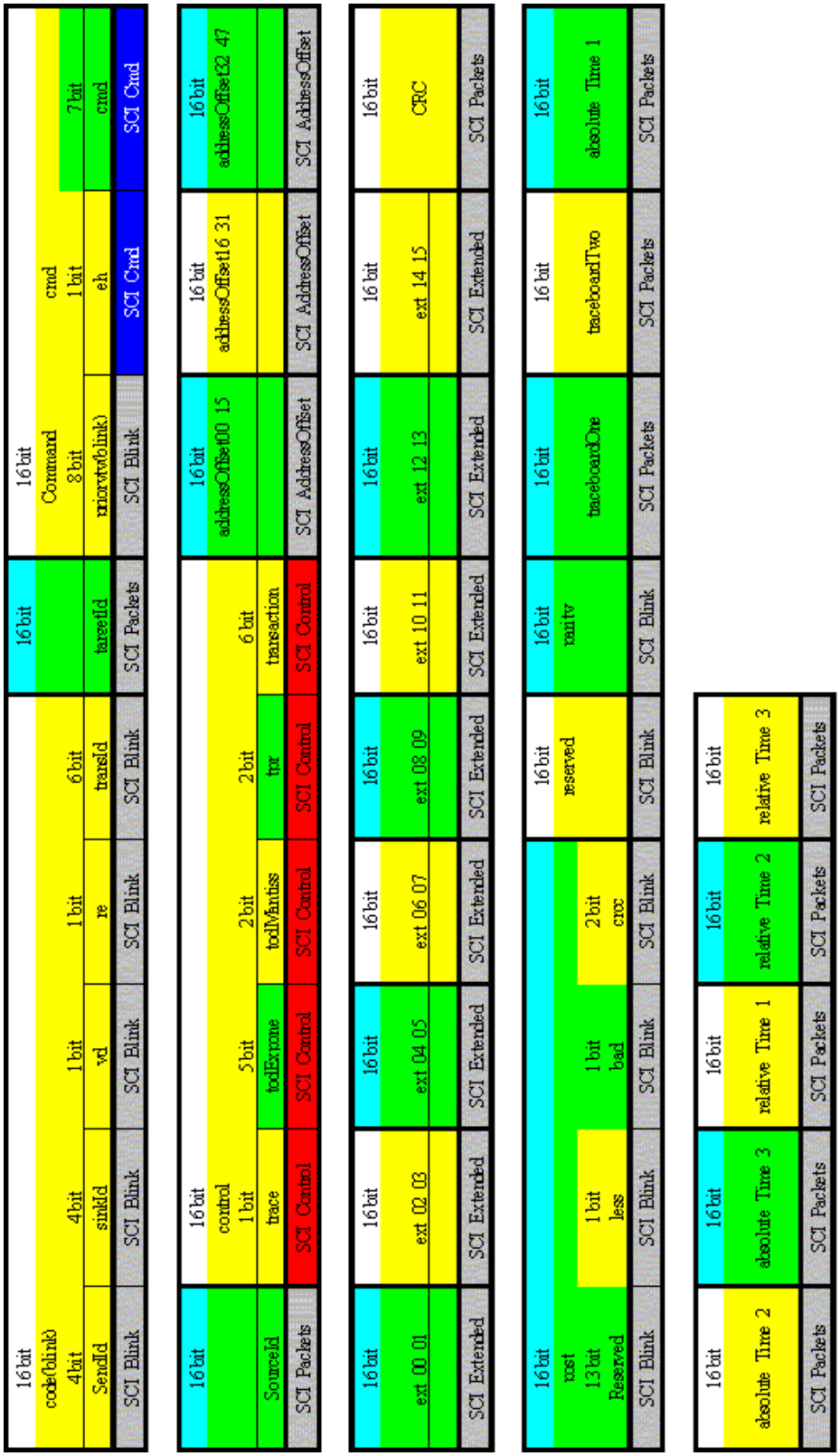
Packet Type 20

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated request-send-packet with extended header and 0 bytes data

Packet Type 21


16 bit
code(blink)
4 bit
SendId
4 bit
1 bit
sinkId
vd
1 bit
re
6 bit
transId
16 bit
Command
8 bit
targetId
cmd
1 bit
eh
7 bit
cmd
SCI Blink
SCI Blink
SCI Blink
SCI Packets
SCI Blink
SCI Cmd
SCI Cmd
16 bit
control
1 bit
SourceId
trace
SCI Packets
SCI Control
5 bit
toolExcone
SCI Control
2 bit
toolVarbiss
SCI Control
2 bit
tar
SCI Control
6 bit
transaction
SCI Control
16 bit
addressOffset00 15
SCI AddressOffset
16 bit
addressOffset16 31
SCI AddressOffset
16 bit
addressOffset32 47
SCI AddressOffset
16 bit
ext 00 01
SCI Extended
16 bit
ext 02 03
SCI Extended
16 bit
ext 04 05
SCI Extended
16 bit
ext 06 07
SCI Extended
16 bit
ext 08 09
SCI Extended
16 bit
ext 10 11
SCI Extended
16 bit
ext 12 13
SCI Extended
16 bit
ext 14 15
SCI Extended
16 bit
CRC
SCI Packets
16 bit
root
13 bit
Reserved
SCI Blink
1 bit
less
1 bit
bad
SCI Blink
2 bit
crcc
SCI Blink
16 bit
reserved
SCI Blink
16 bit
vaaitv
SCI Blink
16 bit
traceboardOne
SCI Packets
16 bit
traceboardTwo
SCI Packets
16 bit
absolute Time 1
SCI Packets
16 bit
absolute Time 2
SCI Packets
16 bit
absolute Time 3
SCI Packets
16 bit
relative Time 1
SCI Packets
16 bit
relative Time 2
SCI Packets
16 bit
relative Time 3
SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated request-send-packet with extended header and 16 bytes data

Packet Type 22

16 bit code(blink)					16 bit	16 bit Command	cmd	
4 bit	4 bit	1 bit	1 bit	6 bit		8 bit	1 bit	7 bit
sendId	sinkId	vc	re	transId	targetId	priority(blink)	eh	cmd
SCI Blink	SCI Blink	SCI Blink	SCI Blink	SCI Blink	SCI Packets	SCI Blink	SCI Cmd	SCI Cmd

16 bit	16 bit control					16 bit	16 bit	16 bit
	1 bit	5 bit	2 bit	2 bit	6 bit	addressOffset00 15	addressOffset16 31	addressOffset32 47
sourceId	trace	todExpone	todMantiss	tpr	transaction			
SCI Packets	SCI Control	SCI Control	SCI Control	SCI Control	SCI Control	SCI AddressOffset	SCI AddressOffset	SCI AddressOffset

16 bit	16 bit	16 bit	16 bit	16 bit	16 bit	16 bit	16 bit	16 byte
ext 00 01	ext 02 03	ext 04 05	ext 06 07	ext 08 09	ext 10 11	ext 12 13	ext 14 15	Data 000 015
SCI Extended	SCI Extended	SCI Extended	SCI Extended	SCI Extended	SCI Extended	SCI Extended	SCI Extended	SCIData 000 015

16 bit	16 bit post				16 bit	16 bit	16 bit	16 bit
CRC	13 bit reserved	1 bit less	1 bit bad	2 bit crcc	reserved	parity	traceboardOne	traceboardTwo
SCI Packets	SCI Blink	SCI Blink	SCI Blink	SCI Blink	SCI Blink	SCI Blink	SCI Packets	SCI Packets

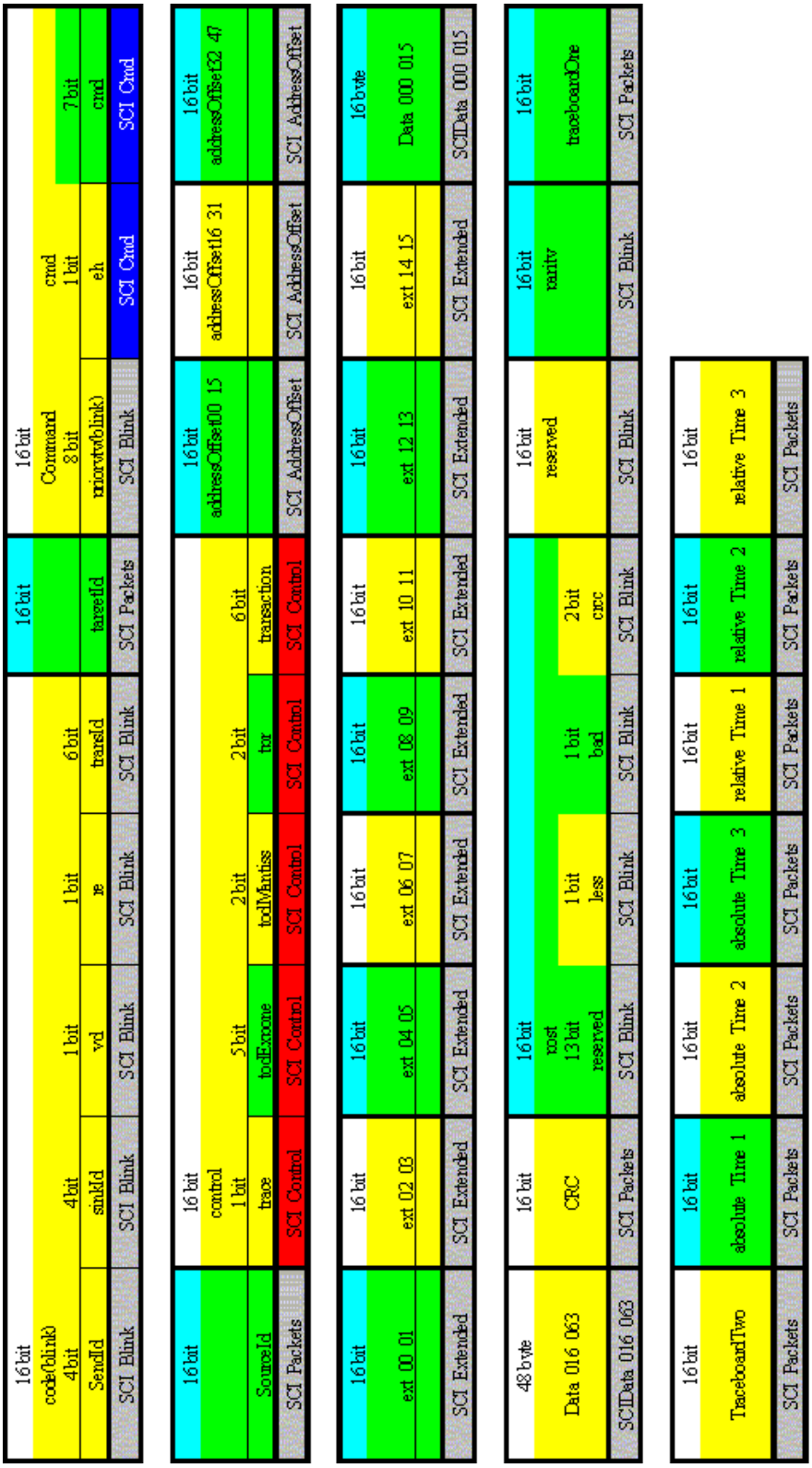
16 bit	16 bit	16 bit	16 bit	16 bit	16 bit
absolute Time 1	absolute Time 2	absolute Time 3	relative Time 1	relative Time 2	relative Time 3
SCI Packets	SCI Packets	SCI Packets	SCI Packets	SCI Packets	SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated request-send-packet with extended header and 64 bytes data

Packet Type 23

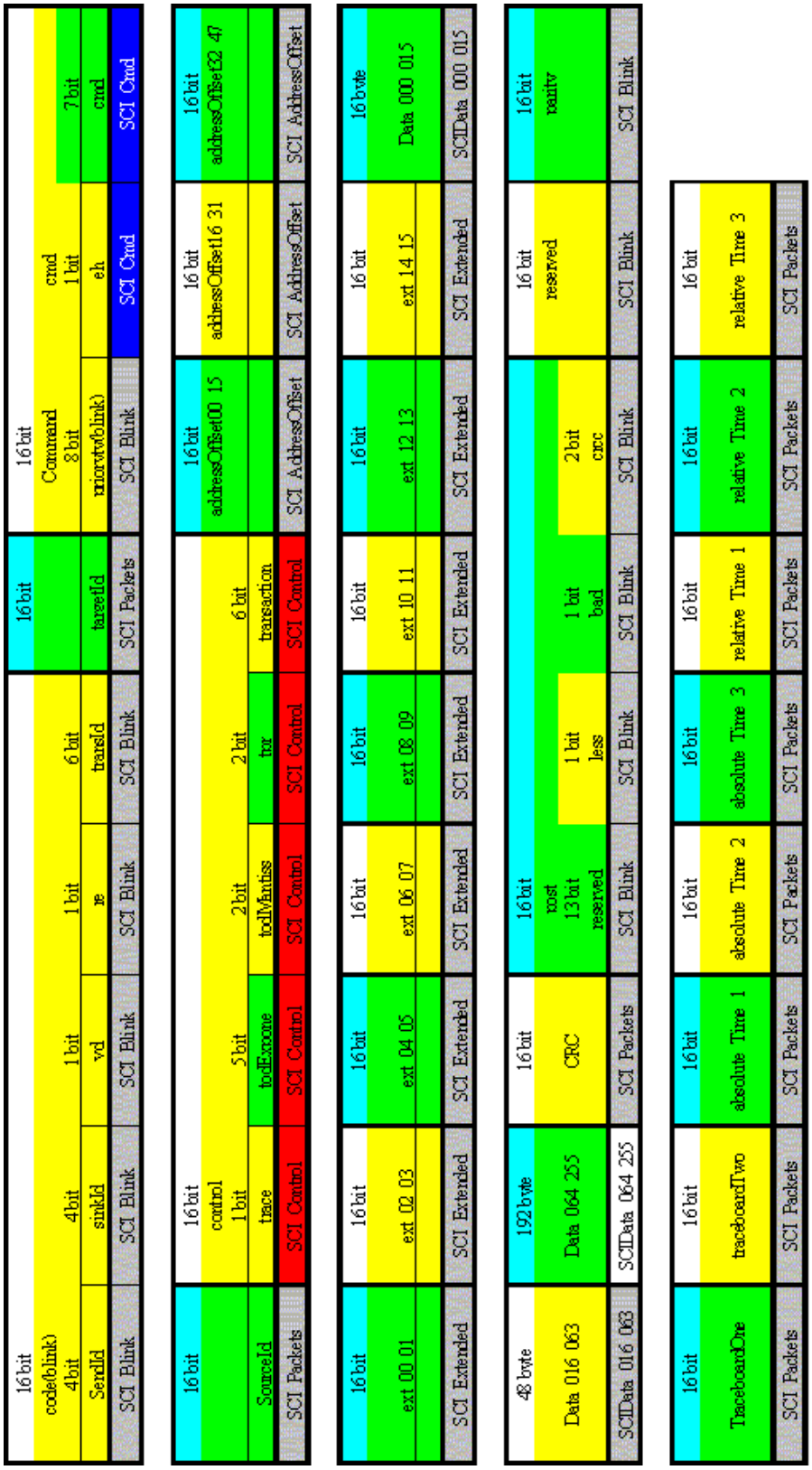


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated request-send-packet with extended header and 256 bytes data

Packet Type 24

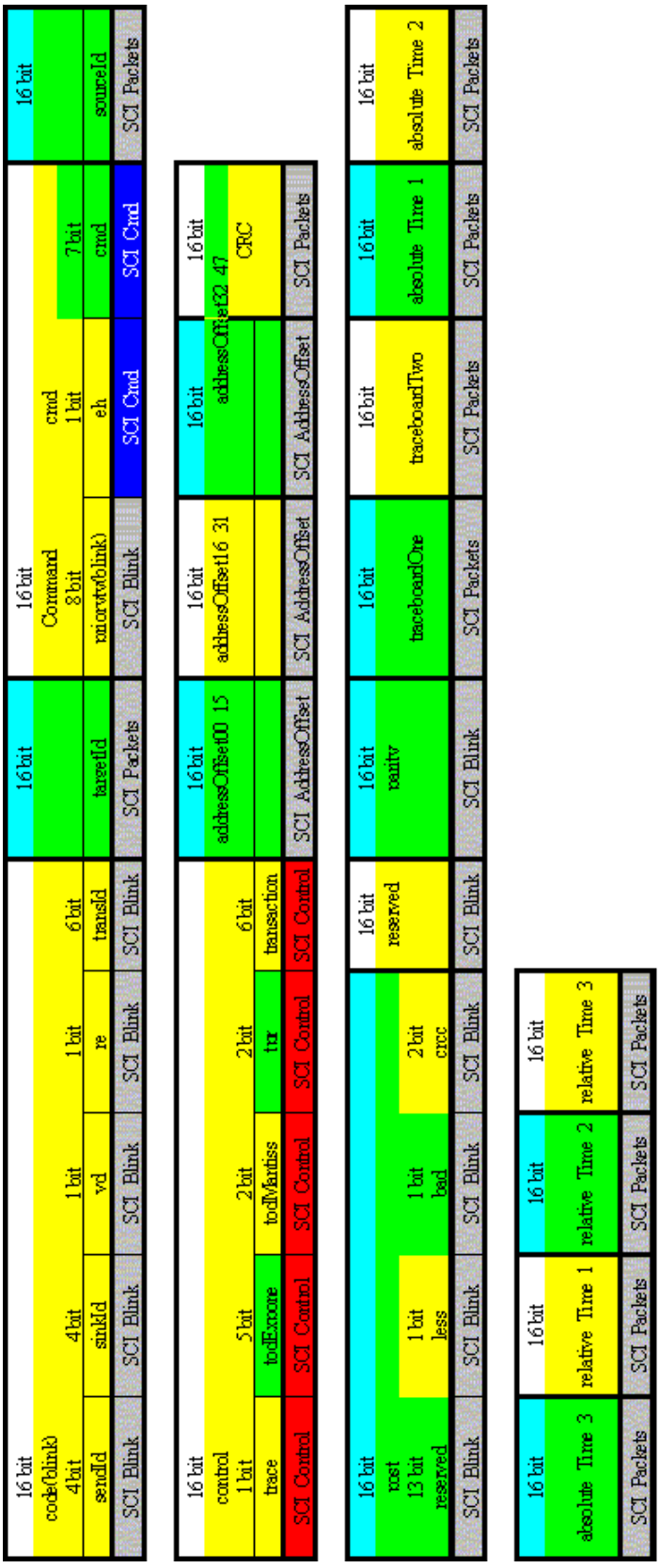

16 bit
code(oblink)
4 bit
SendId
4 bit
sinkId
1 bit
vd
1 bit
se
6 bit
traceId
16 bit
targetId
16 bit
Command
8 bit
micrcvtx(oblink)
cmd
1 bit
eh
7 bit
cmd
SCI Blink
SCI Blink
SCI Blink
SCI Blink
SCI Packets
SCI Blink
SCI Cmd
SCI Cmd
16 bit
SourceId
4 bit
traceId
16 bit
control
1 bit
trace
5 bit
todExzone
2 bit
todMvantiss
2 bit
txr
6 bit
transaction
16 bit
addressOffset00 15
16 bit
addressOffset16 31
16 bit
addressOffset32 47
SCI Packets
SCI Control
SCI Control
SCI Control
SCI Control
SCI Control
SCI AddressOffset
SCI AddressOffset
SCI AddressOffset
16 bit
ext 00 01
16 bit
ext 02 03
16 bit
ext 04 05
16 bit
ext 06 07
16 bit
ext 08 09
16 bit
ext 10 11
16 bit
ext 12 13
16 bit
ext 14 15
16 byte
Data 000 015
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCIData 000 015
48 byte
Data 016 063
192 byte
Data 064 255
16 bit
CRC
16 bit
rcst
13 bit
reserved
1 bit
less
1 bit
bad
2 bit
crcc
16 bit
reserved
16 bit
rcntv
SCIData 016 063
SCIData 064 255
SCI Packets
SCI Blink
SCI Blink
SCI Blink
SCI Blink
SCI Blink
16 bit
TraceboardOne
16 bit
traceboardTwo
16 bit
absolute Time 1
16 bit
absolute Time 2
16 bit
absolute Time 3
16 bit
relative Time 1
16 bit
relative Time 2
16 bit
relative Time 3
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated request-send-packet with 0 bytes data

Packet Type 25

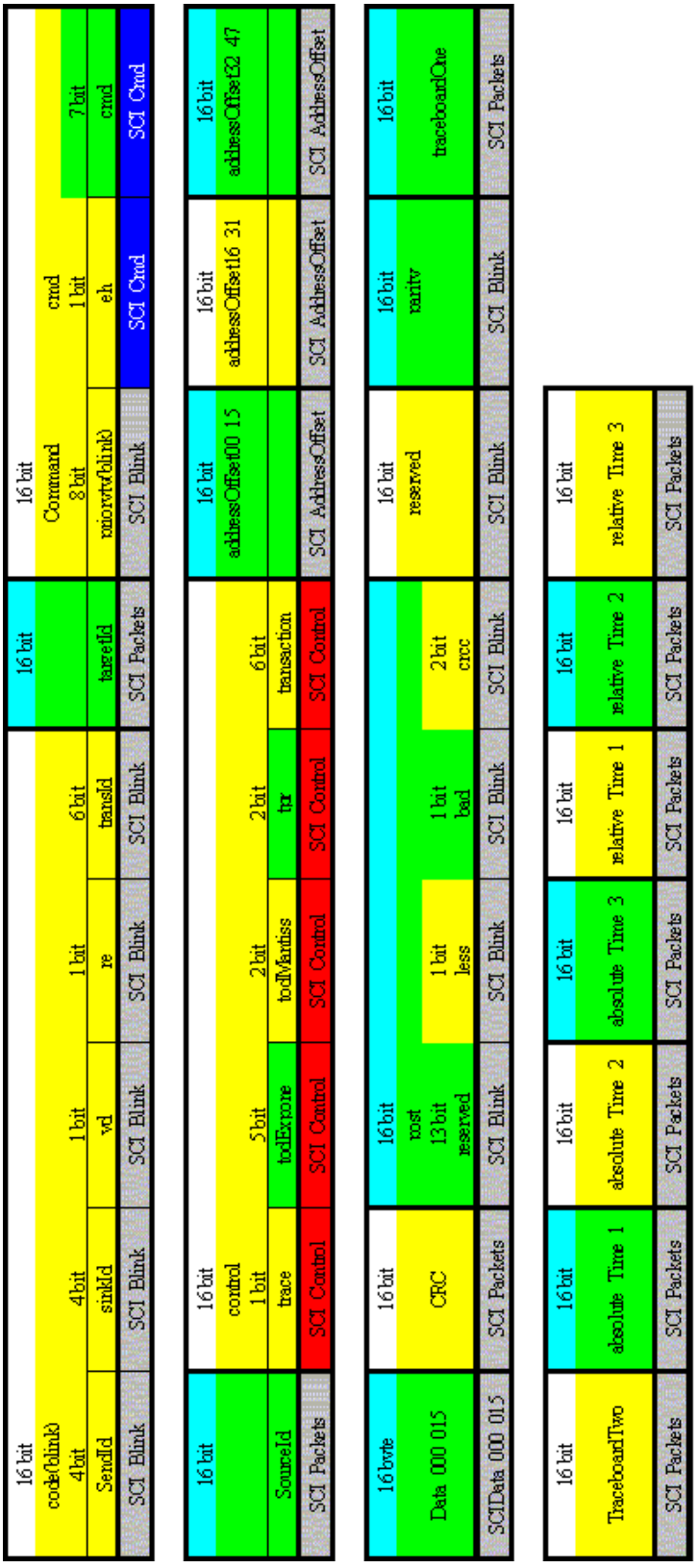


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated request-send-packet with 16 bytes data

Packet Type 26

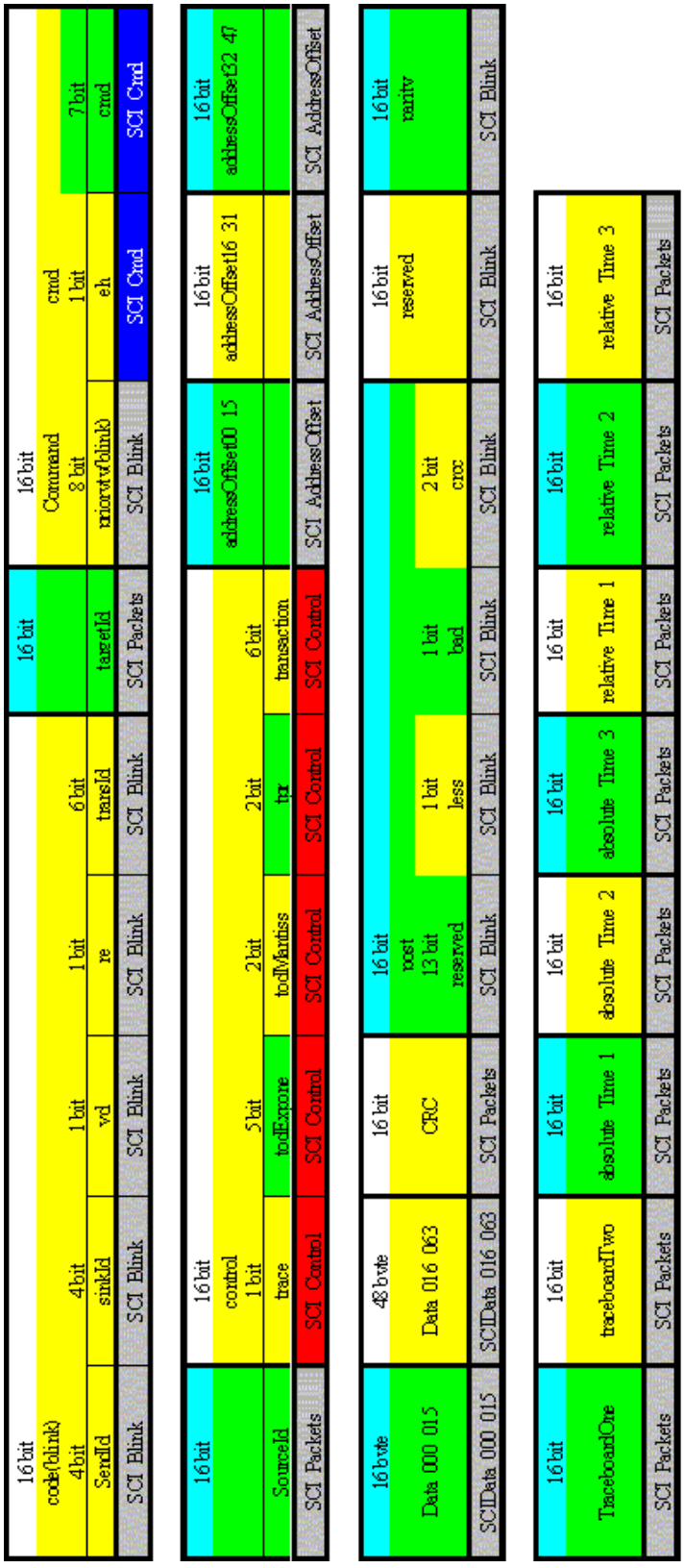


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated request-send-packet with 64 bytes data

Packet Type 27

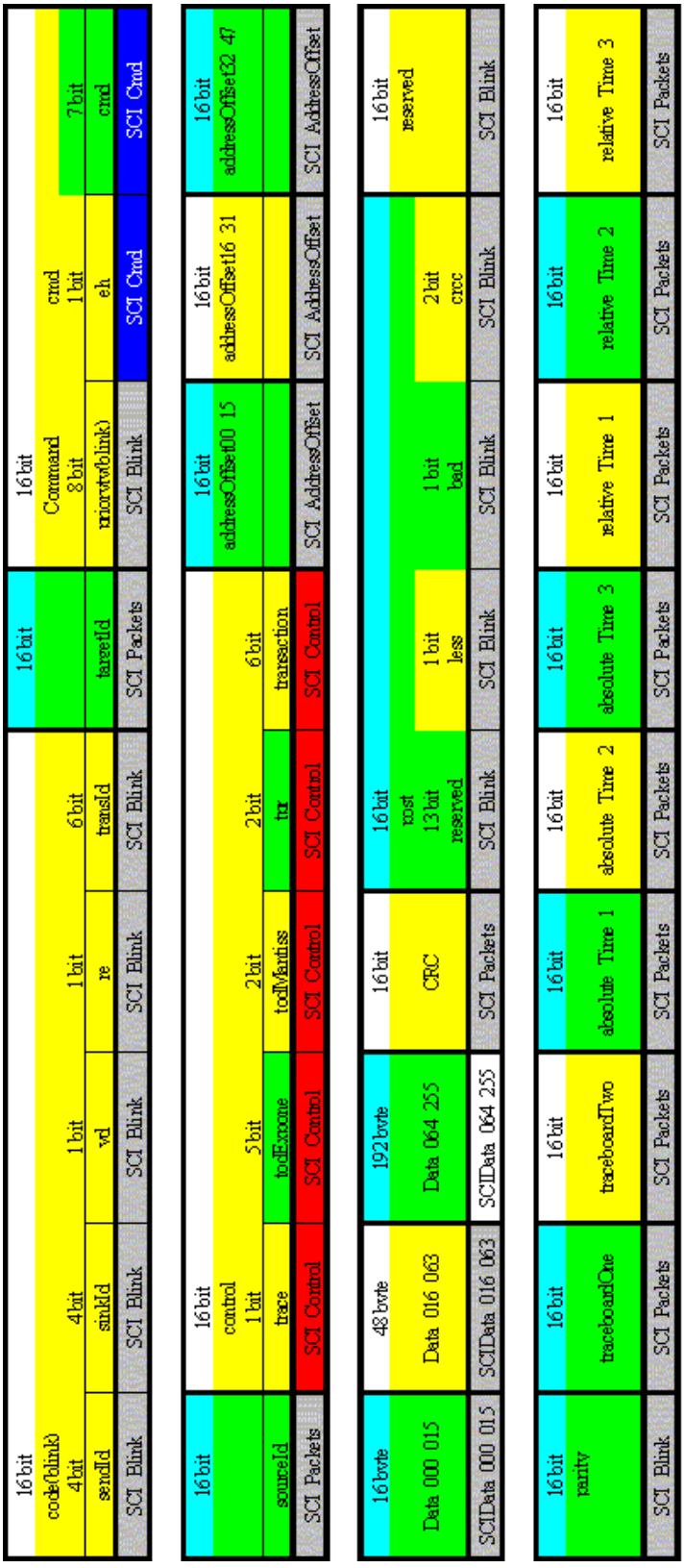


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated request-send-packet with 256 bytes data

Packet Type 28

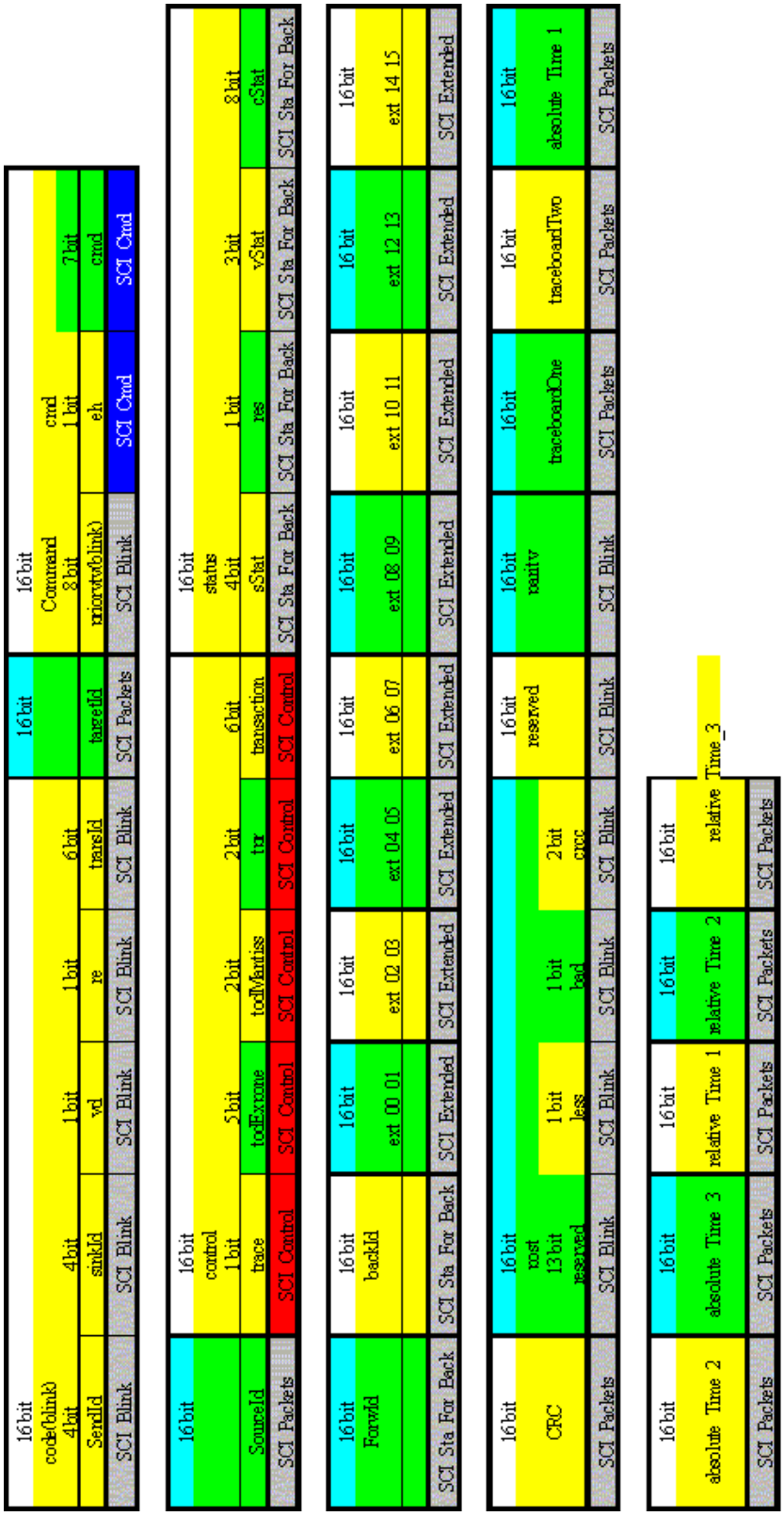


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated response-send-packet with extended header and 0 bytes data

Packet Type 29

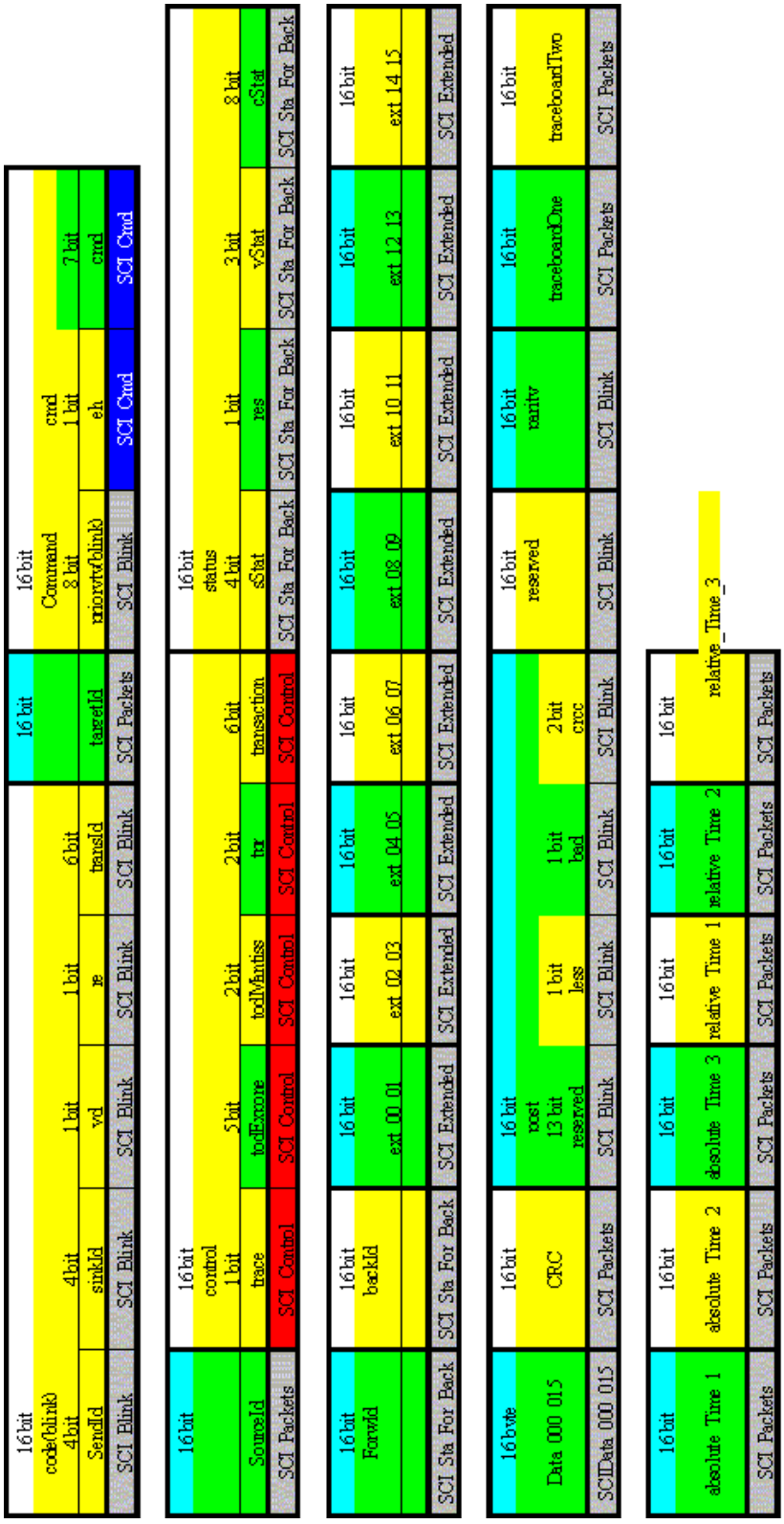

16 bit
code(0xlink)
4 bit
4 bit
1 bit
1 bit
6 bit
16 bit
16 bit
Command
8 bit
cmd
1 bit
7 bit
SendId
sinkId
vd
re
transId
targetId
priority(0xlink)
eh
cmd
SCI Blink
SCI Blink
SCI Blink
SCI Blink
SCI Blink
SCI Packets
SCI Blink
SCI Cmd
SCI Cmd
16 bit
control
1 bit
5 bit
2 bit
2 bit
6 bit
16 bit
status
4 bit
1 bit
3 bit
8 bit
SourceId
trace
toolExname
toolVantiss
tnr
transaction
sStat
res
vStat
eStat
SCI Packets
SCI Control
SCI Control
SCI Control
SCI Control
SCI Control
SCI Sta For Back
SCI Sta For Back
SCI Sta For Back
SCI Sta For Back
16 bit
ForwId
16 bit
backId
16 bit
ext 00 01
16 bit
ext 02 03
16 bit
ext 04 05
16 bit
ext 06 07
16 bit
ext 08 09
16 bit
ext 10 11
16 bit
ext 12 13
16 bit
ext 14 15
SCI Sta For Back
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCI Extended
SCI Extended
16 bit
CRC
16 bit
root
13 bit
reserved
1 bit
less
1 bit
bad
2 bit
crcc
16 bit
reserved
16 bit
reserv
16 bit
traceboardOne
16 bit
traceboardTwo
16 bit
absolute Time 1
SCI Packets
SCI Blink
SCI Blink
SCI Blink
SCI Blink
SCI Blink
SCI Blink
SCI Packets
SCI Packets
16 bit
absolute Time 2
16 bit
absolute Time 3
16 bit
relative Time 1
16 bit
relative Time 2
16 bit
relative Time_3
SCI Packets
SCI Packets
SCI Packets
SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated response-send-packet with extended header and 16 bytes data

Packet Type 30

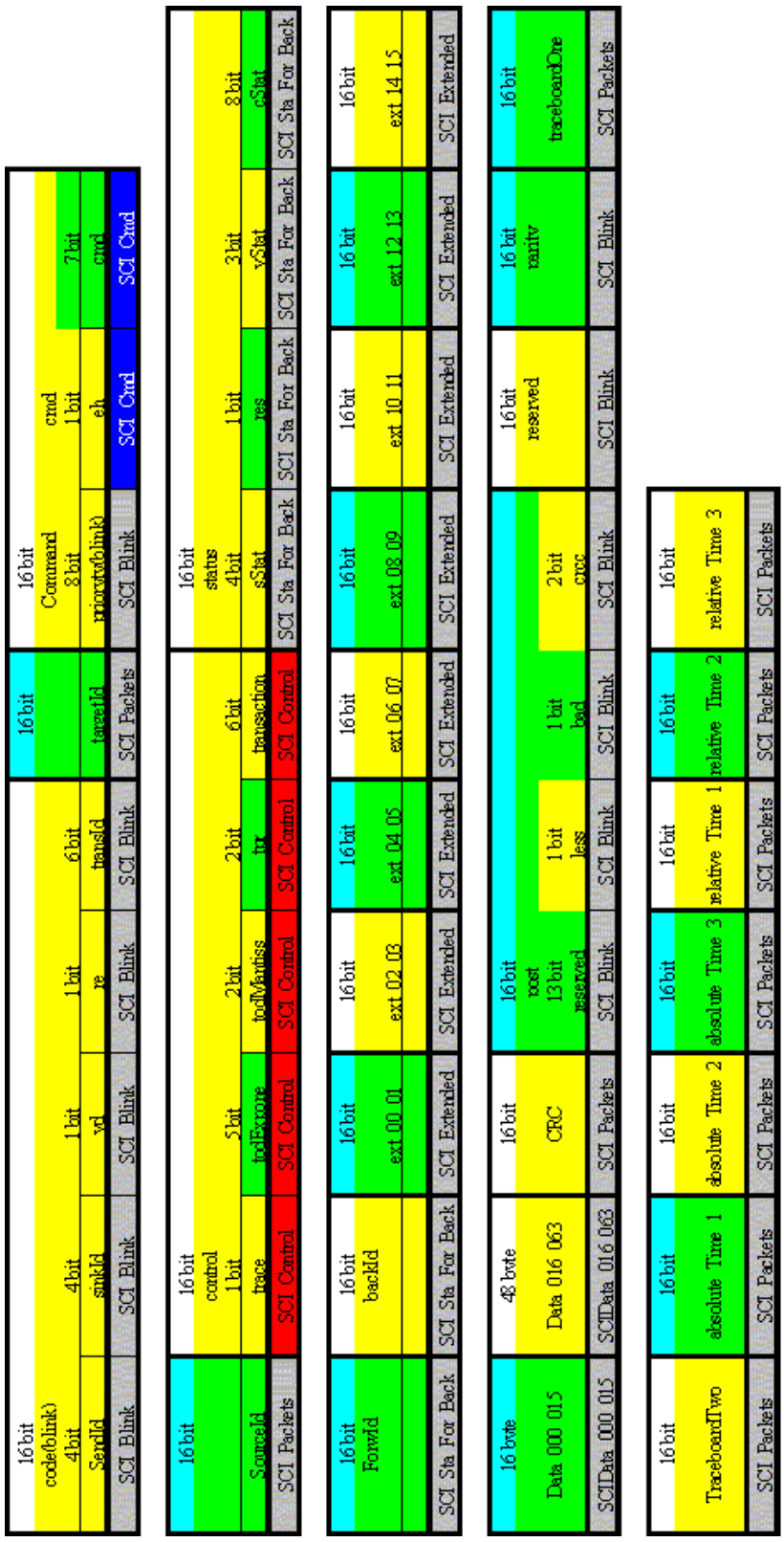


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated response-send-packet with extended header and 64 bytes data

Packet Type 31

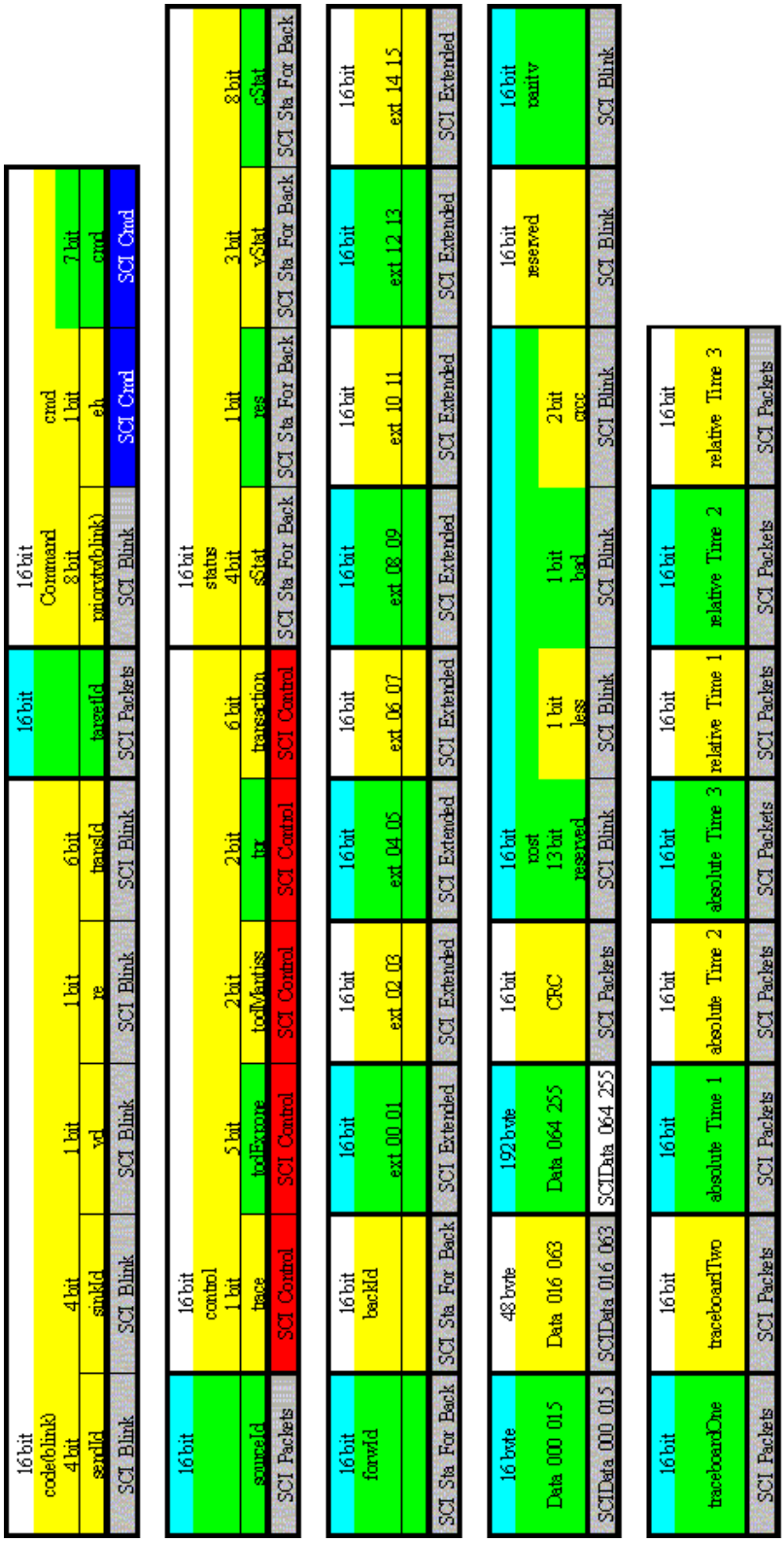


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated response-send-packet with extended header and 256 bytes data

Packet Type 32



The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated response-send-packet with 0 bytes data

Packet Type 33

16 bit code(blink)					16 bit	16 bit Command	cmd	
4 bit	4 bit	1 bit	1 bit	6 bit		8 bit	1 bit	7 bit
SendId	sinkId	vcl	re	transId	targetId	priority(blink)	eh	cmd
SCI Blink	SCI Blink	SCI Blink	SCI Blink	SCI Blink	SCI Packets	SCI Blink	SCI Cmd	SCI Cmd

16 bit	16 bit control					16 bit status			
	1 bit	5 bit	2 bit	2 bit	6 bit	4 bit	1 bit	3 bit	8 bit
SourceId	trace	todExpone	todMantiss	tpr	transaction	sStat	res	vStat	cStat
SCI Packets	SCI Control	SCI Control	SCI Control	SCI Control	SCI Control	SCI Sta For Back	SCI Sta For Back	SCI Sta For Back	SCI Sta For Back

16 bit	16 bit	16 bit	16 bit				16 bit	16 bit	16 bit
ForwId	backId	CRC	post				reserved	parity	traceboardOne
			13 bit	1 bit	1 bit	2 bit			
			reserved	less	bad	crcc			
SCI Sta For Back	SCI Sta For Back	SCI Packets	SCI Blink	SCI Blink	SCI Blink	SCI Blink	SCI Blink	SCI Blink	SCI Packets

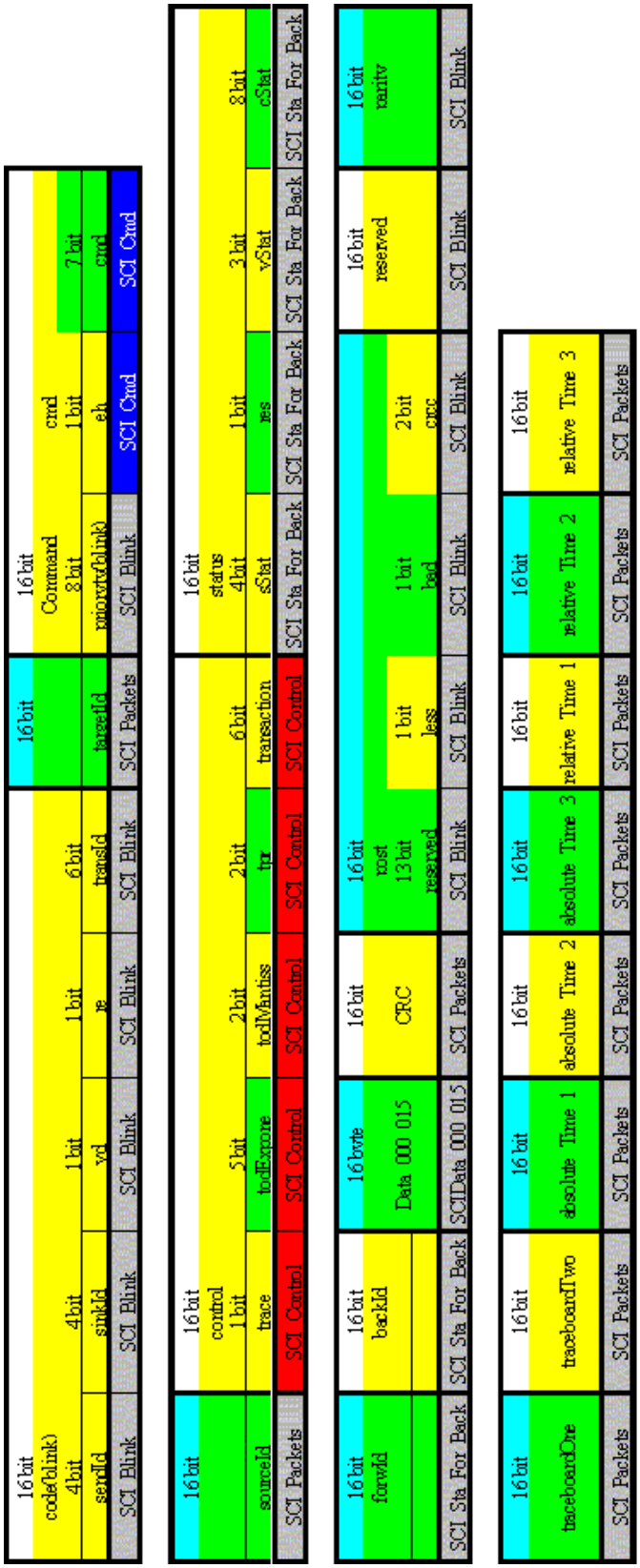
16 bit	16 bit	16 bit	16 bit	16 bit	16 bit	16 bit
TraceboardTwo	absolute Time 1	absolute Time 2	absolute Time 3	relative Time 1	relative Time 2	relative Time 3
SCI Packets	SCI Packets	SCI Packets	SCI Packets	SCI Packets	SCI Packets	SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated response-send-packet with 16 bytes data

Packet Type 34

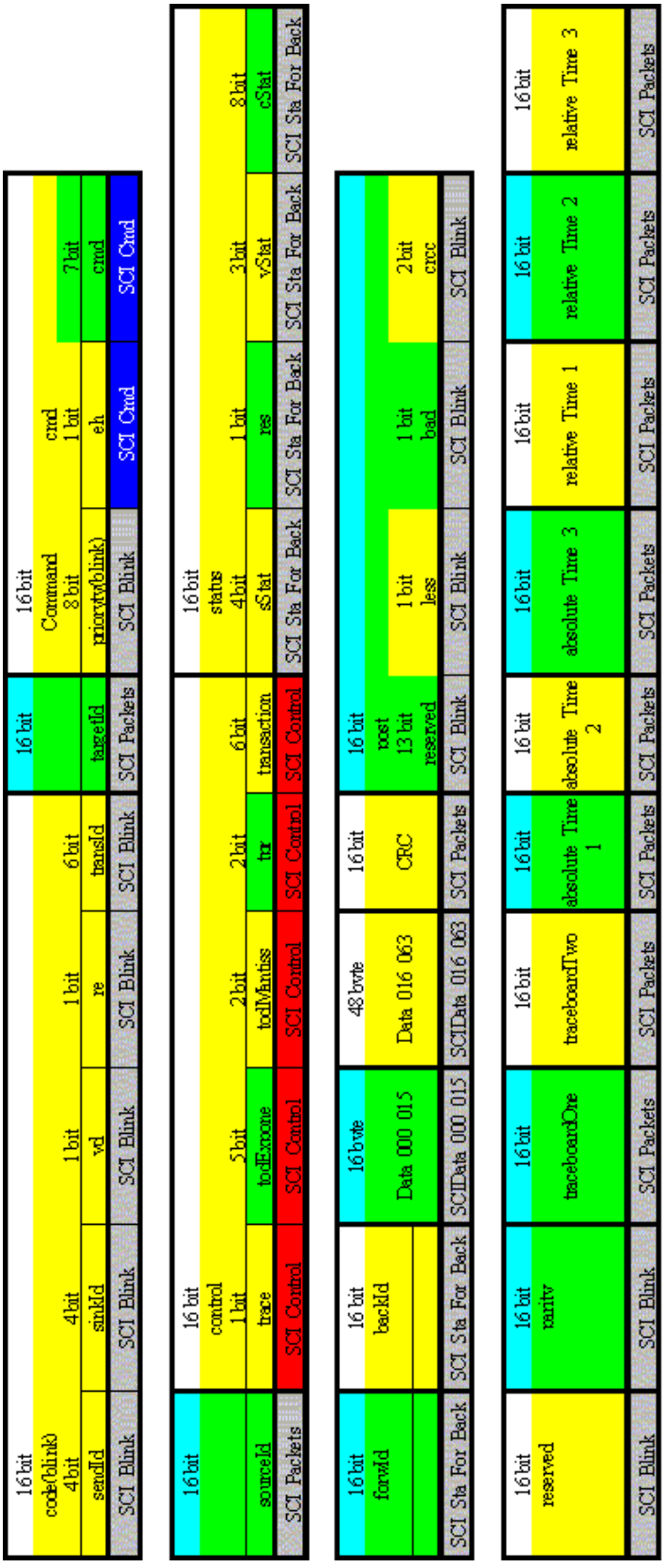


The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated response-send-packet with 64 bytes data

Packet Type 35

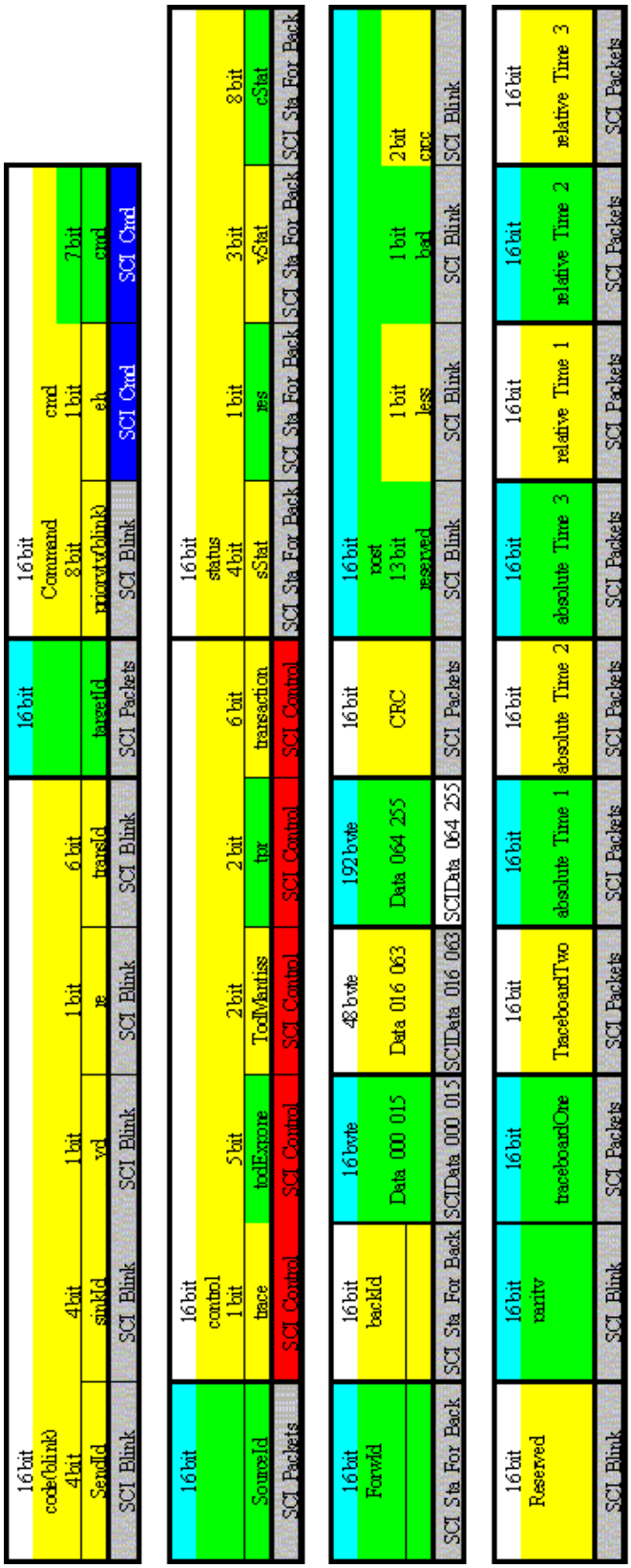

16 bit
code(blink)
4 bit
4 bit
sinkId
1 bit
vd
1 bit
re
6 bit
transId
16 bit
targetId
16 bit
Command
8 bit
priority(blink)
cmd
1 bit
eh
7 bit
cmd
sendId
SCI Blink
SCI Blink
SCI Blink
SCI Blink
SCI Packets
SCI Blink
SCI Cmd
SCI Cmd
16 bit
control
1 bit
trace
5 bit
todExtreme
2 bit
todMantiss
2 bit
hr
6 bit
transaction
16 bit
status
4 bit
sStat
1 bit
res
3 bit
vStat
8 bit
cStat
sourceId
SCI Packets
SCI Control
SCI Control
SCI Control
SCI Control
SCI Control
SCI Sta For Back
SCI Sta For Back
SCI Sta For Back
16 bit
forwId
16 bit
backId
16 byte
Data 000 015
48 byte
Data 016 063
16 bit
CRC
16 bit
root
13 bit
reserved
1 bit
less
1 bit
bad
2 bit
crcc
SCI Sta For Back
SCI Control
SCI Control
SCIData 000 015
SCIData 016 063
SCI Packets
SCI Blink
SCI Blink
SCI Blink
16 bit
reserved
16 bit
parity
16 bit
traceboardOne
16 bit
traceboardTwo
16 bit
absolute Time 1
16 bit
absolute Time 2
16 bit
absolute Time 3
16 bit
relative Time 1
16 bit
relative Time 2
16 bit
relative Time 3
SCI Blink
SCI Blink
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets
SCI Packets

The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3));
```

Appendix D: Encapsulated response-send-packet with 256 bytes data

Packet Type 36



The following SQL Query retrieves all information related a SCI packet type 1 Request-send-packet with extended header and 0 bytes data. In this instance form SCI packets with a TraceId =1 and a PacketId = 3.

```
SELECT SCI_Packets.TraceId,
             SCI_Packets.PacketId,
             SCI_Packets.Packet_Type_Id,
             SCI_Packet_Type_Id.Packet_Type_Description,
             SCI_FlowControl.mpr,
             SCI_FlowControl.spr,
             SCI_FlowControl.phase,
             SCI_FlowControl.old,
             SCI_FlowControl.ech,
             SCI_Cmd.cmd_eh,
             SCI_Cmd.cmd_cmd,
             SCI_Packets.sourceId,
             SCI_Control.control_trace,
             SCI_Control.control_todExpone,
             SCI_Control.control_todMantiss,
             SCI_Control.control_tpr,
             SCI_Control.control_transaction,
             SCI_AddressOffset.addressOffset00_15,
             SCI_AddressOffset.addressOffset16_31,
             SCI_AddressOffset.addressOffset32_47,
             SCI_Extended.ext_00_01,
             SCI_Extended.ext_02_03,
             SCI_Extended.ext_04_05,
             SCI_Extended.ext_06_07,
             SCI_Extended.ext_08_09,
             SCI_Extended.ext_10_11,
             SCI_Extended.ext_12_13,
             SCI_Extended.ext_14_15,
             SCI_Packets.CRC,
             SCI_Packets.traceboardOne,
             SCI_Packets.traceboardTwo,
             SCI_Packets.absolute_Time_1,
             SCI_Packets.absolute_Time_2,
             SCI_Packets.absolute_Time_3,
             SCI_Packets.relative_Time_1,
             SCI_Packets.relative_Time_2,
             SCI_Packets.relative_Time_3
FROM (((((SCI_Packet_Type_Id INNER JOIN SCI_Packets
         ON SCI_Packet_Type_Id.Packet_Type_Id = SCI_Packets.Packet_Type_Id)
                LEFT JOIN SCI_AddressOffset ON (SCI_Packets.PacketId =
SCI_AddressOffset.PacketId)
         AND (SCI_Packets.TraceId = SCI_AddressOffset.TraceId))
                LEFT JOIN SCI_Cmd ON (SCI_Packets.PacketId = SCI_Cmd.PacketId)
         AND (SCI_Packets.TraceId = SCI_Cmd.TraceId))
                LEFT JOIN SCI_Control ON (SCI_Packets.PacketId = SCI_Control.PacketId)
         AND (SCI_Packets.TraceId = SCI_Control.TraceId))
                LEFT JOIN SCI_Extended ON (SCI_Packets.PacketId = SCI_Extended.PacketId)
         AND (SCI_Packets.TraceId = SCI_Extended.TraceId))
                LEFT JOIN SCI_FlowControl ON (SCI_Packets.PacketId =
SCI_FlowControl.PacketId)
         AND (SCI_Packets.TraceId = SCI_FlowControl.TraceId)
WHERE (((SCI_Packets.TraceId) Like 1)
       AND ((SCI_Packets.PacketId) Like 3))
```

Appendix E: Java Trace Database Server

```
import java.sql.*;
import java.util.Properties;
import java.io.*;
import java.net.*;


//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//+   TraceDatabaseServer.java
+
//+   Michael Manzke
+
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

public class TraceDatabaseServer extends Thread {

  public static final int DEFAULT_PORT = 7777;
  protected int port;
  protected ServerSocket server;
  private static final String driverPrefixURL = "jdbc:odbc:";
  private static String username = null;
  private static String password = null;
  private static String dataSource = null;

  public static String getUsername() {
     return username;
  }

  public static String getPassword() {
     return password;
  }

  public static String getDriverURL() {
     return driverPrefixURL+dataSource;
  }


//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//+                                  M A I N
+
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

  public static void main (String args[]) {

    int port=0;


    // Port number from commandline
    if (args.length == 1) {

      try {

        port = Integer.parseInt (args[0]);

      }
      catch (NumberFormatException e) {
      }

    }


    // Load JDBC/ODBC driver
    try {

      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

    }
    catch (Exception e) {

      System.err.println("Failed to load JDBC/ODBC driver.");
      System.exit (1);

    }


    // Look for resource file 'odbc.datasource'
    try {
```

```
      InputStream is = ClassLoader.getSystemResourceAsStream ("odbc.datasource");
      Properties p = new Properties();
      p.load (is);
      dataSource = p.getProperty("datasource.name");
      if (dataSource == null)
        throw new Exception ();
      username = p.getProperty("datasource.username", "");
      password = p.getProperty("datasource.password", "");

    } catch (Exception e) {

      System.err.println("Unable to read resource to get data source");
      System.exit (1);

    }

    //Creates a new server thread
    new TraceDatabaseServer (port);

  }


//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//+                   TraceDatabaseServer Constructor
+
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

  public TraceDatabaseServer (int port) {

    super ("TraceDatabase Server");

      if (port == 0)
      port = DEFAULT_PORT;

    this.port = port;

    try {

       server = new ServerSocket (port);

    } catch (IOException e) {

       System.err.println ("Error creating server");
       System.exit (1);

    }

    //Invoke the thread's start methode
    start();

  }


  // The VM invokes th new tread's run method that makes the thread active
  public void run() {

    System.out.println ("Server Running");
    ThreadGroup connections = new ThreadGroup ("TraceDatabase Connections");
    connections.setMaxPriority(this.getPriority()-1);

    try {

      while (true) {
        Socket client = server.accept();
        System.out.println ("TraceDatabase Connection from: " +
client.getInetAddress().getHostName());
       TraceDatabaseConnection c = new TraceDatabaseConnection (connections, client);
      }

    } catch (IOException e) {

      System.err.println ("Exception listening");
      System.exit (1);

    }

    System.exit (0);

  }

}
```

```
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//+    T r a c e D a t a b a s e C o n n e c t i o n   C l a s s
+
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

class TraceDatabaseConnection extends Thread {

  static int counter = 0;
  protected ObjectInputStream in;
  protected ObjectOutputStream out;


//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//+   TraceDatabaseConnection Constructor
+
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

  public TraceDatabaseConnection (ThreadGroup group, Socket client) {

    super (group, "Connection " + counter++);

    try {

    in = new ObjectInputStream (client.getInputStream ());
    out = new ObjectOutputStream (client.getOutputStream());

    } catch (IOException e) {

      try {

        client.close();

      } catch (IOException e2) {

      }

      System.err.println ("Unable to connect to TraceDatabase");
      return;

    }

    start();

  }


  //The VM invokes the new thread's run method that makes the thread active
  public void run () {

    try {

      String mode = (String)in.readObject();


      //+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//+       I n s e r t   m o d e
+
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

      if (mode.equals ("insert")) {

        String name = (String)in.readObject();
        String username = (String)in.readObject();
        String comments = (String)in.readObject();

        try {

          Connection con = DriverManager.getConnection(
                           TraceDatabaseServer.getDriverURL(),
                           TraceDatabaseServer.getUsername(),
                           TraceDatabaseServer.getPassword());

          PreparedStatement prep = con.prepareStatement("INSERT into comments values (?, ?,
?)");
          prep.setString (1, name);
          prep.setString (2, username);
          prep.setString (3, comments);

          if (prep.executeUpdate () != 1)
            throw new Exception ("Bad Update");
```

```
        } catch (Exception e) {

          System.out.println ("Error updating." + e);
          return;

        }


      //+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//+       Q u e r y   m o d e
+ //+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

      } else if (mode.equals ("query")) {

        try {

          Connection con = DriverManager.getConnection(
                          TraceDatabaseServer.getDriverURL(),
                          TraceDatabaseServer.getUsername(),
                          TraceDatabaseServer.getPassword());

          Statement statement = con.createStatement();
          ResultSet result = statement.executeQuery("SELECT * FROM comments");
          System.out.println("Name\tEmail\tComments");
          int nameCol = result.findColumn ("name");
          int userCol = result.findColumn ("username");
          int commentsCol = result.findColumn ("comments");
          String name, user, comments;

          while(result.next()) {
            name = result.getString(nameCol);
            user = result.getString(userCol);
            comments = result.getString(commentsCol);
            System.out.println (name + "\t" + user + "\t" + comments);
          }

          statement.close();
          con.close();

        } catch (Exception e) {

          System.out.println ("Error querying." + e);
          return;
        }

      } else if (mode.equals ("scidata")) {
        try {

          System.out.println ("Start Connection to TraceDatabase");

          Connection con = DriverManager.getConnection(TraceDatabaseServer.getDriverURL(),
                                TraceDatabaseServer.getUsername(),
                                TraceDatabaseServer.getPassword());

      Statement statement = con.createStatement();

          System.out.println ("Connection to TraceDatabase is Established");
      System.out.println ("Start SCI data Query");

      ResultSet result = statement.executeQuery(
                                              "SELECT * FROM SCIData_064_255 WHERE ID = 1");

      System.out.println ("Finished TraceDatabase Query");

      int[] SCIData_064_255 = new int[97];

      System.out.println ( "The SCIData_064_255.length is:"
                            + Integer.toString(SCIData_064_255.length));

      if (result.next()) {

         for (int i = 1; i < SCIData_064_255.length + 1; i++) {
            SCIData_064_255[i-1] = result.getInt(i);
            System.out.println (  "Database field "
                                  + Integer.toString(i)
                                  + " : "
                                  + Integer.toString(SCIData_064_255[i-1]));
         }

      } else {
```

```
        System.out.println ("There are no more rows in the result set.");

      }

      out.writeObject (SCIData_064_255);
      out.flush();

      statement.close();
      con.close();

        } catch (Exception e) {

          System.out.println ("Error querying." + e);
          return;

        }


      } else {

        System.out.println ("Invalid Command: " + mode);

      }

      //out.close();

    } catch (Exception e) {

      System.out.println ("Error reading Stream." + e);

    }
    //out.close();
  }
}
```

Appendix F: Java Trace Data Decoding and Database Import

```
import java.io.*;
import java.util.Vector;


class SCIPacket{

   private static long nextSCIPacketId;
   private static long nextSCITraceId;
   //private Vector symbolbuf;
   //private byte[] bytearray;
   //FileInputStream streamDT1;
   //FileInputStream streamDT2;
   //private BufferedInputStream bufStreamDT1;
   //private BufferedInputStream bufStreamDT2;
   //private SCIPacket SCIPacketObject;


   //data from SCI_Packet:DatabaseTable

   private long SCIPacketId;
   private long SCITraceId;
   private long Packet_Type_Id;
   private long targetId;
   private long sourceId;
   private long CRC;
   private long traceboardOne;
   private long traceboardTwo;
   private long absolute_Time_1;
   private long absolute_Time_2;
   private long absolute_Time_3;
   private long relative_Time_1;
   private long relative_Time_2;
   private long relative_Time_3;


   //data from SCI_AddressOffset:DatabaseTable

   private long addressOffset00_15;
   private long addressOffset16_31;
   private long addressOffset32_47;


   //data from SCI_Blink:DatabaseTable

   private long code_sendId;
   private long code_sinkId;
   private long code_vd;
   private long code_re;
   private long code_transId;
   private long priority;
   private long post_reserved;
   private long post_less;
   private long post_bad;
   private long post_crcc;
   private long reserved;
   private long parity;


   //data from SCI_CMD:DatabaseTable

   private long cmd_eh;
   private long cmd_cmd;


   //data from SCI_CMD_Echo:DatabaseTable

   private long cmd_bsy;
   private long cmd_res;
   private long cmd_transactionId;


   //data from SCI_Control:DatabaseTable

   private long control_trace;
   private long control_todExpone;
   private long control_todMantiss;
```

```
private long control_tpr;
private long control_transaction;


//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
// Constructor for the SCIPacket class
//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

SCIPacket(){

   SCIPacketId = nextSCITraceId;

}



//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
// It follows the declarations for all 36 SCI packet types.
// These packet types are declared as inner classes.
//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


public class SCIPacketType01{

   private long SCIPacketType01Data;

   SCIPacketType01(long SCIPacketType01Data){

      this.SCIPacketType01Data = SCIPacketType01Data;

   }
}

public class SCIPacketType02{

   private long SCIPacketType02Data;

   SCIPacketType02(long SCIPacketType02Data){

      this.SCIPacketType02Data = SCIPacketType02Data;

   }
}

public class SCIPacketType03{

   private long SCIPacketType03Data;

   SCIPacketType03(long SCIPacketType03Data){

      this.SCIPacketType03Data = SCIPacketType03Data;

   }
}

public class SCIPacketType04{

   private long SCIPacketType04Data;

   SCIPacketType04(long SCIPacketType04Data){

      this.SCIPacketType04Data = SCIPacketType04Data;

   }
}

public class SCIPacketType05{

   private long SCIPacketType05Data;

   SCIPacketType05(long SCIPacketType05Data){

      this.SCIPacketType05Data = SCIPacketType05Data;

   }
}

public class SCIPacketType06{

   private long SCIPacketType06Data;

   SCIPacketType06(long SCIPacketType06Data){
```

```
        this.SCIPacketType06Data = SCIPacketType06Data;
    }
}
public class SCIPacketType07{
    private long SCIPacketType07Data;
    SCIPacketType07(long SCIPacketType07Data){
        this.SCIPacketType07Data = SCIPacketType07Data;
    }
}
public class SCIPacketType08{
    private long SCIPacketType08Data;
    SCIPacketType08(long SCIPacketType08Data){
        this.SCIPacketType08Data = SCIPacketType08Data;
    }
}
public class SCIPacketType09{
    private long SCIPacketType09Data;
    SCIPacketType09(long SCIPacketType09Data){
        this.SCIPacketType09Data = SCIPacketType09Data;
    }
}
public class SCIPacketType10{
    private long SCIPacketType10Data;
    SCIPacketType10(long SCIPacketType10Data){
        this.SCIPacketType10Data = SCIPacketType10Data;
    }
}
public class SCIPacketType11{
    private long SCIPacketType11Data;
    SCIPacketType11(long SCIPacketType11Data){
        this.SCIPacketType11Data = SCIPacketType11Data;
    }
}
public class SCIPacketType12{
    private long SCIPacketType12Data;
    SCIPacketType12(long SCIPacketType12Data){
        this.SCIPacketType12Data = SCIPacketType12Data;
    }
}
public class SCIPacketType13{
    private long SCIPacketType13Data;
    SCIPacketType13(long SCIPacketType13Data){
        this.SCIPacketType13Data = SCIPacketType13Data;
    }
}
```

```
public class SCIPacketType14{
   private long SCIPacketType14Data;
   SCIPacketType14(long SCIPacketType14Data){
      this.SCIPacketType14Data = SCIPacketType14Data;
   }
}
public class SCIPacketType15{
   private long SCIPacketType15Data;
   SCIPacketType15(long SCIPacketType15Data){
      this.SCIPacketType15Data = SCIPacketType15Data;
   }
}
public class SCIPacketType16{
   private long SCIPacketType16Data;
   SCIPacketType16(long SCIPacketType16Data){
      this.SCIPacketType16Data = SCIPacketType16Data;
   }
}
public class SCIPacketType17{
   private long SCIPacketType17Data;
   SCIPacketType17(long SCIPacketType17Data){
      this.SCIPacketType17Data = SCIPacketType17Data;
   }
}
public class SCIPacketType18{
   private long SCIPacketType18Data;
   SCIPacketType18(long SCIPacketType18Data){
      this.SCIPacketType18Data = SCIPacketType18Data;
   }
}
public class SCIPacketType19{
   private long SCIPacketType19Data;
   SCIPacketType19(long SCIPacketType19Data){
      this.SCIPacketType19Data = SCIPacketType19Data;
   }
}
public class SCIPacketType20{
   private long SCIPacketType20Data;
   SCIPacketType20(long SCIPacketType20Data){
      this.SCIPacketType20Data = SCIPacketType20Data;
   }
}
public class SCIPacketType21{
   private long SCIPacketType21Data;
```

```
    SCIPacketType21(long SCIPacketType21Data){
        this.SCIPacketType21Data = SCIPacketType21Data;
    }
}
public class SCIPacketType22{
    private long SCIPacketType22Data;
    SCIPacketType22(long SCIPacketType22Data){
        this.SCIPacketType22Data = SCIPacketType22Data;
    }
}
public class SCIPacketType23{
    private long SCIPacketType23Data;
    SCIPacketType23(long SCIPacketType23Data){
        this.SCIPacketType23Data = SCIPacketType23Data;
    }
}
public class SCIPacketType24{
    private long SCIPacketType24Data;
    SCIPacketType24(long SCIPacketType24Data){
        this.SCIPacketType24Data = SCIPacketType24Data;
    }
}
public class SCIPacketType25{
    private long SCIPacketType25Data;
    SCIPacketType25(long SCIPacketType25Data){
        this.SCIPacketType25Data = SCIPacketType25Data;
    }
}
public class SCIPacketType26{
    private long SCIPacketType26Data;
    SCIPacketType26(long SCIPacketType26Data){
        this.SCIPacketType26Data = SCIPacketType26Data;
    }
}
public class SCIPacketType27{
    private long SCIPacketType27Data;
    SCIPacketType27(long SCIPacketType27Data){
        this.SCIPacketType27Data = SCIPacketType27Data;
    }
}
public class SCIPacketType28{
    private long SCIPacketType28Data;
    SCIPacketType28(long SCIPacketType28Data){
        this.SCIPacketType28Data = SCIPacketType28Data;
    }
```

```
}
public class SCIPacketType29{
   private long SCIPacketType29Data;
   SCIPacketType29(long SCIPacketType29Data){
      this.SCIPacketType29Data = SCIPacketType29Data;
   }
}
public class SCIPacketType30{
   private long SCIPacketType30Data;
   SCIPacketType30(long SCIPacketType30Data){
      this.SCIPacketType30Data = SCIPacketType30Data;
   }
}
public class SCIPacketType31{
   private long SCIPacketType31Data;
   SCIPacketType31(long SCIPacketType31Data){
      this.SCIPacketType31Data = SCIPacketType31Data;
   }
}
public class SCIPacketType32{
   private long SCIPacketType32Data;
   SCIPacketType32(long SCIPacketType32Data){
      this.SCIPacketType32Data = SCIPacketType32Data;
   }
}
public class SCIPacketType33{
   private long SCIPacketType33Data;
   SCIPacketType33(long SCIPacketType33Data){
      this.SCIPacketType33Data = SCIPacketType33Data;
   }
}
public class SCIPacketType34{
   private long SCIPacketType34Data;
   SCIPacketType34(long SCIPacketType34Data){
      this.SCIPacketType34Data = SCIPacketType34Data;
   }
}
public class SCIPacketType35{
   private long SCIPacketType35Data;
   SCIPacketType35(long SCIPacketType35Data){
      this.SCIPacketType35Data = SCIPacketType35Data;
   }
}
public class SCIPacketType36{
   private long SCIPacketType36Data;
```

```
      SCIPacketType36(long SCIPacketType36Data){

         this.SCIPacketType36Data = SCIPacketType36Data;

      }
   }

   // The end of the inner class declaration for all the 36 SCI packet types.


   //+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   //                             M a i n
   //+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


   public static void main(String[] args){

      SCIPacket SCIPacketObject = new SCIPacket();

      try {


         // Open buffered input streams from the two trace files TraceBoard1.sci and
TraceBoard2.sci

         File TraceBoard1File = new File("TraceBoard1.sci");
         System.out.println("Created File Object for TraceBoard1.sci");
         File TraceBoard2File = new File("TraceBoard2.sci");
         System.out.println("Created File Object for TraceBoard2.sci");

         FileInputStream streamDT1 = new FileInputStream(TraceBoard1File);
         System.out.println("Open byte stream from TraceBoard1.sci");
         FileInputStream streamDT2 = new FileInputStream(TraceBoard2File);
         System.out.println("Open byte stream from TraceBoard2.sci");


         // create buffered input streams for streamDT1 and streamDT2

         BufferedInputStream bufStreamDT1 = new BufferedInputStream(streamDT1);
         BufferedInputStream bufStreamDT2 = new BufferedInputStream(streamDT2);

         System.out.println(bufStreamDT1.available() + " Bytes Opened may be read from
TraceBoard1.sci");
         System.out.println(bufStreamDT2.available() + " Bytes Opened may be read from
TraceBoard2.sci");


         // Create a 100 elements Vector of byte[12] arrays.
         // The Vector will be used as FIFO for the two BufferedInputStreams
         // from the TraceBoard1.sci file and the TraceBoard2.sci file.


         Vector symbolbuf = new Vector(100);

         System.out.println(symbolbuf.size() + " is symbolbuf Vector size before the bytes
arrays were set");
         System.out.println(symbolbuf.capacity() + " is symbolbuf Vector capacity before the
bytes arrays were set");


         // Fill the Vector to its capacity with dat from bufStreamDT1 and bufStreamDT2


         for (int symbolindex = symbolbuf.size(); symbolindex < symbolbuf.capacity();
symbolindex++){

             byte[] bytearray = new byte[12];

             // bufStreamDT1.read(bytearray,0,6); // this makes no difference
             // bufStreamDT2.read(bytearray,6,6);

             for (int bytearrayindex = 0; bytearrayindex < 6; bytearrayindex++ ){
                 bytearray[bytearrayindex] = (byte)bufStreamDT1.read();
             }

             for (int bytearrayindex = 6; bytearrayindex < 12; bytearrayindex++ ){
                 bytearray[bytearrayindex] = (byte)bufStreamDT2.read();
             }

             symbolbuf.addElement(bytearray);
         }
```

```
        //for (int symbolindex = symbolbuf.size(); symbolindex < symbolbuf.capacity();
symbolindex++){
        //
        //    SCIPacketObject.vectorFill();
        //
        //}


        // Print the vector to the standard system output

        for (int symbolindex = 0; symbolindex < symbolbuf.size(); symbolindex++){

            byte[] bytearray = new byte[12];

            bytearray = (byte[])symbolbuf.elementAt(symbolindex);

            System.out.println( "symbolindex = " + symbolindex + "   " +
                                bytearray[0]      + " " +
                                bytearray[1]      + " " +
                                bytearray[2]      + " " +
                                bytearray[3]      + " " +
                                bytearray[4]      + " " +
                                bytearray[5]      + " " +
                                bytearray[6]      + " " +
                                bytearray[7]      + " " +
                                bytearray[8]      + " " +
                                bytearray[9]      + " " +
                                bytearray[10]     + " " +
                                bytearray[11] );

        }


        System.out.println(symbolbuf.size() + " The Vector size after the it has been
filled");
        System.out.println(symbolbuf.capacity() + " the Vector capacity after it has been
filled");

        System.out.println("&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&");
        System.out.println("   Remove the first packet from the Vector  ");
        System.out.println("&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&");


        for (int removeindex = 0; removeindex < 50; removeindex++){

            symbolbuf.removeElementAt(0);

        }

        System.out.println(symbolbuf.size() + " The Vector size after 50 elements have been
removed");
        System.out.println(symbolbuf.capacity() + " the Vector capacity after 50 elements
have been removed");


        // Print the vector to the standard system output

        for (int symbolindex = 0; symbolindex < symbolbuf.size(); symbolindex++){

            byte[] bytearray = new byte[12];

            bytearray = (byte[])symbolbuf.elementAt(symbolindex);

            System.out.println( "symbolindex = " + symbolindex + "   " +
                                bytearray[0]      + " " +
                                bytearray[1]      + " " +
                                bytearray[2]      + " " +
                                bytearray[3]      + " " +
                                bytearray[4]      + " " +
                                bytearray[5]      + " " +
                                bytearray[6]      + " " +
                                bytearray[7]      + " " +
                                bytearray[8]      + " " +
                                bytearray[9]      + " " +
                                bytearray[10]     + " " +
                                bytearray[11] );
```

```
        }

        System.out.println("&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&");
        System.out.println("   Refill the Vector with new SCI Packets   ");
        System.out.println("&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&");


        // Fill the Vector to its capacity with dat from bufStreamDT1 and bufStreamDT2

        // TheObject.vectorFill();


        // Fill the Vector to its capacity with data from bufStreamDT1 and bufStreamDT2

        SCIPacketObject.vectorFill(symbolbuf.capacity() - symbolbuf.size(), bufStreamDT1,
bufStreamDT2, symbolbuf);


        // Print the vector to the standard system output

        for (int symbolindex = 0; symbolindex < symbolbuf.size(); symbolindex++){

            byte[] bytearray = new byte[12];

            bytearray = (byte[])symbolbuf.elementAt(symbolindex);

            System.out.println( "symbolindex = " + symbolindex + "   " +
                                bytearray[0]      + " " +
                                bytearray[1]      + " " +
                                bytearray[2]      + " " +
                                bytearray[3]      + " " +
                                bytearray[4]      + " " +
                                bytearray[5]      + " " +
                                bytearray[6]      + " " +
                                bytearray[7]      + " " +
                                bytearray[8]      + " " +
                                bytearray[9]      + " " +
                                bytearray[10]     + " " +
                                bytearray[11] );

        }


        System.out.println(symbolbuf.size() + " The Vector size after the it has been
refilled");
        System.out.println(symbolbuf.capacity() + " the Vector capacity after it has been
refilled");


        // close the buffered input streams form TraceBoard1.sci and TraceBoard2.sci

        bufStreamDT1.close();
        System.out.println("Closed byte stream from TraceBoard1.sci");
        bufStreamDT2.close();
        System.out.println("Closed byte stream from TraceBoard2.sci");


     } catch (IOException e) {

        e.printStackTrace();

     }

  }


  //+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  //                      M e t h o d s
  //+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


  public void vectorFill(int fillIterations, BufferedInputStream stream1,
BufferedInputStream stream2, Vector symbolVec) {

     try {
```

```
         for (int symbolindex = 0; symbolindex < fillIterations; symbolindex++){

            symbolVec.addElement(new byte[12]);

            stream1.read((byte[])symbolVec.lastElement(),0,6);
            stream2.read((byte[])symbolVec.lastElement(),6,6);

         }

         symbolVec = null;


      } catch (IOException e) {

         e.printStackTrace();

      }
   }

} // End of the SCIPacket class
```

Appendix G: Java Trace Database GUI and Analysis Tool

```
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//                                                                SCIDatabase.java
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

/**

 * This is a template.  You may modify this file.

 *

 * Runtime vendor: SunSoft, Inc.

 * Runtime version: 1.0

 *

 * Visual vendor: SunSoft, Inc.

 * Visual version: 1.0

 */



import sunsoft.jws.visual.rt.base.*;

import sunsoft.jws.visual.rt.shadow.java.awt.*;
import sunsoft.jws.visual.rt.shadow.java.awt.TextAreaShadow;

import sunsoft.jws.visual.rt.shadow.java.awt.ButtonShadow;

import sunsoft.jws.visual.rt.shadow.java.awt.TextFieldShadow;
import java.lang.Integer.*;

import java.awt.*;
import java.io.*;
import java.net.*;
import java.awt.event.*;
import java.util.*;



public class SCIDatabase extends Group {


  private SCIDatabaseRoot gui;
  public static final int DEFAULT_PORT = 7777;

  private static final String SCIdataString = "scidata";
  public int[] SCIData_064_255 = new int[97];

  private int port = 0;

  private String host = null;

  private OutputStream os = null;



  /**

   * Sample method call ordering during a group's lifetime:

   *

   * Constructor

   * initRoot

   * initGroup
```

```
 * (setOnGroup and getOnGroup may be called at any time in any
 *  order after initGroup has been called)
 * createGroup
 * showGroup/hideGroup + startGroup/stopGroup
 * destroyGroup
 */

/**
 * All the attributes used by the group must be defined in the
 * constructor.  setOnGroup is called at initialization for all
 * the attributes.  If the attribute has not been set prior to
 * initialization, setOnGroup is called with the default value.
 */
public SCIDatabase() {
  /**
   * Define the group's custom attributes here.
   *
   * For example:
   *
   * attributes.add("customString", "java.lang.String",
   *           "Default String", 0);
   */

  /**
   * This method defines the attributes that will be forwarded to
   * the main child (either a window or a panel).  All attributes
   * defined by this method are marked with the FORWARD flag.
   */
  addForwardedAttributes();
}

/**
 * initRoot must be overridden in group subclasses to initialize
 * the shadow tree.  The return value must be the root of the
 * newly initialized shadow tree.
 */
protected Root initRoot() {
  /**
   * Initialize the gui components
   */
  gui = new SCIDatabaseRoot(this);
```

```
    /**
     * This method registers an attribute manager with the group, such
     * that attributes marked with the FORWARD flag will be sent to
     * this attribute manager.
     */
    addAttributeForward(gui.getMainChild());

    return gui;
  }

  /**
   * initGroup is called during initialization.  It is called just after
   * initRoot is called, but before the sub-groups are initialized and
   * before the attributes are sent to the setOnGroup method.
   *
   * initGroup is only called once in the lifetime of the Group.
   * This is because groups cannot be uninitialized.  Anything that
   * needs to be cleaned up should be created in createGroup instead
   * of initGroup, and then can be cleaned up in destroyGroup.
   * createGroup and destroyGroup may be called multiple times during
   * the lifetime of a group.
   */
  protected void initGroup() { }

  /**
   * showGroup may be overridden by group subclasses that want
   * to know when the group becomes visible.  It is called just before
   * the group becomes visible.  The group will already be initialized
   * and created at this point.
   */
  protected void showGroup() { }

  /**
   * hideGroup may be overridden by group subclasses that want
   * to know when the group becomes non-visible.  It is called just
   * before the group becomes non-visible.
   */
  protected void hideGroup() { }

  /**
```

```
     * createGroup is called during group creation.  Groups can be
     * created and destroyed multiple times during their lifetime.
     * Anything that is created in createGroup should be cleaned up
     * in destroyGroup.  createGroup is called just after the group
     * has been created.  Anything that needs to be done before the
     * group is created should be done in initGroup.
     */
    protected void createGroup() { }

    /**
     * destroyGroup is called during the destroy operation.  Groups can
     * be created and destroyed multiple times during their lifetime.
     * Anything that has been created in createGroup should be cleaned up
     * in destroyGroup.  destroyGroup is called just before the group
     * is destroyed.
     */
    protected void destroyGroup() { }

    /**
     * This method may be overridden by group subclasses that want
     * to be informed when the application is starting.  This method is
     * only called after the entire application has been initialized and
     * created.
     *
     * For applets, startGroup is called whenever start is called on the
     * applet.
     */
    protected void startGroup() { }

    /**
     * This method may be overridden by group subclasses that want
     * to be informed when the application is stopping.  This method
     * will be called before a destroy is done.
     *
     * For applets, stopGroup is called whenever stop is called on the
     * applet.
     */
    protected void stopGroup() { }

    /**
     * "getOnGroup" may be overridden by sub-groups that
```

```
 * store attribute values themselves, and do not depend on the
 * group superclass to store them.  This method should be overridden
 * instead of "get".  Any attributes handled in setOnGroup where
 * super.setOnGroup is not called must also be handled in getOnGroup.
 *
 * The default implementation of getOnGroup retrieves the value
 * from the attribute table.
 *
 * The reason that "getOnGroup" should be overridden instead
 * of "get" is that "getOnGroup" is guaranteed not to be called
 * until the group class is initialized.  This means that initRoot
 * will always be called before any calls to getOnGroup are made.
 *
 * Also, this method is only for attributes that are defined in the
 * sub-groups.  It is not called for forwarded attributes.
 */
protected Object getOnGroup(String key) {
  return super.getOnGroup(key);
}

/**
 * "setOnGroup" may be overridden by sub-groups that
 * want notification when attributes are changed.  This method
 * should be overridden instead of "set".  Any attributes handled
 * in setOnGroup where super.setOnGroup is not called must also be
 * handled in getOnGroup.
 *
 * The default implementation of setOnGroup puts the value
 * in the attribute table.
 *
 * The reason that "setOnGroup" should be overridden instead
 * of "set" is that "setOnGroup" is guaranteed not to be called
 * until the group class is initialized.  This means that initRoot
 * will always be called before any calls to setOnGroup are made.
 *
 * During initialization, "setOnGroup" will be called for all
 * the group's attributes even if they have not be changed from
 * the default value.  But for attributes that have the DEFAULT
 * flag set, "setOnGroup" will only be called if the value
 * of the attribute has changed from the default.
 *
```

```
 * Also, this method is only called when attributes defined in the
 * sub-groups are updated.  It is not called for forwarded attributes.
 */
protected void setOnGroup(String key, Object value) {
  super.setOnGroup(key, value);
}

/**
 * handleMessage may be overridden by subclasses that want to act
 * on messages that are sent to the group.  Typically, messages are
 * either AWT events that have been translated to messages, or they
 * are messages that have been sent by other groups.
 * super.handleMessage should be called for any messages that aren't
 * handled.  If super.handleMessage is not called, then handleEvent
 * will not be called.
 *
 * The default implementation of handleMessage returns "true".  This
 * means that no events will be passed up the group tree, unless a
 * subclass overrides this method to return "false".  AWT events are
 * not propagated regardless of the return value from handleEvent.
 *
 * If you want a message to go to the parent group, override
 * handleMessage to return false for that message.
 *
 * If you want an AWT event to go to the parent group, you need to
 * call postMessageToParent() with the event message.
 */
public boolean handleMessage(Message msg) {
  return super.handleMessage(msg);
}

/**
 * handleEvent may be overridden by subclasses that want to get
 * notified when AWT events that are sent by the gui components.
 * The return value should be true for handled events, and
 * super.handleEvent should be called for unhandled events.
 * If super.handleEvent is not called, then the specific event
 * handling methods will not be called.
 *
 * The message's target is set to the shadow that sent the event.
 * The event's target is set to the AWT component that sent the event.
```

```
   *
   *
   * The following specific event handling methods may also be overridden:
   *
   * public boolean mouseDown(Message msg, Event evt, int x, int y);
   * public boolean mouseDrag(Message msg, Event evt, int x, int y);
   * public boolean mouseUp(Message msg, Event evt, int x, int y);
   * public boolean mouseMove(Message msg, Event evt, int x, int y);
   * public boolean mouseEnter(Message msg, Event evt, int x, int y);
   * public boolean mouseExit(Message msg, Event evt, int x, int y);
   * public boolean keyDown(Message msg, Event evt, int key);
   * public boolean keyUp(Message msg, Event evt, int key);
   * public boolean action(Message msg, Event evt, Object what);
   * public boolean gotFocus(Message msg, Event evt, Object what);
   * public boolean lostFocus(Message msg, Event evt, Object what);
   */
  public boolean handleEvent(Message msg, Event evt) {
    return super.handleEvent(msg, evt);
  }
  // This method calls the SCI Database client
  public boolean action(Message msg, Event evt, Object what) {
    boolean handled = false;
    // Query
    if (msg.target == gui.b1) {
      // test
      //gui.ta.set("text", "This test should demonstrate the Query");
      // test end

      //ByteArrayOutputStream bao = new ByteArrayOutputStream();
      //CommentsClient cc = new CommentsClient ((String)gui.hostTextField.get("text"), 0,
bao);
      //gui.ta.set("text", bao.toString());
    }


    // Get SCI Packet Data
    else if (msg.target == gui.bscipacket) {

    PrintWriter out = new PrintWriter (os, true);

    try {

      Socket s = new Socket ("localhost", DEFAULT_PORT);

      // ObjectOutputStream
      ObjectOutputStream oos = new ObjectOutputStream (s.getOutputStream());

      //oos.writeObject (QueryString);
      oos.writeObject (SCIdataString);

      oos.flush();


      ObjectInputStream  in = new ObjectInputStream (s.getInputStream());
      SCIData_064_255 = (int[]) in.readObject();

      System.out.println ("The GUI-Client SCIData_064_255.length is:" +
Integer.toString(SCIData_064_255.length));
      for (int i = 1; i < SCIData_064_255.length + 1; i++) {
     System.out.println ("Database field " + Integer.toString(i) + " : GUI-Client : " +
Integer.toString(SCIData_064_255[i-1]));
      }
```

```
gui.tfPacketID.set("text", Integer.toString(SCIData_064_255[0]));

gui.tfSCIData_064_065.set("text", IntegerToHEXString(SCIData_064_255[1]));
gui.tfSCIData_066_067.set("text", IntegerToHEXString(SCIData_064_255[2]));
gui.tfSCIData_068_069.set("text", IntegerToHEXString(SCIData_064_255[3]));
gui.tfSCIData_070_071.set("text", IntegerToHEXString(SCIData_064_255[4]));
gui.tfSCIData_072_073.set("text", IntegerToHEXString(SCIData_064_255[5]));
gui.tfSCIData_074_075.set("text", IntegerToHEXString(SCIData_064_255[6]));
gui.tfSCIData_076_077.set("text", IntegerToHEXString(SCIData_064_255[7]));
gui.tfSCIData_078_079.set("text", IntegerToHEXString(SCIData_064_255[8]));
gui.tfSCIData_080_081.set("text", IntegerToHEXString(SCIData_064_255[9]));
gui.tfSCIData_082_083.set("text", IntegerToHEXString(SCIData_064_255[10]));
gui.tfSCIData_084_085.set("text", IntegerToHEXString(SCIData_064_255[11]));
gui.tfSCIData_086_087.set("text", IntegerToHEXString(SCIData_064_255[12]));
gui.tfSCIData_088_089.set("text", IntegerToHEXString(SCIData_064_255[13]));
gui.tfSCIData_090_091.set("text", IntegerToHEXString(SCIData_064_255[14]));
gui.tfSCIData_092_093.set("text", IntegerToHEXString(SCIData_064_255[15]));
gui.tfSCIData_094_095.set("text", IntegerToHEXString(SCIData_064_255[16]));
gui.tfSCIData_096_097.set("text", IntegerToHEXString(SCIData_064_255[17]));
gui.tfSCIData_098_099.set("text", IntegerToHEXString(SCIData_064_255[18]));
gui.tfSCIData_100_101.set("text", IntegerToHEXString(SCIData_064_255[19]));
gui.tfSCIData_102_103.set("text", IntegerToHEXString(SCIData_064_255[20]));
gui.tfSCIData_104_105.set("text", IntegerToHEXString(SCIData_064_255[21]));
gui.tfSCIData_106_107.set("text", IntegerToHEXString(SCIData_064_255[22]));
gui.tfSCIData_108_109.set("text", IntegerToHEXString(SCIData_064_255[23]));
gui.tfSCIData_110_111.set("text", IntegerToHEXString(SCIData_064_255[24]));
gui.tfSCIData_112_113.set("text", IntegerToHEXString(SCIData_064_255[25]));
gui.tfSCIData_114_115.set("text", IntegerToHEXString(SCIData_064_255[26]));
gui.tfSCIData_116_117.set("text", IntegerToHEXString(SCIData_064_255[27]));
gui.tfSCIData_118_119.set("text", IntegerToHEXString(SCIData_064_255[28]));
gui.tfSCIData_120_121.set("text", IntegerToHEXString(SCIData_064_255[29]));
gui.tfSCIData_122_123.set("text", IntegerToHEXString(SCIData_064_255[30]));
gui.tfSCIData_124_125.set("text", IntegerToHEXString(SCIData_064_255[31]));
gui.tfSCIData_126_127.set("text", IntegerToHEXString(SCIData_064_255[32]));
gui.tfSCIData_128_129.set("text", IntegerToHEXString(SCIData_064_255[33]));
gui.tfSCIData_130_131.set("text", IntegerToHEXString(SCIData_064_255[34]));
gui.tfSCIData_132_133.set("text", IntegerToHEXString(SCIData_064_255[35]));
gui.tfSCIData_134_135.set("text", IntegerToHEXString(SCIData_064_255[36]));
gui.tfSCIData_136_137.set("text", IntegerToHEXString(SCIData_064_255[37]));
gui.tfSCIData_138_139.set("text", IntegerToHEXString(SCIData_064_255[38]));
gui.tfSCIData_140_141.set("text", IntegerToHEXString(SCIData_064_255[39]));
gui.tfSCIData_142_143.set("text", IntegerToHEXString(SCIData_064_255[40]));
gui.tfSCIData_144_145.set("text", IntegerToHEXString(SCIData_064_255[41]));
gui.tfSCIData_146_147.set("text", IntegerToHEXString(SCIData_064_255[42]));
gui.tfSCIData_148_149.set("text", IntegerToHEXString(SCIData_064_255[43]));
gui.tfSCIData_150_151.set("text", IntegerToHEXString(SCIData_064_255[44]));
gui.tfSCIData_152_153.set("text", IntegerToHEXString(SCIData_064_255[45]));
gui.tfSCIData_154_155.set("text", IntegerToHEXString(SCIData_064_255[46]));
gui.tfSCIData_156_157.set("text", IntegerToHEXString(SCIData_064_255[47]));
gui.tfSCIData_158_159.set("text", IntegerToHEXString(SCIData_064_255[48]));
gui.tfSCIData_160_161.set("text", IntegerToHEXString(SCIData_064_255[49]));
gui.tfSCIData_162_163.set("text", IntegerToHEXString(SCIData_064_255[50]));
gui.tfSCIData_164_165.set("text", IntegerToHEXString(SCIData_064_255[51]));
gui.tfSCIData_166_167.set("text", IntegerToHEXString(SCIData_064_255[52]));
gui.tfSCIData_168_169.set("text", IntegerToHEXString(SCIData_064_255[53]));
gui.tfSCIData_170_171.set("text", IntegerToHEXString(SCIData_064_255[54]));
gui.tfSCIData_172_173.set("text", IntegerToHEXString(SCIData_064_255[55]));
gui.tfSCIData_174_175.set("text", IntegerToHEXString(SCIData_064_255[56]));
gui.tfSCIData_176_177.set("text", IntegerToHEXString(SCIData_064_255[57]));
gui.tfSCIData_178_179.set("text", IntegerToHEXString(SCIData_064_255[58]));
gui.tfSCIData_180_181.set("text", IntegerToHEXString(SCIData_064_255[59]));
gui.tfSCIData_182_183.set("text", IntegerToHEXString(SCIData_064_255[60]));
gui.tfSCIData_184_185.set("text", IntegerToHEXString(SCIData_064_255[61]));
gui.tfSCIData_186_187.set("text", IntegerToHEXString(SCIData_064_255[62]));
gui.tfSCIData_188_189.set("text", IntegerToHEXString(SCIData_064_255[63]));
gui.tfSCIData_190_191.set("text", IntegerToHEXString(SCIData_064_255[64]));
gui.tfSCIData_192_193.set("text", IntegerToHEXString(SCIData_064_255[65]));
gui.tfSCIData_194_195.set("text", IntegerToHEXString(SCIData_064_255[66]));
gui.tfSCIData_196_197.set("text", IntegerToHEXString(SCIData_064_255[67]));
gui.tfSCIData_198_199.set("text", IntegerToHEXString(SCIData_064_255[68]));
gui.tfSCIData_200_201.set("text", IntegerToHEXString(SCIData_064_255[69]));
gui.tfSCIData_202_203.set("text", IntegerToHEXString(SCIData_064_255[70]));
gui.tfSCIData_204_205.set("text", IntegerToHEXString(SCIData_064_255[71]));
gui.tfSCIData_206_207.set("text", IntegerToHEXString(SCIData_064_255[72]));
gui.tfSCIData_208_209.set("text", IntegerToHEXString(SCIData_064_255[73]));
gui.tfSCIData_210_211.set("text", IntegerToHEXString(SCIData_064_255[74]));
gui.tfSCIData_212_213.set("text", IntegerToHEXString(SCIData_064_255[75]));
gui.tfSCIData_214_215.set("text", IntegerToHEXString(SCIData_064_255[76]));
gui.tfSCIData_216_217.set("text", IntegerToHEXString(SCIData_064_255[77]));
gui.tfSCIData_218_219.set("text", IntegerToHEXString(SCIData_064_255[78]));
gui.tfSCIData_220_221.set("text", IntegerToHEXString(SCIData_064_255[79]));
```

```
      gui.tfSCIData_222_223.set("text", IntegerToHEXString(SCIData_064_255[80]));
      gui.tfSCIData_224_225.set("text", IntegerToHEXString(SCIData_064_255[81]));
      gui.tfSCIData_226_227.set("text", IntegerToHEXString(SCIData_064_255[82]));
      gui.tfSCIData_228_229.set("text", IntegerToHEXString(SCIData_064_255[83]));
      gui.tfSCIData_230_231.set("text", IntegerToHEXString(SCIData_064_255[84]));
      gui.tfSCIData_232_233.set("text", IntegerToHEXString(SCIData_064_255[85]));
      gui.tfSCIData_234_235.set("text", IntegerToHEXString(SCIData_064_255[86]));
      gui.tfSCIData_236_237.set("text", IntegerToHEXString(SCIData_064_255[87]));
      gui.tfSCIData_238_239.set("text", IntegerToHEXString(SCIData_064_255[88]));
      gui.tfSCIData_240_241.set("text", IntegerToHEXString(SCIData_064_255[89]));
      gui.tfSCIData_242_243.set("text", IntegerToHEXString(SCIData_064_255[90]));
      gui.tfSCIData_244_245.set("text", IntegerToHEXString(SCIData_064_255[91]));
      gui.tfSCIData_246_247.set("text", IntegerToHEXString(SCIData_064_255[92]));
      gui.tfSCIData_248_249.set("text", IntegerToHEXString(SCIData_064_255[93]));
      gui.tfSCIData_250_251.set("text", IntegerToHEXString(SCIData_064_255[94]));
      gui.tfSCIData_252_253.set("text", IntegerToHEXString(SCIData_064_255[95]));
      gui.tfSCIData_254_255.set("text", IntegerToHEXString(SCIData_064_255[96]));

      in.close();
      oos.close();

      s.close();


    } catch (IOException e) {

      out.println ("Error querying." + e);

          } catch (ClassNotFoundException e) {

      out.println ("Error querying." + e);

    }
  }


    // Insert
    else if (msg.target == gui.b2) {
      // test
      //gui.ta.set("text", "This test should demonstrate the Insert");
      // test end
      ByteArrayOutputStream bao = new ByteArrayOutputStream();
      CommentsClient cc = new CommentsClient((String)gui.hostTextField.get("text"), 0, bao,
      (String)gui.name.get("text"), (String)gui.user.get("text"),
(String)gui.comments.get("text"));
      gui.ta.set("text",bao.toString());
    }
    // Integer convertion
    else if (msg.target == gui.bint) {

      Integer signedDBint = new Integer((String)gui.textfieldint.get("text"));

      int signedint = signedDBint.intValue();

      Integer bits01_04 = new Integer((signedint)       & 0xF);
      Integer bits05_08 = new Integer((signedint >>  4) & 0xF);
      Integer bits09_12 = new Integer((signedint >>  8) & 0xF);
      Integer bits13_16 = new Integer((signedint >> 12) & 0xF);

      String newSt = bits13_16.toHexString(bits13_16.intValue()) +
            bits09_12.toHexString(bits09_12.intValue()) +
            bits05_08.toHexString(bits05_08.intValue()) +
            bits01_04.toHexString(bits01_04.intValue());

      //if (-32769 < signedint && signedint < 32768) {
    gui.textfielduint.set("text", newSt);
    gui.textfieldhex.set("text", signedDBint.toHexString(signedint));
    gui.textfieldbinary.set("text", signedDBint.toBinaryString(signedint));
      //}
      //else {
      //   gui.textfieldint.set("text", "error");
      //}
    }
    // Unsigned Integer convertion
    else if (msg.target == gui.buint) {
      Integer unsignedDBint = new Integer((String)gui.textfielduint.get("text"));
      long unsignedint = unsignedDBint.longValue();

      BitSet intBits;
      BitSet longBits;

      //gui.textfieldhex.set("text", unsignedDBint.toHexString(unsignedint));
```

```
  gui.textfieldbinary.set("text", gui.textfielduint.get("text"));
  gui.textfieldint.set("text", gui.textfielduint.get("text"));
}

// hex to integer and binary convertion
else if (msg.target == gui.bhex) {

  StringBuffer hexString = new StringBuffer().append(gui.textfieldhex.get("text"));
  int hexToInt = 0;
  short hexToShort = 0;
  short hexToShort01 = 0;
  int stringLoopNumber = 0;

if (hexString.length() < 4) {
  stringLoopNumber = hexString.length();
} else {
  stringLoopNumber = 4;
}

for (int i = 0; i < stringLoopNumber; i++) {

  char ch = hexString.charAt(i);
  int shiftBy =0;

  if (i > 0) {
    shiftBy = 4;
  } else {
    shiftBy = 0;
  }

  if (ch == '0'){        // 0000
    hexToInt = (hexToInt << shiftBy) | 0x0000;
  }
  else if (ch == '1' ) { //0001
     hexToInt = (hexToInt << shiftBy) | 0x0001;
  }
  else if (ch == '2' ) { //0010
    hexToInt = (hexToInt << shiftBy) | 0x0002;
  }
  else if (ch == '3' ) { //0011
     hexToInt = (hexToInt << shiftBy) | 0x0003;
  }
  else if (ch == '4' ) { //0100
    hexToInt = (hexToInt << shiftBy) | 0x0004;
  }
  else if (ch == '5' ) { //0101
    hexToInt = (hexToInt << shiftBy) | 0x0005;
  }
  else if (ch == '6' ) { //0110
    hexToInt = (hexToInt << shiftBy) | 0x0006;
  }
  else if (ch == '7' ) { //0111
    hexToInt = (hexToInt << shiftBy) | 0x0007;
  }
  else if (ch == '8' ) { //1000
    hexToInt = (hexToInt << shiftBy) | 0x0008;
  }
  else if (ch == '9' ) { //1001
    hexToInt = (hexToInt << shiftBy) | 0x0009;
  }
  else if (ch == 'a' | ch == 'A' ) { //1010
    hexToInt = (hexToInt << shiftBy) | 0x000a;
  }
  else if (ch == 'b' | ch == 'B' ) { //1011
    hexToInt = (hexToInt << shiftBy) | 0x000b;
  }
  else if (ch == 'c' | ch == 'C' ) { //1100
    hexToInt = (hexToInt << shiftBy) | 0x000c;
  }
  else if (ch == 'd' | ch == 'D' ) { //1101
    hexToInt = (hexToInt << shiftBy) | 0x000d;
  }
  else if (ch == 'e' | ch == 'E' ) { //1110
    hexToInt = (hexToInt << shiftBy) | 0x000e;
  }
  else if (ch == 'f' | ch == 'F' ) { //1111
    hexToInt = (hexToInt << shiftBy) | 0x000f;
  }
}
  //hexToShort = hexToShort | hexToInt;
```

```
      Integer test01Integer = new Integer(hexToInt);
      Integer test02Integer = new Integer(hexString.length());

      gui.textfieldduint.set("text", test01Integer.toString());
      gui.textfieldint.set("text", test02Integer.toString());
      gui.textfieldbinary.set("text", test01Integer.toBinaryString(hexToInt));

    }

    // Binary convertion
    else if (msg.target == gui.bbinary) {
      gui.textfieldhex.set("text", gui.textfieldbinary.get("text"));
      gui.textfielduint.set("text", gui.textfieldbinary.get("text"));
      gui.textfieldint.set("text", gui.textfieldbinary.get("text"));
    }

    return handled;
    }
    //OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
    public String IntegerToHEXString (int signedint) {


      Integer bits01_04 = new Integer((signedint)       & 0xF);
      Integer bits05_08 = new Integer((signedint >>  4) & 0xF);
      Integer bits09_12 = new Integer((signedint >>  8) & 0xF);
      Integer bits13_16 = new Integer((signedint >> 12) & 0xF);

      String newSt = bits13_16.toHexString(bits13_16.intValue()) +
             bits09_12.toHexString(bits09_12.intValue()) +
             bits05_08.toHexString(bits05_08.intValue()) +
             bits01_04.toHexString(bits01_04.intValue());

      return newSt;
    }
    //OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO

}
```

```
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//                                                          SCIDatabaseMain.java
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

/**
 * This file was automatically generated.  Do not manually modify this file.
 *
 * Runtime vendor: SunSoft, Inc.
 * Runtime version: 1.0
 *
 * Visual vendor: SunSoft, Inc.
 * Visual version: 1.0
 */




import sunsoft.jws.visual.rt.base.Group;
import sunsoft.jws.visual.rt.base.MainHelper;
import java.applet.Applet;

/**
 * Generated Main class
 *
 * @version 1.20, 05/21/96
 */
public class SCIDatabaseMain extends Applet {
  /**
   * Helper class for the generated main class.  This variable is only
   * used when we are running as an applet.
   */
  private MainHelper helper;

  /**
   * Called when application is run from the command line.
   */
  public static void main(String args[]) {
    MainHelper helper = new MainHelper();
    helper.checkVersion(1.0);

    Group group = new SCIDatabase();
    helper.main(group, args);
  }
```

```
  /**
   * Called when the applet is loaded.
   */
  public void init() {
    helper = new MainHelper();
    helper.checkVersion(1.0);

    Group group = new SCIDatabase();
    helper.init(this, group);
  }

  /**
   * Called whenever the applet's page is visited.
   */
  public void start(){
    helper.start();
  }

  /**
   * Called by the browser when the user leaves the page.
   */
  public void stop() {
    helper.stop();
  }

  /**
   * Called by the browser when the applet should be destroyed.
   */
  public void destroy() {
    helper.destroy();
  }
}

//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//                                                      SCIDatabaseRoot.java
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

 * This file was automatically generated.  Do not manually modify this file.
 *
 * Runtime vendor: SunSoft, Inc.
 * Runtime version: 1.0
 *
 * Visual vendor: SunSoft, Inc.
```

```
 * Visual version: 1.0
 */


import sunsoft.jws.visual.rt.awt.GBConstraints;
import sunsoft.jws.visual.rt.base.*;
import sunsoft.jws.visual.rt.shadow.*;
import sunsoft.jws.visual.rt.shadow.java.awt.*;
import sunsoft.jws.visual.rt.type.*;

import sunsoft.jws.visual.rt.base.Root;
import sunsoft.jws.visual.rt.shadow.java.awt.FrameShadow;
import sunsoft.jws.visual.rt.shadow.GBPanelShadow;
import sunsoft.jws.visual.rt.shadow.TabbedFolderShadow;
import sunsoft.jws.visual.rt.shadow.java.awt.TextAreaShadow;
import sunsoft.jws.visual.rt.shadow.java.awt.ButtonShadow;
import sunsoft.jws.visual.rt.shadow.java.awt.TextFieldShadow;
import sunsoft.jws.visual.rt.shadow.java.awt.LabelShadow;
import sunsoft.jws.visual.rt.shadow.LabelBarShadow;

public class SCIDatabaseRoot extends Root {
  public ButtonShadow b1;
  public ButtonShadow b2;
  public ButtonShadow bbinary;
  public ButtonShadow bhex;
  public ButtonShadow bint;
  public ButtonShadow bscipacket;
  public ButtonShadow buint;
  public ButtonShadow button1;
  public ButtonShadow button2;
  public ButtonShadow button4;
  public ButtonShadow button5;
  public TextFieldShadow comments;
  public LabelShadow databasehost;
  public GBPanelShadow gbpanel1;
  public GBPanelShadow gbpanel19;
  public GBPanelShadow gbpanel2;
  public GBPanelShadow gbpanel20;
  public GBPanelShadow gbpanel22;
  public GBPanelShadow gbpanel23;
  public GBPanelShadow gbpanel3;
```

```
public GBPanelShadow gbpanel4;
public GBPanelShadow gbpanel5;
public GBPanelShadow gbpanel6;
public GBPanelShadow gbpanel7;
public GBPanelShadow gbpanel8;
public TextFieldShadow hostTextField;
public LabelShadow label1;
public LabelShadow label10;
public LabelShadow label11;
public LabelShadow label12;
public LabelShadow label13;
public LabelShadow label14;
public LabelShadow label15;
public LabelShadow label16;
public LabelShadow label17;
public LabelShadow label18;
public LabelShadow label19;
public LabelShadow label2;
public LabelShadow label20;
public LabelShadow label21;
public LabelShadow label22;
public LabelShadow label23;
public LabelShadow label24;
public LabelShadow label25;
public LabelShadow label26;
public LabelShadow label28;
public LabelShadow label29;
public LabelShadow label3;
public LabelShadow label30;
public LabelShadow label31;
public LabelShadow label32;
public LabelShadow label33;
public LabelShadow label34;
public LabelShadow label35;
public LabelShadow label36;
public LabelShadow label37;
public LabelShadow label38;
public LabelShadow label39;
public LabelShadow label4;
public LabelShadow label42;
public LabelShadow label43;
```

```
public LabelShadow label44;
public LabelShadow label45;
public LabelShadow label46;
public LabelShadow label47;
public LabelShadow label48;
public LabelShadow label49;
public LabelShadow label5;
public LabelShadow label50;
public LabelShadow label51;
public LabelShadow label6;
public LabelShadow label7;
public LabelShadow label8;
public LabelShadow label9;
public LabelBarShadow labelbar1;
public LabelBarShadow labelbar2;
public LabelBarShadow labelbar3;
public LabelBarShadow labelbar4;
public LabelBarShadow labelbar5;
public TextFieldShadow name;
public Root root1;
public TextAreaShadow ta;
public TabbedFolderShadow tabbedfolder1;
public TextFieldShadow textfield1;
public TextFieldShadow textfield131;
public TextFieldShadow textfield132;
public TextFieldShadow textfield133;
public TextFieldShadow textfield134;
public TextFieldShadow textfield135;
public TextFieldShadow textfield136;
public TextFieldShadow textfield137;
public TextFieldShadow textfield138;
public TextFieldShadow textfield140;
public TextFieldShadow textfield141;
public TextFieldShadow textfield142;
public TextFieldShadow textfield143;
public TextFieldShadow textfield144;
public TextFieldShadow textfield145;
public TextFieldShadow textfield146;
public TextFieldShadow textfield147;
public TextFieldShadow textfield148;
public TextFieldShadow textfield149;
```

```
public TextFieldShadow textfield150;
public TextFieldShadow textfield151;
public TextFieldShadow textfield18;
public TextFieldShadow textfield19;
public TextFieldShadow textfield2;
public TextFieldShadow textfield20;
public TextFieldShadow textfield21;
public TextFieldShadow textfield22;
public TextFieldShadow textfield23;
public TextFieldShadow textfield24;
public TextFieldShadow textfield25;
public TextFieldShadow textfield3;
public TextFieldShadow textfield34;
public TextFieldShadow textfield35;
public TextFieldShadow textfield4;
public TextFieldShadow textfield49;
public TextFieldShadow textfield5;
public TextFieldShadow textfield50;
public TextFieldShadow textfield51;
public TextFieldShadow textfield52;
public TextFieldShadow textfield53;
public TextFieldShadow textfield54;
public TextFieldShadow textfield55;
public TextFieldShadow textfield6;
public TextFieldShadow textfield64;
public TextFieldShadow textfield65;
public TextFieldShadow textfield66;
public TextFieldShadow textfield67;
public TextFieldShadow textfield68;
public TextFieldShadow textfield69;
public TextFieldShadow textfield7;
public TextFieldShadow textfield70;
public TextFieldShadow textfield8;
public TextFieldShadow textfieldbinary;
public TextFieldShadow textfieldhex;
public TextFieldShadow textfieldint;
public TextFieldShadow textfielduint;
public TextFieldShadow tfPacketID;
public TextFieldShadow tfSCIData_064_065;
public TextFieldShadow tfSCIData_066_067;
public TextFieldShadow tfSCIData_068_069;
```

```
public TextFieldShadow tfSCIData_070_071;
public TextFieldShadow tfSCIData_072_073;
public TextFieldShadow tfSCIData_074_075;
public TextFieldShadow tfSCIData_076_077;
public TextFieldShadow tfSCIData_078_079;
public TextFieldShadow tfSCIData_080_081;
public TextFieldShadow tfSCIData_082_083;
public TextFieldShadow tfSCIData_084_085;
public TextFieldShadow tfSCIData_086_087;
public TextFieldShadow tfSCIData_088_089;
public TextFieldShadow tfSCIData_090_091;
public TextFieldShadow tfSCIData_092_093;
public TextFieldShadow tfSCIData_094_095;
public TextFieldShadow tfSCIData_096_097;
public TextFieldShadow tfSCIData_098_099;
public TextFieldShadow tfSCIData_100_101;
public TextFieldShadow tfSCIData_102_103;
public TextFieldShadow tfSCIData_104_105;
public TextFieldShadow tfSCIData_106_107;
public TextFieldShadow tfSCIData_108_109;
public TextFieldShadow tfSCIData_110_111;
public TextFieldShadow tfSCIData_112_113;
public TextFieldShadow tfSCIData_114_115;
public TextFieldShadow tfSCIData_116_117;
public TextFieldShadow tfSCIData_118_119;
public TextFieldShadow tfSCIData_120_121;
public TextFieldShadow tfSCIData_122_123;
public TextFieldShadow tfSCIData_124_125;
public TextFieldShadow tfSCIData_126_127;
public TextFieldShadow tfSCIData_128_129;
public TextFieldShadow tfSCIData_130_131;
public TextFieldShadow tfSCIData_132_133;
public TextFieldShadow tfSCIData_134_135;
public TextFieldShadow tfSCIData_136_137;
public TextFieldShadow tfSCIData_138_139;
public TextFieldShadow tfSCIData_140_141;
public TextFieldShadow tfSCIData_142_143;
public TextFieldShadow tfSCIData_144_145;
public TextFieldShadow tfSCIData_146_147;
public TextFieldShadow tfSCIData_148_149;
public TextFieldShadow tfSCIData_150_151;
```

```
public TextFieldShadow tfSCIData_152_153;
public TextFieldShadow tfSCIData_154_155;
public TextFieldShadow tfSCIData_156_157;
public TextFieldShadow tfSCIData_158_159;
public TextFieldShadow tfSCIData_160_161;
public TextFieldShadow tfSCIData_162_163;
public TextFieldShadow tfSCIData_164_165;
public TextFieldShadow tfSCIData_166_167;
public TextFieldShadow tfSCIData_168_169;
public TextFieldShadow tfSCIData_170_171;
public TextFieldShadow tfSCIData_172_173;
public TextFieldShadow tfSCIData_174_175;
public TextFieldShadow tfSCIData_176_177;
public TextFieldShadow tfSCIData_178_179;
public TextFieldShadow tfSCIData_180_181;
public TextFieldShadow tfSCIData_182_183;
public TextFieldShadow tfSCIData_184_185;
public TextFieldShadow tfSCIData_186_187;
public TextFieldShadow tfSCIData_188_189;
public TextFieldShadow tfSCIData_190_191;
public TextFieldShadow tfSCIData_192_193;
public TextFieldShadow tfSCIData_194_195;
public TextFieldShadow tfSCIData_196_197;
public TextFieldShadow tfSCIData_198_199;
public TextFieldShadow tfSCIData_200_201;
public TextFieldShadow tfSCIData_202_203;
public TextFieldShadow tfSCIData_204_205;
public TextFieldShadow tfSCIData_206_207;
public TextFieldShadow tfSCIData_208_209;
public TextFieldShadow tfSCIData_210_211;
public TextFieldShadow tfSCIData_212_213;
public TextFieldShadow tfSCIData_214_215;
public TextFieldShadow tfSCIData_216_217;
public TextFieldShadow tfSCIData_218_219;
public TextFieldShadow tfSCIData_220_221;
public TextFieldShadow tfSCIData_222_223;
public TextFieldShadow tfSCIData_224_225;
public TextFieldShadow tfSCIData_226_227;
public TextFieldShadow tfSCIData_228_229;
public TextFieldShadow tfSCIData_230_231;
public TextFieldShadow tfSCIData_232_233;
```

```
  public TextFieldShadow tfSCIData_234_235;
  public TextFieldShadow tfSCIData_236_237;
  public TextFieldShadow tfSCIData_238_239;
  public TextFieldShadow tfSCIData_240_241;
  public TextFieldShadow tfSCIData_242_243;
  public TextFieldShadow tfSCIData_244_245;
  public TextFieldShadow tfSCIData_246_247;
  public TextFieldShadow tfSCIData_248_249;
  public TextFieldShadow tfSCIData_250_251;
  public TextFieldShadow tfSCIData_252_253;
  public TextFieldShadow tfSCIData_254_255;
  public LabelShadow uHex;
  public LabelShadow uInt;
  public TextFieldShadow user;

  public SCIDatabaseRoot(Group group) {
    setGroup(group);

    gbpanel1 = new GBPanelShadow();
    gbpanel1.set("name", "gbpanel1");
    add(gbpanel1);
    {
      int _tmp[] = {14,14};
      gbpanel1.set("rowHeights", _tmp);
    }
    gbpanel1.set("GBConstraints", new GBConstraints("x=0;y=0;fill=both"));
    {
      int _tmp[] = {14};
      gbpanel1.set("columnWidths", _tmp);
    }
    gbpanel1.set("borderRelief", new sunsoft.jws.visual.rt.type.ReliefEnum("win95 field
border"));
    gbpanel1.set("borderLabel", "");
    gbpanel1.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
    {
      double _tmp[] = {0.0,0.0};
      gbpanel1.set("rowWeights", _tmp);
    }
    gbpanel1.set("layoutSize", new java.awt.Dimension(948, 722));
    {
      double _tmp[] = {0.0};
```

```
  gbpanel1.set("columnWeights", _tmp);
}
gbpanel1.set("layoutLocation", new java.awt.Point(54, 49));

tabbedfolder1 = new TabbedFolderShadow();
tabbedfolder1.set("name", "tabbedfolder1");
gbpanel1.add(tabbedfolder1);
tabbedfolder1.set("borderWidth", new Integer(0));
tabbedfolder1.set("GBConstraints", new GBConstraints("x=0;y=0;fill=both"));

gbpanel2 = new GBPanelShadow();
gbpanel2.set("name", "gbpanel2");
tabbedfolder1.add(gbpanel2);
{
  int _tmp[] = {14};
  gbpanel2.set("rowHeights", _tmp);
}
{
  int _tmp[] = {14};
  gbpanel2.set("columnWidths", _tmp);
}
gbpanel2.set("layoutName", "Database Info");
{
  double _tmp[] = {0.0};
  gbpanel2.set("rowWeights", _tmp);
}
{
  double _tmp[] = {0.0};
  gbpanel2.set("columnWeights", _tmp);
}

gbpanel3 = new GBPanelShadow();
gbpanel3.set("name", "gbpanel3");
tabbedfolder1.add(gbpanel3);
{
  int _tmp[] = {14,14,14,14,14,14,14};
  gbpanel3.set("rowHeights", _tmp);
}
{
  int _tmp[] = {14,14,14,14};
  gbpanel3.set("columnWidths", _tmp);
```

```
}
gbpanel3.set("layoutName", "Database Config");
gbpanel3.set("visible", Boolean.FALSE);
{
  double _tmp[] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0};
  gbpanel3.set("rowWeights", _tmp);
}
{
  double _tmp[] = {0.0,0.0,0.0,0.0};
  gbpanel3.set("columnWeights", _tmp);
}

ta = new TextAreaShadow();
ta.set("name", "ta");
gbpanel3.add(ta);
ta.set("GBConstraints", new GBConstraints("x=1;y=0;width=3;fill=both"));
ta.set("editable", Boolean.FALSE);

b1 = new ButtonShadow();
b1.set("name", "b1");
gbpanel3.add(b1);
b1.set("GBConstraints", new GBConstraints("x=1;y=1;width=3"));
b1.set("text", "Query");

b2 = new ButtonShadow();
b2.set("name", "b2");
gbpanel3.add(b2);
b2.set("GBConstraints", new GBConstraints("x=1;y=3;width=3"));
b2.set("text", "Insert");

name = new TextFieldShadow();
name.set("name", "name");
gbpanel3.add(name);
name.set("GBConstraints", new GBConstraints("x=1;y=2;fill=horizontal"));

user = new TextFieldShadow();
user.set("name", "user");
gbpanel3.add(user);
user.set("GBConstraints", new GBConstraints("x=2;y=2;fill=horizontal"));

comments = new TextFieldShadow();
```

```
comments.set("name", "comments");
gbpanel3.add(comments);
comments.set("GBConstraints", new GBConstraints("x=3;y=2;fill=horizontal"));
comments.set("numColumns", new Integer(50));

uInt = new LabelShadow();
uInt.set("name", "uInt");
gbpanel3.add(uInt);
uInt.set("GBConstraints", new GBConstraints("x=1;y=4"));
uInt.set("text", "unsigned integer ");

uHex = new LabelShadow();
uHex.set("name", "uHex");
gbpanel3.add(uHex);
uHex.set("GBConstraints", new GBConstraints("x=2;y=4"));
uHex.set("text", "hex");

textfielduint = new TextFieldShadow();
textfielduint.set("name", "textfielduint");
gbpanel3.add(textfielduint);
textfielduint.set("GBConstraints", new GBConstraints("x=1;y=5;fill=horizontal"));
textfielduint.set("numColumns", new Integer(6));

textfieldhex = new TextFieldShadow();
textfieldhex.set("name", "textfieldhex");
gbpanel3.add(textfieldhex);
textfieldhex.set("GBConstraints", new GBConstraints("x=2;y=5;fill=horizontal"));
textfieldhex.set("text", "ddff");
textfieldhex.set("numColumns", new Integer(6));

buint = new ButtonShadow();
buint.set("name", "buint");
gbpanel3.add(buint);
buint.set("GBConstraints", new GBConstraints("x=1;y=6"));
buint.set("text", "convert");

bhex = new ButtonShadow();
bhex.set("name", "bhex");
gbpanel3.add(bhex);
bhex.set("GBConstraints", new GBConstraints("x=2;y=6"));
bhex.set("text", "convert");
```

```
textfieldbinary = new TextFieldShadow();
textfieldbinary.set("name", "textfieldbinary");
gbpanel3.add(textfieldbinary);
textfieldbinary.set("GBConstraints", new GBConstraints("x=3;y=5;fill=horizontal"));
textfieldbinary.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
textfieldbinary.set("numColumns", new Integer(17));

bbinary = new ButtonShadow();
bbinary.set("name", "bbinary");
gbpanel3.add(bbinary);
bbinary.set("GBConstraints", new GBConstraints("x=3;y=6"));
bbinary.set("text", "convert");

label29 = new LabelShadow();
label29.set("name", "label29");
gbpanel3.add(label29);
label29.set("GBConstraints", new GBConstraints("x=3;y=4"));
label29.set("text", "binary");

label30 = new LabelShadow();
label30.set("name", "label30");
gbpanel3.add(label30);
label30.set("GBConstraints", new GBConstraints("x=0;y=4"));
label30.set("text", "integer");

textfieldint = new TextFieldShadow();
textfieldint.set("name", "textfieldint");
gbpanel3.add(textfieldint);
textfieldint.set("GBConstraints", new GBConstraints("x=0;y=5;fill=horizontal"));
textfieldint.set("text", " ");
textfieldint.set("numColumns", new Integer(7));

bint = new ButtonShadow();
bint.set("name", "bint");
gbpanel3.add(bint);
bint.set("GBConstraints", new GBConstraints("x=0;y=6"));
bint.set("text", "convert");

gbpanel4 = new GBPanelShadow();
gbpanel4.set("name", "gbpanel4");
```

```
tabbedfolder1.add(gbpanel4);
{
  int _tmp[] = {14};
  gbpanel4.set("rowHeights", _tmp);
}
{
  int _tmp[] = {14};
  gbpanel4.set("columnWidths", _tmp);
}
gbpanel4.set("layoutName", "SCI Packet Builder");
gbpanel4.set("visible", Boolean.FALSE);
{
  double _tmp[] = {0.0};
  gbpanel4.set("rowWeights", _tmp);
}
{
  double _tmp[] = {0.0};
  gbpanel4.set("columnWeights", _tmp);
}

gbpanel5 = new GBPanelShadow();
gbpanel5.set("name", "gbpanel5");
tabbedfolder1.add(gbpanel5);
{
  int _tmp[] = {14};
  gbpanel5.set("rowHeights", _tmp);
}
{
  int _tmp[] = {14};
  gbpanel5.set("columnWidths", _tmp);
}
gbpanel5.set("layoutName", "SCI Trigger Builder");
gbpanel5.set("visible", Boolean.FALSE);
{
  double _tmp[] = {0.0};
  gbpanel5.set("rowWeights", _tmp);
}
{
  double _tmp[] = {0.0};
  gbpanel5.set("columnWeights", _tmp);
}
```

```
    gbpanel6 = new GBPanelShadow();
    gbpanel6.set("name", "gbpanel6");
    tabbedfolder1.add(gbpanel6);
    {
      int _tmp[] = {14};
      gbpanel6.set("rowHeights", _tmp);
    }
    {
      int _tmp[] = {14,14};
      gbpanel6.set("columnWidths", _tmp);
    }
    gbpanel6.set("layoutName", "SCI Packet Viewer");
    gbpanel6.set("visible", Boolean.FALSE);
    {
      double _tmp[] = {0.0};
      gbpanel6.set("rowWeights", _tmp);
    }
    {
      double _tmp[] = {0.0,0.0};
      gbpanel6.set("columnWeights", _tmp);
    }

    gbpanel20 = new GBPanelShadow();
    gbpanel20.set("name", "gbpanel20");
    gbpanel6.add(gbpanel20);
    {
      int _tmp[] = {14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14};
      gbpanel20.set("rowHeights", _tmp);
    }
    gbpanel20.set("borderWidth", new Integer(0));
    gbpanel20.set("GBConstraints", new GBConstraints("x=0;y=0;fill=both"));
    {
      int _tmp[] = {14,14,14,14,14,14,14,14,14,14,14};
      gbpanel20.set("columnWidths", _tmp);
    }
    {
      double _tmp[] =
{0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
      gbpanel20.set("rowWeights", _tmp);
    }
```

```
{
  double _tmp[] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
  gbpanel20.set("columnWeights", _tmp);
}

textfield1 = new TextFieldShadow();
textfield1.set("name", "textfield1");
gbpanel20.add(textfield1);
textfield1.set("GBConstraints", new GBConstraints("x=3;y=3;fill=horizontal"));
textfield1.set("text", "ac61");
textfield1.set("background", convert("java.awt.Color", "yellow"));
textfield1.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield1.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield1.set("numColumns", new Integer(3));

textfield2 = new TextFieldShadow();
textfield2.set("name", "textfield2");
gbpanel20.add(textfield2);
textfield2.set("GBConstraints", new GBConstraints("x=4;y=3;fill=horizontal"));
textfield2.set("text", "6eff");
textfield2.set("background", convert("java.awt.Color", "yellow"));
textfield2.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield2.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield2.set("numColumns", new Integer(3));

textfield3 = new TextFieldShadow();
textfield3.set("name", "textfield3");
gbpanel20.add(textfield3);
textfield3.set("GBConstraints", new GBConstraints("x=5;y=3;fill=horizontal"));
textfield3.set("text", "fb56");
textfield3.set("background", convert("java.awt.Color", "yellow"));
textfield3.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield3.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield3.set("numColumns", new Integer(3));

textfield4 = new TextFieldShadow();
textfield4.set("name", "textfield4");
gbpanel20.add(textfield4);
textfield4.set("GBConstraints", new GBConstraints("x=6;y=3;fill=horizontal"));
textfield4.set("text", "4298");
textfield4.set("background", convert("java.awt.Color", "yellow"));
```

```
textfield4.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield4.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield4.set("numColumns", new Integer(3));

textfield5 = new TextFieldShadow();
textfield5.set("name", "textfield5");
gbpanel20.add(textfield5);
textfield5.set("GBConstraints", new GBConstraints("x=7;y=3;fill=horizontal"));
textfield5.set("text", "efff");
textfield5.set("background", convert("java.awt.Color", "yellow"));
textfield5.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield5.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield5.set("numColumns", new Integer(3));

textfield6 = new TextFieldShadow();
textfield6.set("name", "textfield6");
gbpanel20.add(textfield6);
textfield6.set("GBConstraints", new GBConstraints("x=8;y=3;fill=horizontal"));
textfield6.set("text", "6df3");
textfield6.set("background", convert("java.awt.Color", "yellow"));
textfield6.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield6.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield6.set("numColumns", new Integer(3));

textfield7 = new TextFieldShadow();
textfield7.set("name", "textfield7");
gbpanel20.add(textfield7);
textfield7.set("GBConstraints", new GBConstraints("x=9;y=3;fill=horizontal"));
textfield7.set("text", "78e4");
textfield7.set("background", convert("java.awt.Color", "yellow"));
textfield7.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield7.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield7.set("numColumns", new Integer(3));

textfield8 = new TextFieldShadow();
textfield8.set("name", "textfield8");
gbpanel20.add(textfield8);
textfield8.set("GBConstraints", new GBConstraints("x=10;y=3;fill=horizontal"));
textfield8.set("text", "3232");
textfield8.set("background", convert("java.awt.Color", "yellow"));
textfield8.set("insets", new java.awt.Insets(0, 0, 0, 0));
```

```
    textfield8.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield8.set("numColumns", new Integer(3));


    tfSCIData_066_067 = new TextFieldShadow();
    tfSCIData_066_067.set("name", "tfSCIData_066_067");
    gbpanel20.add(tfSCIData_066_067);
    tfSCIData_066_067.set("GBConstraints", new GBConstraints("x=4;y=7;fill=horizontal"));
    tfSCIData_066_067.set("text", "####");
    tfSCIData_066_067.set("background", convert("java.awt.Color", "green"));
    tfSCIData_066_067.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_066_067.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_066_067.set("numColumns", new Integer(3));


    tfSCIData_068_069 = new TextFieldShadow();
    tfSCIData_068_069.set("name", "tfSCIData_068_069");
    gbpanel20.add(tfSCIData_068_069);
    tfSCIData_068_069.set("GBConstraints", new GBConstraints("x=5;y=7;fill=horizontal"));
    tfSCIData_068_069.set("text", "####");
    tfSCIData_068_069.set("background", convert("java.awt.Color", "green"));
    tfSCIData_068_069.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_068_069.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_068_069.set("numColumns", new Integer(3));


    tfSCIData_070_071 = new TextFieldShadow();
    tfSCIData_070_071.set("name", "tfSCIData_070_071");
    gbpanel20.add(tfSCIData_070_071);
    tfSCIData_070_071.set("GBConstraints", new GBConstraints("x=6;y=7;fill=horizontal"));
    tfSCIData_070_071.set("text", "####");
    tfSCIData_070_071.set("background", convert("java.awt.Color", "green"));
    tfSCIData_070_071.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_070_071.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_070_071.set("numColumns", new Integer(3));


    tfSCIData_072_073 = new TextFieldShadow();
    tfSCIData_072_073.set("name", "tfSCIData_072_073");
    gbpanel20.add(tfSCIData_072_073);
    tfSCIData_072_073.set("GBConstraints", new GBConstraints("x=7;y=7;fill=horizontal"));
    tfSCIData_072_073.set("text", "####");
    tfSCIData_072_073.set("background", convert("java.awt.Color", "green"));
```

```
    tfSCIData_072_073.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_072_073.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_072_073.set("numColumns", new Integer(3));


    tfSCIData_074_075 = new TextFieldShadow();

    tfSCIData_074_075.set("name", "tfSCIData_074_075");

    gbpanel20.add(tfSCIData_074_075);

    tfSCIData_074_075.set("GBConstraints", new GBConstraints("x=8;y=7;fill=horizontal"));

    tfSCIData_074_075.set("text", "####");

    tfSCIData_074_075.set("background", convert("java.awt.Color", "green"));

    tfSCIData_074_075.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_074_075.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_074_075.set("numColumns", new Integer(3));


    tfSCIData_076_077 = new TextFieldShadow();

    tfSCIData_076_077.set("name", "tfSCIData_076_077");

    gbpanel20.add(tfSCIData_076_077);

    tfSCIData_076_077.set("GBConstraints", new GBConstraints("x=9;y=7;fill=horizontal"));

    tfSCIData_076_077.set("text", "####");

    tfSCIData_076_077.set("background", convert("java.awt.Color", "green"));

    tfSCIData_076_077.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_076_077.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_076_077.set("numColumns", new Integer(3));


    tfSCIData_126_127 = new TextFieldShadow();

    tfSCIData_126_127.set("name", "tfSCIData_126_127");

    gbpanel20.add(tfSCIData_126_127);

    tfSCIData_126_127.set("GBConstraints", new GBConstraints("x=10;y=10;fill=horizontal"));

    tfSCIData_126_127.set("text", "####");

    tfSCIData_126_127.set("background", convert("java.awt.Color", "green"));

    tfSCIData_126_127.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_126_127.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_126_127.set("numColumns", new Integer(3));


    tfSCIData_136_137 = new TextFieldShadow();

    tfSCIData_136_137.set("name", "tfSCIData_136_137");

    gbpanel20.add(tfSCIData_136_137);

    tfSCIData_136_137.set("GBConstraints", new GBConstraints("x=7;y=11;fill=horizontal"));

    tfSCIData_136_137.set("text", "####");
```

```
    tfSCIData_136_137.set("background", convert("java.awt.Color", "green"));
    tfSCIData_136_137.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_136_137.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_136_137.set("numColumns", new Integer(3));

    textfield18 = new TextFieldShadow();
    textfield18.set("name", "textfield18");
    gbpanel20.add(textfield18);
    textfield18.set("GBConstraints", new GBConstraints("x=3;y=4;fill=horizontal"));
    textfield18.set("text", "ad3d");
    textfield18.set("background", convert("java.awt.Color", "springgreen"));
    textfield18.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield18.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield18.set("numColumns", new Integer(3));

    textfield19 = new TextFieldShadow();
    textfield19.set("name", "textfield19");
    gbpanel20.add(textfield19);
    textfield19.set("GBConstraints", new GBConstraints("x=4;y=4;fill=horizontal"));
    textfield19.set("text", "b345");
    textfield19.set("background", convert("java.awt.Color", "springgreen"));
    textfield19.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield19.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield19.set("numColumns", new Integer(3));

    textfield20 = new TextFieldShadow();
    textfield20.set("name", "textfield20");
    gbpanel20.add(textfield20);
    textfield20.set("GBConstraints", new GBConstraints("x=5;y=4;fill=horizontal"));
    textfield20.set("text", "5ef1");
    textfield20.set("background", convert("java.awt.Color", "springgreen"));
    textfield20.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield20.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield20.set("numColumns", new Integer(3));

    textfield21 = new TextFieldShadow();
    textfield21.set("name", "textfield21");
    gbpanel20.add(textfield21);
    textfield21.set("GBConstraints", new GBConstraints("x=6;y=4;fill=horizontal"));
    textfield21.set("text", "76cd");
```

```
textfield21.set("background", convert("java.awt.Color", "springgreen"));
textfield21.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield21.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield21.set("numColumns", new Integer(3));

textfield22 = new TextFieldShadow();
textfield22.set("name", "textfield22");
gbpanel20.add(textfield22);
textfield22.set("GBConstraints", new GBConstraints("x=7;y=4;fill=horizontal"));
textfield22.set("text", "dc43");
textfield22.set("background", convert("java.awt.Color", "springgreen"));
textfield22.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield22.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield22.set("numColumns", new Integer(3));

textfield23 = new TextFieldShadow();
textfield23.set("name", "textfield23");
gbpanel20.add(textfield23);
textfield23.set("GBConstraints", new GBConstraints("x=8;y=4;fill=horizontal"));
textfield23.set("text", "ffab");
textfield23.set("background", convert("java.awt.Color", "springgreen"));
textfield23.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield23.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield23.set("numColumns", new Integer(3));

textfield24 = new TextFieldShadow();
textfield24.set("name", "textfield24");
gbpanel20.add(textfield24);
textfield24.set("GBConstraints", new GBConstraints("x=9;y=4;fill=horizontal"));
textfield24.set("text", "52fe");
textfield24.set("background", convert("java.awt.Color", "springgreen"));
textfield24.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield24.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield24.set("numColumns", new Integer(3));

textfield25 = new TextFieldShadow();
textfield25.set("name", "textfield25");
gbpanel20.add(textfield25);
textfield25.set("GBConstraints", new GBConstraints("x=10;y=4;fill=horizontal"));
textfield25.set("text", "4319");
textfield25.set("background", convert("java.awt.Color", "springgreen"));
```

```
    textfield25.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield25.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield25.set("numColumns", new Integer(3));


    tfSCIData_078_079 = new TextFieldShadow();
    tfSCIData_078_079.set("name", "tfSCIData_078_079");
    gbpanel20.add(tfSCIData_078_079);
    tfSCIData_078_079.set("GBConstraints", new GBConstraints("x=10;y=7;fill=horizontal"));
    tfSCIData_078_079.set("text", "####");
    tfSCIData_078_079.set("background", convert("java.awt.Color", "green"));
    tfSCIData_078_079.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_078_079.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_078_079.set("numColumns", new Integer(3));


    tfSCIData_086_087 = new TextFieldShadow();
    tfSCIData_086_087.set("name", "tfSCIData_086_087");
    gbpanel20.add(tfSCIData_086_087);
    tfSCIData_086_087.set("GBConstraints", new GBConstraints("x=6;y=8;fill=horizontal"));
    tfSCIData_086_087.set("text", "####");
    tfSCIData_086_087.set("background", convert("java.awt.Color", "green"));
    tfSCIData_086_087.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_086_087.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_086_087.set("numColumns", new Integer(3));


    tfSCIData_090_091 = new TextFieldShadow();
    tfSCIData_090_091.set("name", "tfSCIData_090_091");
    gbpanel20.add(tfSCIData_090_091);
    tfSCIData_090_091.set("GBConstraints", new GBConstraints("x=8;y=8;fill=horizontal"));
    tfSCIData_090_091.set("text", "####");
    tfSCIData_090_091.set("background", convert("java.awt.Color", "green"));
    tfSCIData_090_091.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_090_091.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_090_091.set("numColumns", new Integer(3));


    tfSCIData_100_101 = new TextFieldShadow();
    tfSCIData_100_101.set("name", "tfSCIData_100_101");
    gbpanel20.add(tfSCIData_100_101);
    tfSCIData_100_101.set("GBConstraints", new GBConstraints("x=5;y=9;fill=horizontal"));
    tfSCIData_100_101.set("text", "####");
```

```
    tfSCIData_100_101.set("background", convert("java.awt.Color", "green"));

    tfSCIData_100_101.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_100_101.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_100_101.set("numColumns", new Integer(3));


    tfSCIData_106_107 = new TextFieldShadow();

    tfSCIData_106_107.set("name", "tfSCIData_106_107");

    gbpanel20.add(tfSCIData_106_107);

    tfSCIData_106_107.set("GBConstraints", new GBConstraints("x=8;y=9;fill=horizontal"));

    tfSCIData_106_107.set("text", "####");

    tfSCIData_106_107.set("background", convert("java.awt.Color", "green"));

    tfSCIData_106_107.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_106_107.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_106_107.set("numColumns", new Integer(3));


    tfSCIData_120_121 = new TextFieldShadow();

    tfSCIData_120_121.set("name", "tfSCIData_120_121");

    gbpanel20.add(tfSCIData_120_121);

    tfSCIData_120_121.set("GBConstraints", new GBConstraints("x=7;y=10;fill=horizontal"));

    tfSCIData_120_121.set("text", "####");

    tfSCIData_120_121.set("background", convert("java.awt.Color", "green"));

    tfSCIData_120_121.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_120_121.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_120_121.set("numColumns", new Integer(3));


    tfSCIData_124_125 = new TextFieldShadow();

    tfSCIData_124_125.set("name", "tfSCIData_124_125");

    gbpanel20.add(tfSCIData_124_125);

    tfSCIData_124_125.set("GBConstraints", new GBConstraints("x=9;y=10;fill=horizontal"));

    tfSCIData_124_125.set("text", "####");

    tfSCIData_124_125.set("background", convert("java.awt.Color", "green"));

    tfSCIData_124_125.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_124_125.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_124_125.set("numColumns", new Integer(3));


    tfSCIData_138_139 = new TextFieldShadow();

    tfSCIData_138_139.set("name", "tfSCIData_138_139");

    gbpanel20.add(tfSCIData_138_139);

    tfSCIData_138_139.set("GBConstraints", new GBConstraints("x=8;y=11;fill=horizontal"));
```

```
    tfSCIData_138_139.set("text", "####");
    tfSCIData_138_139.set("background", convert("java.awt.Color", "green"));
    tfSCIData_138_139.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_138_139.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_138_139.set("numColumns", new Integer(3));


    textfield34 = new TextFieldShadow();
    textfield34.set("name", "textfield34");
    gbpanel20.add(textfield34);
    textfield34.set("GBConstraints", new GBConstraints("x=3;y=5;fill=horizontal"));
    textfield34.set("text", "eebe");
    textfield34.set("background", convert("java.awt.Color", "springgreen"));
    textfield34.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield34.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield34.set("numColumns", new Integer(3));


    textfield35 = new TextFieldShadow();
    textfield35.set("name", "textfield35");
    gbpanel20.add(textfield35);
    textfield35.set("GBConstraints", new GBConstraints("x=3;y=6;fill=horizontal"));
    textfield35.set("text", "d23b");
    textfield35.set("background", convert("java.awt.Color", "springgreen"));
    textfield35.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield35.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield35.set("numColumns", new Integer(3));


    tfSCIData_064_065 = new TextFieldShadow();
    tfSCIData_064_065.set("name", "tfSCIData_064_065");
    gbpanel20.add(tfSCIData_064_065);
    tfSCIData_064_065.set("GBConstraints", new GBConstraints("x=3;y=7;fill=horizontal"));
    tfSCIData_064_065.set("text", "####");
    tfSCIData_064_065.set("background", convert("java.awt.Color", "green"));
    tfSCIData_064_065.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_064_065.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_064_065.set("numColumns", new Integer(3));


    tfSCIData_080_081 = new TextFieldShadow();
    tfSCIData_080_081.set("name", "tfSCIData_080_081");
    gbpanel20.add(tfSCIData_080_081);
    tfSCIData_080_081.set("GBConstraints", new GBConstraints("x=3;y=8;fill=horizontal"));
```

```
    tfSCIData_080_081.set("text", "####");

    tfSCIData_080_081.set("background", convert("java.awt.Color", "green"));

    tfSCIData_080_081.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_080_081.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_080_081.set("numColumns", new Integer(3));


    tfSCIData_096_097 = new TextFieldShadow();

    tfSCIData_096_097.set("name", "tfSCIData_096_097");

    gbpanel20.add(tfSCIData_096_097);

    tfSCIData_096_097.set("GBConstraints", new GBConstraints("x=3;y=9;fill=horizontal"));

    tfSCIData_096_097.set("text", "####");

    tfSCIData_096_097.set("background", convert("java.awt.Color", "green"));

    tfSCIData_096_097.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_096_097.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_096_097.set("numColumns", new Integer(3));


    tfSCIData_112_113 = new TextFieldShadow();

    tfSCIData_112_113.set("name", "tfSCIData_112_113");

    gbpanel20.add(tfSCIData_112_113);

    tfSCIData_112_113.set("GBConstraints", new GBConstraints("x=3;y=10;fill=horizontal"));

    tfSCIData_112_113.set("text", "####");

    tfSCIData_112_113.set("background", convert("java.awt.Color", "green"));

    tfSCIData_112_113.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_112_113.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_112_113.set("numColumns", new Integer(3));


    tfSCIData_128_129 = new TextFieldShadow();

    tfSCIData_128_129.set("name", "tfSCIData_128_129");

    gbpanel20.add(tfSCIData_128_129);

    tfSCIData_128_129.set("GBConstraints", new GBConstraints("x=3;y=11;fill=horizontal"));

    tfSCIData_128_129.set("text", "####");

    tfSCIData_128_129.set("background", convert("java.awt.Color", "green"));

    tfSCIData_128_129.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_128_129.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_128_129.set("numColumns", new Integer(3));


    tfSCIData_144_145 = new TextFieldShadow();

    tfSCIData_144_145.set("name", "tfSCIData_144_145");

    gbpanel20.add(tfSCIData_144_145);
```

```
    tfSCIData_144_145.set("GBConstraints", new GBConstraints("x=3;y=12;fill=horizontal"));

    tfSCIData_144_145.set("text", "####");

    tfSCIData_144_145.set("background", convert("java.awt.Color", "green"));

    tfSCIData_144_145.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_144_145.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_144_145.set("numColumns", new Integer(3));


    tfSCIData_160_161 = new TextFieldShadow();

    tfSCIData_160_161.set("name", "tfSCIData_160_161");

    gbpanel20.add(tfSCIData_160_161);

    tfSCIData_160_161.set("GBConstraints", new GBConstraints("x=3;y=13;fill=horizontal"));

    tfSCIData_160_161.set("text", "####");

    tfSCIData_160_161.set("background", convert("java.awt.Color", "green"));

    tfSCIData_160_161.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_160_161.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_160_161.set("numColumns", new Integer(3));


    tfSCIData_176_177 = new TextFieldShadow();

    tfSCIData_176_177.set("name", "tfSCIData_176_177");

    gbpanel20.add(tfSCIData_176_177);

    tfSCIData_176_177.set("GBConstraints", new GBConstraints("x=3;y=14;fill=horizontal"));

    tfSCIData_176_177.set("text", "####");

    tfSCIData_176_177.set("background", convert("java.awt.Color", "green"));

    tfSCIData_176_177.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_176_177.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_176_177.set("numColumns", new Integer(3));


    tfSCIData_192_193 = new TextFieldShadow();

    tfSCIData_192_193.set("name", "tfSCIData_192_193");

    gbpanel20.add(tfSCIData_192_193);

    tfSCIData_192_193.set("GBConstraints", new GBConstraints("x=3;y=15;fill=horizontal"));

    tfSCIData_192_193.set("text", "####");

    tfSCIData_192_193.set("background", convert("java.awt.Color", "green"));

    tfSCIData_192_193.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_192_193.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_192_193.set("numColumns", new Integer(3));


    tfSCIData_208_209 = new TextFieldShadow();

    tfSCIData_208_209.set("name", "tfSCIData_208_209");
```

```
    gbpanel20.add(tfSCIData_208_209);
    tfSCIData_208_209.set("GBConstraints", new GBConstraints("x=3;y=16;fill=horizontal"));
    tfSCIData_208_209.set("text", "####");
    tfSCIData_208_209.set("background", convert("java.awt.Color", "green"));
    tfSCIData_208_209.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_208_209.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_208_209.set("numColumns", new Integer(3));


    tfSCIData_224_225 = new TextFieldShadow();
    tfSCIData_224_225.set("name", "tfSCIData_224_225");
    gbpanel20.add(tfSCIData_224_225);
    tfSCIData_224_225.set("GBConstraints", new GBConstraints("x=3;y=17;fill=horizontal"));
    tfSCIData_224_225.set("text", "####");
    tfSCIData_224_225.set("background", convert("java.awt.Color", "green"));
    tfSCIData_224_225.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_224_225.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_224_225.set("numColumns", new Integer(3));


    tfSCIData_240_241 = new TextFieldShadow();
    tfSCIData_240_241.set("name", "tfSCIData_240_241");
    gbpanel20.add(tfSCIData_240_241);
    tfSCIData_240_241.set("GBConstraints", new GBConstraints("x=3;y=18;fill=horizontal"));
    tfSCIData_240_241.set("text", "####");
    tfSCIData_240_241.set("background", convert("java.awt.Color", "green"));
    tfSCIData_240_241.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_240_241.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_240_241.set("numColumns", new Integer(3));


    textfield49 = new TextFieldShadow();
    textfield49.set("name", "textfield49");
    gbpanel20.add(textfield49);
    textfield49.set("GBConstraints", new GBConstraints("x=4;y=5;fill=horizontal"));
    textfield49.set("text", "f451");
    textfield49.set("background", convert("java.awt.Color", "springgreen"));
    textfield49.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield49.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield49.set("numColumns", new Integer(3));


    textfield50 = new TextFieldShadow();
```

```
textfield50.set("name", "textfield50");
gbpanel20.add(textfield50);
textfield50.set("GBConstraints", new GBConstraints("x=5;y=5;fill=horizontal"));
textfield50.set("text", "5e1f");
textfield50.set("background", convert("java.awt.Color", "springgreen"));
textfield50.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield50.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield50.set("numColumns", new Integer(3));

textfield51 = new TextFieldShadow();
textfield51.set("name", "textfield51");
gbpanel20.add(textfield51);
textfield51.set("GBConstraints", new GBConstraints("x=6;y=5;fill=horizontal"));
textfield51.set("text", "4561");
textfield51.set("background", convert("java.awt.Color", "springgreen"));
textfield51.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield51.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield51.set("numColumns", new Integer(3));

textfield52 = new TextFieldShadow();
textfield52.set("name", "textfield52");
gbpanel20.add(textfield52);
textfield52.set("GBConstraints", new GBConstraints("x=7;y=5;fill=horizontal"));
textfield52.set("text", "e423");
textfield52.set("background", convert("java.awt.Color", "springgreen"));
textfield52.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield52.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield52.set("numColumns", new Integer(3));

textfield53 = new TextFieldShadow();
textfield53.set("name", "textfield53");
gbpanel20.add(textfield53);
textfield53.set("GBConstraints", new GBConstraints("x=8;y=5;fill=horizontal"));
textfield53.set("text", "5d1e");
textfield53.set("background", convert("java.awt.Color", "springgreen"));
textfield53.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield53.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield53.set("numColumns", new Integer(3));

textfield54 = new TextFieldShadow();
textfield54.set("name", "textfield54");
```

```
    gbpanel20.add(textfield54);
    textfield54.set("GBConstraints", new GBConstraints("x=9;y=5;fill=horizontal"));
    textfield54.set("text", "ff21");
    textfield54.set("background", convert("java.awt.Color", "springgreen"));
    textfield54.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield54.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield54.set("numColumns", new Integer(3));

    textfield55 = new TextFieldShadow();
    textfield55.set("name", "textfield55");
    gbpanel20.add(textfield55);
    textfield55.set("GBConstraints", new GBConstraints("x=10;y=5;fill=horizontal"));
    textfield55.set("text", "eac5");
    textfield55.set("background", convert("java.awt.Color", "springgreen"));
    textfield55.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield55.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield55.set("numColumns", new Integer(3));

    tfSCIData_082_083 = new TextFieldShadow();
    tfSCIData_082_083.set("name", "tfSCIData_082_083");
    gbpanel20.add(tfSCIData_082_083);
    tfSCIData_082_083.set("GBConstraints", new GBConstraints("x=4;y=8;fill=horizontal"));
    tfSCIData_082_083.set("text", "####");
    tfSCIData_082_083.set("background", convert("java.awt.Color", "green"));
    tfSCIData_082_083.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_082_083.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_082_083.set("numColumns", new Integer(3));

    tfSCIData_088_089 = new TextFieldShadow();
    tfSCIData_088_089.set("name", "tfSCIData_088_089");
    gbpanel20.add(tfSCIData_088_089);
    tfSCIData_088_089.set("GBConstraints", new GBConstraints("x=7;y=8;fill=horizontal"));
    tfSCIData_088_089.set("text", "####");
    tfSCIData_088_089.set("background", convert("java.awt.Color", "green"));
    tfSCIData_088_089.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_088_089.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_088_089.set("numColumns", new Integer(3));

    tfSCIData_098_099 = new TextFieldShadow();
    tfSCIData_098_099.set("name", "tfSCIData_098_099");
```

```
    gbpanel20.add(tfSCIData_098_099);

    tfSCIData_098_099.set("GBConstraints", new GBConstraints("x=4;y=9;fill=horizontal"));

    tfSCIData_098_099.set("text", "####");

    tfSCIData_098_099.set("background", convert("java.awt.Color", "green"));

    tfSCIData_098_099.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_098_099.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_098_099.set("numColumns", new Integer(3));


    tfSCIData_102_103 = new TextFieldShadow();

    tfSCIData_102_103.set("name", "tfSCIData_102_103");

    gbpanel20.add(tfSCIData_102_103);

    tfSCIData_102_103.set("GBConstraints", new GBConstraints("x=6;y=9;fill=horizontal"));

    tfSCIData_102_103.set("text", "####");

    tfSCIData_102_103.set("background", convert("java.awt.Color", "green"));

    tfSCIData_102_103.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_102_103.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_102_103.set("numColumns", new Integer(3));


    tfSCIData_104_105 = new TextFieldShadow();

    tfSCIData_104_105.set("name", "tfSCIData_104_105");

    gbpanel20.add(tfSCIData_104_105);

    tfSCIData_104_105.set("GBConstraints", new GBConstraints("x=7;y=9;fill=horizontal"));

    tfSCIData_104_105.set("text", "####");

    tfSCIData_104_105.set("background", convert("java.awt.Color", "green"));

    tfSCIData_104_105.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_104_105.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_104_105.set("numColumns", new Integer(3));


    tfSCIData_118_119 = new TextFieldShadow();

    tfSCIData_118_119.set("name", "tfSCIData_118_119");

    gbpanel20.add(tfSCIData_118_119);

    tfSCIData_118_119.set("GBConstraints", new GBConstraints("x=6;y=10;fill=horizontal"));

    tfSCIData_118_119.set("text", "####");

    tfSCIData_118_119.set("background", convert("java.awt.Color", "green"));

    tfSCIData_118_119.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_118_119.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_118_119.set("numColumns", new Integer(3));


    tfSCIData_122_123 = new TextFieldShadow();
```

```
    tfSCIData_122_123.set("name", "tfSCIData_122_123");
    gbpanel20.add(tfSCIData_122_123);
    tfSCIData_122_123.set("GBConstraints", new GBConstraints("x=8;y=10;fill=horizontal"));
    tfSCIData_122_123.set("text", "####");
    tfSCIData_122_123.set("background", convert("java.awt.Color", "green"));
    tfSCIData_122_123.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_122_123.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_122_123.set("numColumns", new Integer(3));


    tfSCIData_140_141 = new TextFieldShadow();
    tfSCIData_140_141.set("name", "tfSCIData_140_141");
    gbpanel20.add(tfSCIData_140_141);
    tfSCIData_140_141.set("GBConstraints", new GBConstraints("x=9;y=11;fill=horizontal"));
    tfSCIData_140_141.set("text", "####");
    tfSCIData_140_141.set("background", convert("java.awt.Color", "green"));
    tfSCIData_140_141.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_140_141.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_140_141.set("numColumns", new Integer(3));


    textfield64 = new TextFieldShadow();
    textfield64.set("name", "textfield64");
    gbpanel20.add(textfield64);
    textfield64.set("GBConstraints", new GBConstraints("x=4;y=6;fill=horizontal"));
    textfield64.set("text", "abd3");
    textfield64.set("background", convert("java.awt.Color", "springgreen"));
    textfield64.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield64.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield64.set("numColumns", new Integer(3));


    textfield65 = new TextFieldShadow();
    textfield65.set("name", "textfield65");
    gbpanel20.add(textfield65);
    textfield65.set("GBConstraints", new GBConstraints("x=5;y=6;fill=horizontal"));
    textfield65.set("text", "4ef1");
    textfield65.set("background", convert("java.awt.Color", "springgreen"));
    textfield65.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield65.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield65.set("numColumns", new Integer(3));


    textfield66 = new TextFieldShadow();
```

```
textfield66.set("name", "textfield66");
gbpanel20.add(textfield66);
textfield66.set("GBConstraints", new GBConstraints("x=6;y=6;fill=horizontal"));
textfield66.set("text", "ba3e");
textfield66.set("background", convert("java.awt.Color", "springgreen"));
textfield66.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield66.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield66.set("numColumns", new Integer(3));

textfield67 = new TextFieldShadow();
textfield67.set("name", "textfield67");
gbpanel20.add(textfield67);
textfield67.set("GBConstraints", new GBConstraints("x=7;y=6;fill=horizontal"));
textfield67.set("text", "924e");
textfield67.set("background", convert("java.awt.Color", "springgreen"));
textfield67.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield67.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield67.set("numColumns", new Integer(3));

textfield68 = new TextFieldShadow();
textfield68.set("name", "textfield68");
gbpanel20.add(textfield68);
textfield68.set("GBConstraints", new GBConstraints("x=8;y=6;fill=horizontal"));
textfield68.set("text", "31fe");
textfield68.set("background", convert("java.awt.Color", "springgreen"));
textfield68.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield68.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield68.set("numColumns", new Integer(3));

textfield69 = new TextFieldShadow();
textfield69.set("name", "textfield69");
gbpanel20.add(textfield69);
textfield69.set("GBConstraints", new GBConstraints("x=9;y=6;fill=horizontal"));
textfield69.set("text", "ffff");
textfield69.set("background", convert("java.awt.Color", "springgreen"));
textfield69.set("insets", new java.awt.Insets(0, 0, 0, 0));
textfield69.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
textfield69.set("numColumns", new Integer(3));

textfield70 = new TextFieldShadow();
textfield70.set("name", "textfield70");
```

```
    gbpanel20.add(textfield70);
    textfield70.set("GBConstraints", new GBConstraints("x=10;y=6;fill=horizontal"));
    textfield70.set("text", "f237");
    textfield70.set("background", convert("java.awt.Color", "springgreen"));
    textfield70.set("insets", new java.awt.Insets(0, 0, 0, 0));
    textfield70.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
    textfield70.set("numColumns", new Integer(3));

    tfSCIData_084_085 = new TextFieldShadow();
    tfSCIData_084_085.set("name", "tfSCIData_084_085");
    gbpanel20.add(tfSCIData_084_085);
    tfSCIData_084_085.set("GBConstraints", new GBConstraints("x=5;y=8;fill=horizontal"));
    tfSCIData_084_085.set("text", "####");
    tfSCIData_084_085.set("background", convert("java.awt.Color", "green"));
    tfSCIData_084_085.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_084_085.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_084_085.set("numColumns", new Integer(3));

    tfSCIData_092_093 = new TextFieldShadow();
    tfSCIData_092_093.set("name", "tfSCIData_092_093");
    gbpanel20.add(tfSCIData_092_093);
    tfSCIData_092_093.set("GBConstraints", new GBConstraints("x=9;y=8;fill=horizontal"));
    tfSCIData_092_093.set("text", "####");
    tfSCIData_092_093.set("background", convert("java.awt.Color", "green"));
    tfSCIData_092_093.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_092_093.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_092_093.set("numColumns", new Integer(3));

    tfSCIData_094_095 = new TextFieldShadow();
    tfSCIData_094_095.set("name", "tfSCIData_094_095");
    gbpanel20.add(tfSCIData_094_095);
    tfSCIData_094_095.set("GBConstraints", new GBConstraints("x=10;y=8;fill=horizontal"));
    tfSCIData_094_095.set("text", "####");
    tfSCIData_094_095.set("background", convert("java.awt.Color", "green"));
    tfSCIData_094_095.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_094_095.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_094_095.set("numColumns", new Integer(3));

    tfSCIData_108_109 = new TextFieldShadow();
```

```
    tfSCIData_108_109.set("name", "tfSCIData_108_109");

    gbpanel20.add(tfSCIData_108_109);

    tfSCIData_108_109.set("GBConstraints", new GBConstraints("x=9;y=9;fill=horizontal"));

    tfSCIData_108_109.set("text", "####");

    tfSCIData_108_109.set("background", convert("java.awt.Color", "green"));

    tfSCIData_108_109.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_108_109.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_108_109.set("numColumns", new Integer(3));


    tfSCIData_110_111 = new TextFieldShadow();

    tfSCIData_110_111.set("name", "tfSCIData_110_111");

    gbpanel20.add(tfSCIData_110_111);

    tfSCIData_110_111.set("GBConstraints", new GBConstraints("x=10;y=9;fill=horizontal"));

    tfSCIData_110_111.set("text", "####");

    tfSCIData_110_111.set("background", convert("java.awt.Color", "green"));

    tfSCIData_110_111.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_110_111.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_110_111.set("numColumns", new Integer(3));


    tfSCIData_114_115 = new TextFieldShadow();

    tfSCIData_114_115.set("name", "tfSCIData_114_115");

    gbpanel20.add(tfSCIData_114_115);

    tfSCIData_114_115.set("GBConstraints", new GBConstraints("x=4;y=10;fill=horizontal"));

    tfSCIData_114_115.set("text", "####");

    tfSCIData_114_115.set("background", convert("java.awt.Color", "green"));

    tfSCIData_114_115.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_114_115.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_114_115.set("numColumns", new Integer(3));


    tfSCIData_116_117 = new TextFieldShadow();

    tfSCIData_116_117.set("name", "tfSCIData_116_117");

    gbpanel20.add(tfSCIData_116_117);

    tfSCIData_116_117.set("GBConstraints", new GBConstraints("x=5;y=10;fill=horizontal"));

    tfSCIData_116_117.set("text", "####");

    tfSCIData_116_117.set("background", convert("java.awt.Color", "green"));

    tfSCIData_116_117.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_116_117.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_116_117.set("numColumns", new Integer(3));
```

```
    tfSCIData_142_143 = new TextFieldShadow();
    tfSCIData_142_143.set("name", "tfSCIData_142_143");
    gbpanel20.add(tfSCIData_142_143);
    tfSCIData_142_143.set("GBConstraints", new GBConstraints("x=10;y=11;fill=horizontal"));
    tfSCIData_142_143.set("text", "####");
    tfSCIData_142_143.set("background", convert("java.awt.Color", "green"));
    tfSCIData_142_143.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_142_143.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_142_143.set("numColumns", new Integer(3));


    label2 = new LabelShadow();
    label2.set("name", "label2");
    gbpanel20.add(label2);
    label2.set("GBConstraints", new GBConstraints("x=3;y=2"));
    label2.set("text", "00&01");
    label2.set("alignment", new sunsoft.jws.visual.rt.type.AlignmentEnum("left"));
    label2.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
    label2.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));


    label3 = new LabelShadow();
    label3.set("name", "label3");
    gbpanel20.add(label3);
    label3.set("GBConstraints", new GBConstraints("x=4;y=2"));
    label3.set("text", "02&03");
    label3.set("alignment", new sunsoft.jws.visual.rt.type.AlignmentEnum("left"));
    label3.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
    label3.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));


    label4 = new LabelShadow();
    label4.set("name", "label4");
    gbpanel20.add(label4);
    label4.set("GBConstraints", new GBConstraints("x=5;y=2"));
    label4.set("text", "04&05");
    label4.set("alignment", new sunsoft.jws.visual.rt.type.AlignmentEnum("left"));
    label4.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
    label4.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));


    label5 = new LabelShadow();
    label5.set("name", "label5");
    gbpanel20.add(label5);
```

```
label5.set("GBConstraints", new GBConstraints("x=6;y=2"));
label5.set("text", "06&07");
label5.set("alignment", new sunsoft.jws.visual.rt.type.AlignmentEnum("left"));
label5.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
label5.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label6 = new LabelShadow();
label6.set("name", "label6");
gbpanel20.add(label6);
label6.set("GBConstraints", new GBConstraints("x=7;y=2"));
label6.set("text", "08&10");
label6.set("alignment", new sunsoft.jws.visual.rt.type.AlignmentEnum("left"));
label6.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
label6.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label7 = new LabelShadow();
label7.set("name", "label7");
gbpanel20.add(label7);
label7.set("GBConstraints", new GBConstraints("x=8;y=2"));
label7.set("text", "10&11");
label7.set("alignment", new sunsoft.jws.visual.rt.type.AlignmentEnum("left"));
label7.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
label7.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label8 = new LabelShadow();
label8.set("name", "label8");
gbpanel20.add(label8);
label8.set("GBConstraints", new GBConstraints("x=9;y=2"));
label8.set("text", "12&13");
label8.set("alignment", new sunsoft.jws.visual.rt.type.AlignmentEnum("left"));
label8.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
label8.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label9 = new LabelShadow();
label9.set("name", "label9");
gbpanel20.add(label9);
label9.set("GBConstraints", new GBConstraints("x=10;y=2"));
label9.set("text", "14&15");
label9.set("alignment", new sunsoft.jws.visual.rt.type.AlignmentEnum("left"));
label9.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
label9.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));
```

```
label10 = new LabelShadow();
label10.set("name", "label10");
gbpanel20.add(label10);
label10.set("GBConstraints", new GBConstraints("x=2;y=3"));
label10.set("text", "000 to 015");
label10.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label11 = new LabelShadow();
label11.set("name", "label11");
gbpanel20.add(label11);
label11.set("GBConstraints", new GBConstraints("x=2;y=4"));
label11.set("text", "016 to 031");
label11.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label12 = new LabelShadow();
label12.set("name", "label12");
gbpanel20.add(label12);
label12.set("GBConstraints", new GBConstraints("x=2;y=5"));
label12.set("text", "032 to 047");
label12.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label13 = new LabelShadow();
label13.set("name", "label13");
gbpanel20.add(label13);
label13.set("GBConstraints", new GBConstraints("x=2;y=6"));
label13.set("text", "048 to 063");
label13.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label14 = new LabelShadow();
label14.set("name", "label14");
gbpanel20.add(label14);
label14.set("GBConstraints", new GBConstraints("x=2;y=7"));
label14.set("text", "064 to 079");
label14.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label15 = new LabelShadow();
label15.set("name", "label15");
gbpanel20.add(label15);
label15.set("GBConstraints", new GBConstraints("x=2;y=8"));
label15.set("text", "080 to 095");
```

```
    label15.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

    label16 = new LabelShadow();
    label16.set("name", "label16");
    gbpanel20.add(label16);
    label16.set("GBConstraints", new GBConstraints("x=2;y=9"));
    label16.set("text", "096 to 111");
    label16.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

    label17 = new LabelShadow();
    label17.set("name", "label17");
    gbpanel20.add(label17);
    label17.set("GBConstraints", new GBConstraints("x=2;y=10"));
    label17.set("text", "112 to127");
    label17.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

    label18 = new LabelShadow();
    label18.set("name", "label18");
    gbpanel20.add(label18);
    label18.set("GBConstraints", new GBConstraints("x=2;y=2"));
    label18.set("text", "byte");
    label18.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

    tfSCIData_130_131 = new TextFieldShadow();
    tfSCIData_130_131.set("name", "tfSCIData_130_131");
    gbpanel20.add(tfSCIData_130_131);
    tfSCIData_130_131.set("GBConstraints", new GBConstraints("x=4;y=11;fill=horizontal"));
    tfSCIData_130_131.set("text", "####");
    tfSCIData_130_131.set("background", convert("java.awt.Color", "green"));
    tfSCIData_130_131.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_130_131.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_130_131.set("numColumns", new Integer(3));

    tfSCIData_132_133 = new TextFieldShadow();
    tfSCIData_132_133.set("name", "tfSCIData_132_133");
    gbpanel20.add(tfSCIData_132_133);
    tfSCIData_132_133.set("GBConstraints", new GBConstraints("x=5;y=11;fill=horizontal"));
    tfSCIData_132_133.set("text", "####");
    tfSCIData_132_133.set("background", convert("java.awt.Color", "green"));
    tfSCIData_132_133.set("insets", new java.awt.Insets(0, 0, 0, 0));
```

```
    tfSCIData_132_133.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_132_133.set("numColumns", new Integer(3));


    tfSCIData_134_135 = new TextFieldShadow();

    tfSCIData_134_135.set("name", "tfSCIData_134_135");

    gbpanel20.add(tfSCIData_134_135);

    tfSCIData_134_135.set("GBConstraints", new GBConstraints("x=6;y=11;fill=horizontal"));

    tfSCIData_134_135.set("text", "####");

    tfSCIData_134_135.set("background", convert("java.awt.Color", "green"));

    tfSCIData_134_135.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_134_135.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_134_135.set("numColumns", new Integer(3));


    tfSCIData_146_147 = new TextFieldShadow();

    tfSCIData_146_147.set("name", "tfSCIData_146_147");

    gbpanel20.add(tfSCIData_146_147);

    tfSCIData_146_147.set("GBConstraints", new GBConstraints("x=4;y=12;fill=horizontal"));

    tfSCIData_146_147.set("text", "####");

    tfSCIData_146_147.set("background", convert("java.awt.Color", "green"));

    tfSCIData_146_147.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_146_147.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_146_147.set("numColumns", new Integer(3));


    tfSCIData_148_149 = new TextFieldShadow();

    tfSCIData_148_149.set("name", "tfSCIData_148_149");

    gbpanel20.add(tfSCIData_148_149);

    tfSCIData_148_149.set("GBConstraints", new GBConstraints("x=5;y=12;fill=horizontal"));

    tfSCIData_148_149.set("text", "####");

    tfSCIData_148_149.set("background", convert("java.awt.Color", "green"));

    tfSCIData_148_149.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_148_149.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_148_149.set("numColumns", new Integer(3));


    tfSCIData_150_151 = new TextFieldShadow();

    tfSCIData_150_151.set("name", "tfSCIData_150_151");

    gbpanel20.add(tfSCIData_150_151);

    tfSCIData_150_151.set("GBConstraints", new GBConstraints("x=6;y=12;fill=horizontal"));

    tfSCIData_150_151.set("text", "####");

    tfSCIData_150_151.set("background", convert("java.awt.Color", "green"));
```

```
    tfSCIData_150_151.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_150_151.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_150_151.set("numColumns", new Integer(3));


    tfSCIData_152_153 = new TextFieldShadow();

    tfSCIData_152_153.set("name", "tfSCIData_152_153");

    gbpanel20.add(tfSCIData_152_153);

    tfSCIData_152_153.set("GBConstraints", new GBConstraints("x=7;y=12;fill=horizontal"));

    tfSCIData_152_153.set("text", "####");

    tfSCIData_152_153.set("background", convert("java.awt.Color", "green"));

    tfSCIData_152_153.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_152_153.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_152_153.set("numColumns", new Integer(3));


    tfSCIData_154_155 = new TextFieldShadow();

    tfSCIData_154_155.set("name", "tfSCIData_154_155");

    gbpanel20.add(tfSCIData_154_155);

    tfSCIData_154_155.set("GBConstraints", new GBConstraints("x=8;y=12;fill=horizontal"));

    tfSCIData_154_155.set("text", "####");

    tfSCIData_154_155.set("background", convert("java.awt.Color", "green"));

    tfSCIData_154_155.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_154_155.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_154_155.set("numColumns", new Integer(3));


    tfSCIData_156_157 = new TextFieldShadow();

    tfSCIData_156_157.set("name", "tfSCIData_156_157");

    gbpanel20.add(tfSCIData_156_157);

    tfSCIData_156_157.set("GBConstraints", new GBConstraints("x=9;y=12;fill=horizontal"));

    tfSCIData_156_157.set("text", "####");

    tfSCIData_156_157.set("background", convert("java.awt.Color", "green"));

    tfSCIData_156_157.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_156_157.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_156_157.set("numColumns", new Integer(3));


    tfSCIData_158_159 = new TextFieldShadow();

    tfSCIData_158_159.set("name", "tfSCIData_158_159");

    gbpanel20.add(tfSCIData_158_159);

    tfSCIData_158_159.set("GBConstraints", new GBConstraints("x=10;y=12;fill=horizontal"));

    tfSCIData_158_159.set("text", "####");
```

```
    tfSCIData_158_159.set("background", convert("java.awt.Color", "green"));

    tfSCIData_158_159.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_158_159.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_158_159.set("numColumns", new Integer(3));


    tfSCIData_162_163 = new TextFieldShadow();

    tfSCIData_162_163.set("name", "tfSCIData_162_163");

    gbpanel20.add(tfSCIData_162_163);

    tfSCIData_162_163.set("GBConstraints", new GBConstraints("x=4;y=13;fill=horizontal"));

    tfSCIData_162_163.set("text", "####");

    tfSCIData_162_163.set("background", convert("java.awt.Color", "green"));

    tfSCIData_162_163.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_162_163.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_162_163.set("numColumns", new Integer(3));


    tfSCIData_164_165 = new TextFieldShadow();

    tfSCIData_164_165.set("name", "tfSCIData_164_165");

    gbpanel20.add(tfSCIData_164_165);

    tfSCIData_164_165.set("GBConstraints", new GBConstraints("x=5;y=13;fill=horizontal"));

    tfSCIData_164_165.set("text", "####");

    tfSCIData_164_165.set("background", convert("java.awt.Color", "green"));

    tfSCIData_164_165.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_164_165.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_164_165.set("numColumns", new Integer(3));


    tfSCIData_166_167 = new TextFieldShadow();

    tfSCIData_166_167.set("name", "tfSCIData_166_167");

    gbpanel20.add(tfSCIData_166_167);

    tfSCIData_166_167.set("GBConstraints", new GBConstraints("x=6;y=13;fill=horizontal"));

    tfSCIData_166_167.set("text", "####");

    tfSCIData_166_167.set("background", convert("java.awt.Color", "green"));

    tfSCIData_166_167.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_166_167.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_166_167.set("numColumns", new Integer(3));


    tfSCIData_168_169 = new TextFieldShadow();

    tfSCIData_168_169.set("name", "tfSCIData_168_169");

    gbpanel20.add(tfSCIData_168_169);

    tfSCIData_168_169.set("GBConstraints", new GBConstraints("x=7;y=13;fill=horizontal"));
```

```
    tfSCIData_168_169.set("text", "####");

    tfSCIData_168_169.set("background", convert("java.awt.Color", "green"));

    tfSCIData_168_169.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_168_169.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_168_169.set("numColumns", new Integer(3));


    tfSCIData_170_171 = new TextFieldShadow();

    tfSCIData_170_171.set("name", "tfSCIData_170_171");

    gbpanel20.add(tfSCIData_170_171);

    tfSCIData_170_171.set("GBConstraints", new GBConstraints("x=8;y=13;fill=horizontal"));

    tfSCIData_170_171.set("text", "####");

    tfSCIData_170_171.set("background", convert("java.awt.Color", "green"));

    tfSCIData_170_171.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_170_171.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_170_171.set("numColumns", new Integer(3));


    tfSCIData_172_173 = new TextFieldShadow();

    tfSCIData_172_173.set("name", "tfSCIData_172_173");

    gbpanel20.add(tfSCIData_172_173);

    tfSCIData_172_173.set("GBConstraints", new GBConstraints("x=9;y=13;fill=horizontal"));

    tfSCIData_172_173.set("text", "####");

    tfSCIData_172_173.set("background", convert("java.awt.Color", "green"));

    tfSCIData_172_173.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_172_173.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_172_173.set("numColumns", new Integer(3));


    tfSCIData_174_175 = new TextFieldShadow();

    tfSCIData_174_175.set("name", "tfSCIData_174_175");

    gbpanel20.add(tfSCIData_174_175);

    tfSCIData_174_175.set("GBConstraints", new GBConstraints("x=10;y=13;fill=horizontal"));

    tfSCIData_174_175.set("text", "####");

    tfSCIData_174_175.set("background", convert("java.awt.Color", "green"));

    tfSCIData_174_175.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_174_175.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_174_175.set("numColumns", new Integer(3));


    tfSCIData_178_179 = new TextFieldShadow();

    tfSCIData_178_179.set("name", "tfSCIData_178_179");

    gbpanel20.add(tfSCIData_178_179);
```

```
    tfSCIData_178_179.set("GBConstraints", new GBConstraints("x=4;y=14;fill=horizontal"));

    tfSCIData_178_179.set("text", "####");

    tfSCIData_178_179.set("background", convert("java.awt.Color", "green"));

    tfSCIData_178_179.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_178_179.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_178_179.set("numColumns", new Integer(3));


    tfSCIData_180_181 = new TextFieldShadow();

    tfSCIData_180_181.set("name", "tfSCIData_180_181");

    gbpanel20.add(tfSCIData_180_181);

    tfSCIData_180_181.set("GBConstraints", new GBConstraints("x=5;y=14;fill=horizontal"));

    tfSCIData_180_181.set("text", "####");

    tfSCIData_180_181.set("background", convert("java.awt.Color", "green"));

    tfSCIData_180_181.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_180_181.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_180_181.set("numColumns", new Integer(3));


    tfSCIData_182_183 = new TextFieldShadow();

    tfSCIData_182_183.set("name", "tfSCIData_182_183");

    gbpanel20.add(tfSCIData_182_183);

    tfSCIData_182_183.set("GBConstraints", new GBConstraints("x=6;y=14;fill=horizontal"));

    tfSCIData_182_183.set("text", "####");

    tfSCIData_182_183.set("background", convert("java.awt.Color", "green"));

    tfSCIData_182_183.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_182_183.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_182_183.set("numColumns", new Integer(3));


    tfSCIData_184_185 = new TextFieldShadow();

    tfSCIData_184_185.set("name", "tfSCIData_184_185");

    gbpanel20.add(tfSCIData_184_185);

    tfSCIData_184_185.set("GBConstraints", new GBConstraints("x=7;y=14;fill=horizontal"));

    tfSCIData_184_185.set("text", "####");

    tfSCIData_184_185.set("background", convert("java.awt.Color", "green"));

    tfSCIData_184_185.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_184_185.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_184_185.set("numColumns", new Integer(3));


    tfSCIData_186_187 = new TextFieldShadow();

    tfSCIData_186_187.set("name", "tfSCIData_186_187");
```

```
    gbpanel20.add(tfSCIData_186_187);

    tfSCIData_186_187.set("GBConstraints", new GBConstraints("x=8;y=14;fill=horizontal"));

    tfSCIData_186_187.set("text", "####");

    tfSCIData_186_187.set("background", convert("java.awt.Color", "green"));

    tfSCIData_186_187.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_186_187.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_186_187.set("numColumns", new Integer(3));


    tfSCIData_188_189 = new TextFieldShadow();

    tfSCIData_188_189.set("name", "tfSCIData_188_189");

    gbpanel20.add(tfSCIData_188_189);

    tfSCIData_188_189.set("GBConstraints", new GBConstraints("x=9;y=14;fill=horizontal"));

    tfSCIData_188_189.set("text", "####");

    tfSCIData_188_189.set("background", convert("java.awt.Color", "green"));

    tfSCIData_188_189.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_188_189.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_188_189.set("numColumns", new Integer(3));


    tfSCIData_190_191 = new TextFieldShadow();

    tfSCIData_190_191.set("name", "tfSCIData_190_191");

    gbpanel20.add(tfSCIData_190_191);

    tfSCIData_190_191.set("GBConstraints", new GBConstraints("x=10;y=14;fill=horizontal"));

    tfSCIData_190_191.set("text", "####");

    tfSCIData_190_191.set("background", convert("java.awt.Color", "green"));

    tfSCIData_190_191.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_190_191.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_190_191.set("numColumns", new Integer(3));


    tfSCIData_194_195 = new TextFieldShadow();

    tfSCIData_194_195.set("name", "tfSCIData_194_195");

    gbpanel20.add(tfSCIData_194_195);

    tfSCIData_194_195.set("GBConstraints", new GBConstraints("x=4;y=15;fill=horizontal"));

    tfSCIData_194_195.set("text", "####");

    tfSCIData_194_195.set("background", convert("java.awt.Color", "green"));

    tfSCIData_194_195.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_194_195.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_194_195.set("numColumns", new Integer(3));


    tfSCIData_196_197 = new TextFieldShadow();
```

```
    tfSCIData_196_197.set("name", "tfSCIData_196_197");

    gbpanel20.add(tfSCIData_196_197);

    tfSCIData_196_197.set("GBConstraints", new GBConstraints("x=5;y=15;fill=horizontal"));

    tfSCIData_196_197.set("text", "####");

    tfSCIData_196_197.set("background", convert("java.awt.Color", "green"));

    tfSCIData_196_197.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_196_197.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_196_197.set("numColumns", new Integer(3));


    tfSCIData_198_199 = new TextFieldShadow();

    tfSCIData_198_199.set("name", "tfSCIData_198_199");

    gbpanel20.add(tfSCIData_198_199);

    tfSCIData_198_199.set("GBConstraints", new GBConstraints("x=6;y=15;fill=horizontal"));

    tfSCIData_198_199.set("text", "####");

    tfSCIData_198_199.set("background", convert("java.awt.Color", "green"));

    tfSCIData_198_199.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_198_199.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_198_199.set("numColumns", new Integer(3));


    tfSCIData_200_201 = new TextFieldShadow();

    tfSCIData_200_201.set("name", "tfSCIData_200_201");

    gbpanel20.add(tfSCIData_200_201);

    tfSCIData_200_201.set("GBConstraints", new GBConstraints("x=7;y=15;fill=horizontal"));

    tfSCIData_200_201.set("text", "####");

    tfSCIData_200_201.set("background", convert("java.awt.Color", "green"));

    tfSCIData_200_201.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_200_201.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_200_201.set("numColumns", new Integer(3));


    tfSCIData_202_203 = new TextFieldShadow();

    tfSCIData_202_203.set("name", "tfSCIData_202_203");

    gbpanel20.add(tfSCIData_202_203);

    tfSCIData_202_203.set("GBConstraints", new GBConstraints("x=8;y=15;fill=horizontal"));

    tfSCIData_202_203.set("text", "####");

    tfSCIData_202_203.set("background", convert("java.awt.Color", "green"));

    tfSCIData_202_203.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_202_203.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_202_203.set("numColumns", new Integer(3));
```

```
    tfSCIData_204_205 = new TextFieldShadow();

    tfSCIData_204_205.set("name", "tfSCIData_204_205");

    gbpanel20.add(tfSCIData_204_205);

    tfSCIData_204_205.set("GBConstraints", new GBConstraints("x=9;y=15;fill=horizontal"));

    tfSCIData_204_205.set("text", "####");

    tfSCIData_204_205.set("background", convert("java.awt.Color", "green"));

    tfSCIData_204_205.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_204_205.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_204_205.set("numColumns", new Integer(3));


    tfSCIData_206_207 = new TextFieldShadow();

    tfSCIData_206_207.set("name", "tfSCIData_206_207");

    gbpanel20.add(tfSCIData_206_207);

    tfSCIData_206_207.set("GBConstraints", new GBConstraints("x=10;y=15;fill=horizontal"));

    tfSCIData_206_207.set("text", "####");

    tfSCIData_206_207.set("background", convert("java.awt.Color", "green"));

    tfSCIData_206_207.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_206_207.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_206_207.set("numColumns", new Integer(3));


    tfSCIData_210_211 = new TextFieldShadow();

    tfSCIData_210_211.set("name", "tfSCIData_210_211");

    gbpanel20.add(tfSCIData_210_211);

    tfSCIData_210_211.set("GBConstraints", new GBConstraints("x=4;y=16;fill=horizontal"));

    tfSCIData_210_211.set("text", "####");

    tfSCIData_210_211.set("background", convert("java.awt.Color", "green"));

    tfSCIData_210_211.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_210_211.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_210_211.set("numColumns", new Integer(3));


    tfSCIData_212_213 = new TextFieldShadow();

    tfSCIData_212_213.set("name", "tfSCIData_212_213");

    gbpanel20.add(tfSCIData_212_213);

    tfSCIData_212_213.set("GBConstraints", new GBConstraints("x=5;y=16;fill=horizontal"));

    tfSCIData_212_213.set("text", "####");

    tfSCIData_212_213.set("background", convert("java.awt.Color", "green"));

    tfSCIData_212_213.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_212_213.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_212_213.set("numColumns", new Integer(3));
```

```
    tfSCIData_214_215 = new TextFieldShadow();

    tfSCIData_214_215.set("name", "tfSCIData_214_215");

    gbpanel20.add(tfSCIData_214_215);

    tfSCIData_214_215.set("GBConstraints", new GBConstraints("x=6;y=16;fill=horizontal"));

    tfSCIData_214_215.set("text", "####");

    tfSCIData_214_215.set("background", convert("java.awt.Color", "green"));

    tfSCIData_214_215.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_214_215.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_214_215.set("numColumns", new Integer(3));


    tfSCIData_216_217 = new TextFieldShadow();

    tfSCIData_216_217.set("name", "tfSCIData_216_217");

    gbpanel20.add(tfSCIData_216_217);

    tfSCIData_216_217.set("GBConstraints", new GBConstraints("x=7;y=16;fill=horizontal"));

    tfSCIData_216_217.set("text", "####");

    tfSCIData_216_217.set("background", convert("java.awt.Color", "green"));

    tfSCIData_216_217.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_216_217.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_216_217.set("numColumns", new Integer(3));


    tfSCIData_218_219 = new TextFieldShadow();

    tfSCIData_218_219.set("name", "tfSCIData_218_219");

    gbpanel20.add(tfSCIData_218_219);

    tfSCIData_218_219.set("GBConstraints", new GBConstraints("x=8;y=16;fill=horizontal"));

    tfSCIData_218_219.set("text", "####");

    tfSCIData_218_219.set("background", convert("java.awt.Color", "green"));

    tfSCIData_218_219.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_218_219.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_218_219.set("numColumns", new Integer(3));


    tfSCIData_220_221 = new TextFieldShadow();

    tfSCIData_220_221.set("name", "tfSCIData_220_221");

    gbpanel20.add(tfSCIData_220_221);

    tfSCIData_220_221.set("GBConstraints", new GBConstraints("x=9;y=16;fill=horizontal"));

    tfSCIData_220_221.set("text", "####");

    tfSCIData_220_221.set("background", convert("java.awt.Color", "green"));

    tfSCIData_220_221.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_220_221.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
```

```
    tfSCIData_220_221.set("numColumns", new Integer(3));


    tfSCIData_222_223 = new TextFieldShadow();
    tfSCIData_222_223.set("name", "tfSCIData_222_223");
    gbpanel20.add(tfSCIData_222_223);
    tfSCIData_222_223.set("GBConstraints", new GBConstraints("x=10;y=16;fill=horizontal"));
    tfSCIData_222_223.set("text", "####");
    tfSCIData_222_223.set("background", convert("java.awt.Color", "green"));
    tfSCIData_222_223.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_222_223.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_222_223.set("numColumns", new Integer(3));


    tfSCIData_226_227 = new TextFieldShadow();
    tfSCIData_226_227.set("name", "tfSCIData_226_227");
    gbpanel20.add(tfSCIData_226_227);
    tfSCIData_226_227.set("GBConstraints", new GBConstraints("x=4;y=17;fill=horizontal"));
    tfSCIData_226_227.set("text", "####");
    tfSCIData_226_227.set("background", convert("java.awt.Color", "green"));
    tfSCIData_226_227.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_226_227.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_226_227.set("numColumns", new Integer(3));


    tfSCIData_228_229 = new TextFieldShadow();
    tfSCIData_228_229.set("name", "tfSCIData_228_229");
    gbpanel20.add(tfSCIData_228_229);
    tfSCIData_228_229.set("GBConstraints", new GBConstraints("x=5;y=17;fill=horizontal"));
    tfSCIData_228_229.set("text", "####");
    tfSCIData_228_229.set("background", convert("java.awt.Color", "green"));
    tfSCIData_228_229.set("insets", new java.awt.Insets(0, 0, 0, 0));
    tfSCIData_228_229.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));
    tfSCIData_228_229.set("numColumns", new Integer(3));


    tfSCIData_230_231 = new TextFieldShadow();
    tfSCIData_230_231.set("name", "tfSCIData_230_231");
    gbpanel20.add(tfSCIData_230_231);
    tfSCIData_230_231.set("GBConstraints", new GBConstraints("x=6;y=17;fill=horizontal"));
    tfSCIData_230_231.set("text", "####");
    tfSCIData_230_231.set("background", convert("java.awt.Color", "green"));
    tfSCIData_230_231.set("insets", new java.awt.Insets(0, 0, 0, 0));
```

```
    tfSCIData_230_231.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_230_231.set("numColumns", new Integer(3));


    tfSCIData_232_233 = new TextFieldShadow();

    tfSCIData_232_233.set("name", "tfSCIData_232_233");

    gbpanel20.add(tfSCIData_232_233);

    tfSCIData_232_233.set("GBConstraints", new GBConstraints("x=7;y=17;fill=horizontal"));

    tfSCIData_232_233.set("text", "####");

    tfSCIData_232_233.set("background", convert("java.awt.Color", "green"));

    tfSCIData_232_233.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_232_233.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_232_233.set("numColumns", new Integer(3));


    tfSCIData_234_235 = new TextFieldShadow();

    tfSCIData_234_235.set("name", "tfSCIData_234_235");

    gbpanel20.add(tfSCIData_234_235);

    tfSCIData_234_235.set("GBConstraints", new GBConstraints("x=8;y=17;fill=horizontal"));

    tfSCIData_234_235.set("text", "####");

    tfSCIData_234_235.set("background", convert("java.awt.Color", "green"));

    tfSCIData_234_235.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_234_235.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_234_235.set("numColumns", new Integer(3));


    tfSCIData_236_237 = new TextFieldShadow();

    tfSCIData_236_237.set("name", "tfSCIData_236_237");

    gbpanel20.add(tfSCIData_236_237);

    tfSCIData_236_237.set("GBConstraints", new GBConstraints("x=9;y=17;fill=horizontal"));

    tfSCIData_236_237.set("text", "####");

    tfSCIData_236_237.set("background", convert("java.awt.Color", "green"));

    tfSCIData_236_237.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_236_237.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_236_237.set("numColumns", new Integer(3));


    tfSCIData_238_239 = new TextFieldShadow();

    tfSCIData_238_239.set("name", "tfSCIData_238_239");

    gbpanel20.add(tfSCIData_238_239);

    tfSCIData_238_239.set("GBConstraints", new GBConstraints("x=10;y=17;fill=horizontal"));

    tfSCIData_238_239.set("text", "####");

    tfSCIData_238_239.set("background", convert("java.awt.Color", "green"));
```

```
    tfSCIData_238_239.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_238_239.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_238_239.set("numColumns", new Integer(3));


    tfSCIData_242_243 = new TextFieldShadow();

    tfSCIData_242_243.set("name", "tfSCIData_242_243");

    gbpanel20.add(tfSCIData_242_243);

    tfSCIData_242_243.set("GBConstraints", new GBConstraints("x=4;y=18;fill=horizontal"));

    tfSCIData_242_243.set("text", "####");

    tfSCIData_242_243.set("background", convert("java.awt.Color", "green"));

    tfSCIData_242_243.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_242_243.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_242_243.set("numColumns", new Integer(3));


    tfSCIData_244_245 = new TextFieldShadow();

    tfSCIData_244_245.set("name", "tfSCIData_244_245");

    gbpanel20.add(tfSCIData_244_245);

    tfSCIData_244_245.set("GBConstraints", new GBConstraints("x=5;y=18;fill=horizontal"));

    tfSCIData_244_245.set("text", "####");

    tfSCIData_244_245.set("background", convert("java.awt.Color", "green"));

    tfSCIData_244_245.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_244_245.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_244_245.set("numColumns", new Integer(3));


    tfSCIData_246_247 = new TextFieldShadow();

    tfSCIData_246_247.set("name", "tfSCIData_246_247");

    gbpanel20.add(tfSCIData_246_247);

    tfSCIData_246_247.set("GBConstraints", new GBConstraints("x=6;y=18;fill=horizontal"));

    tfSCIData_246_247.set("text", "####");

    tfSCIData_246_247.set("background", convert("java.awt.Color", "green"));

    tfSCIData_246_247.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_246_247.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_246_247.set("numColumns", new Integer(3));


    tfSCIData_248_249 = new TextFieldShadow();

    tfSCIData_248_249.set("name", "tfSCIData_248_249");

    gbpanel20.add(tfSCIData_248_249);

    tfSCIData_248_249.set("GBConstraints", new GBConstraints("x=7;y=18;fill=horizontal"));

    tfSCIData_248_249.set("text", "####");
```

```
    tfSCIData_248_249.set("background", convert("java.awt.Color", "green"));

    tfSCIData_248_249.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_248_249.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_248_249.set("numColumns", new Integer(3));


    tfSCIData_250_251 = new TextFieldShadow();

    tfSCIData_250_251.set("name", "tfSCIData_250_251");

    gbpanel20.add(tfSCIData_250_251);

    tfSCIData_250_251.set("GBConstraints", new GBConstraints("x=8;y=18;fill=horizontal"));

    tfSCIData_250_251.set("text", "####");

    tfSCIData_250_251.set("background", convert("java.awt.Color", "green"));

    tfSCIData_250_251.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_250_251.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_250_251.set("numColumns", new Integer(3));


    tfSCIData_252_253 = new TextFieldShadow();

    tfSCIData_252_253.set("name", "tfSCIData_252_253");

    gbpanel20.add(tfSCIData_252_253);

    tfSCIData_252_253.set("GBConstraints", new GBConstraints("x=9;y=18;fill=horizontal"));

    tfSCIData_252_253.set("text", "####");

    tfSCIData_252_253.set("background", convert("java.awt.Color", "green"));

    tfSCIData_252_253.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_252_253.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_252_253.set("numColumns", new Integer(3));


    tfSCIData_254_255 = new TextFieldShadow();

    tfSCIData_254_255.set("name", "tfSCIData_254_255");

    gbpanel20.add(tfSCIData_254_255);

    tfSCIData_254_255.set("GBConstraints", new GBConstraints("x=10;y=18;fill=horizontal"));

    tfSCIData_254_255.set("text", "####");

    tfSCIData_254_255.set("background", convert("java.awt.Color", "green"));

    tfSCIData_254_255.set("insets", new java.awt.Insets(0, 0, 0, 0));

    tfSCIData_254_255.set("font", convert("java.awt.Font",
"name=Dialog;style=plain;size=10"));

    tfSCIData_254_255.set("numColumns", new Integer(3));


    labelbar1 = new LabelBarShadow();

    labelbar1.set("name", "labelbar1");

    gbpanel20.add(labelbar1);

    labelbar1.set("GBConstraints", new GBConstraints("x=0;y=1;width=11;fill=horizontal"));
```

```
label28 = new LabelShadow();
label28.set("name", "label28");
gbpanel20.add(label28);
label28.set("GBConstraints", new GBConstraints("x=2;y=11"));
label28.set("text", "128 to 143");
label28.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label31 = new LabelShadow();
label31.set("name", "label31");
gbpanel20.add(label31);
label31.set("GBConstraints", new GBConstraints("x=2;y=12"));
label31.set("text", "144 to 159");
label31.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label32 = new LabelShadow();
label32.set("name", "label32");
gbpanel20.add(label32);
label32.set("GBConstraints", new GBConstraints("x=2;y=13"));
label32.set("text", "160 to 175");
label32.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label33 = new LabelShadow();
label33.set("name", "label33");
gbpanel20.add(label33);
label33.set("GBConstraints", new GBConstraints("x=2;y=14"));
label33.set("text", "176 to 191");
label33.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label34 = new LabelShadow();
label34.set("name", "label34");
gbpanel20.add(label34);
label34.set("GBConstraints", new GBConstraints("x=2;y=15"));
label34.set("text", "192 to 207");
label34.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label35 = new LabelShadow();
label35.set("name", "label35");
gbpanel20.add(label35);
label35.set("GBConstraints", new GBConstraints("x=2;y=16"));
label35.set("text", "208 to 223");
```

```
label35.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label36 = new LabelShadow();
label36.set("name", "label36");
gbpanel20.add(label36);
label36.set("GBConstraints", new GBConstraints("x=2;y=17"));
label36.set("text", "224 to 239");
label36.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

label37 = new LabelShadow();
label37.set("name", "label37");
gbpanel20.add(label37);
label37.set("GBConstraints", new GBConstraints("x=2;y=18"));
label37.set("text", "240 to 255");
label37.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=10"));

gbpanel19 = new GBPanelShadow();
gbpanel19.set("name", "gbpanel19");
gbpanel20.add(gbpanel19);
{
  int _tmp[] = {14,14,14,14,14,14,14};
  gbpanel19.set("rowHeights", _tmp);
}
gbpanel19.set("GBConstraints", new GBConstraints("x=0;y=0;width=11;fill=both"));
{
  int _tmp[] = {14,14,14,14,14,14,14,14};
  gbpanel19.set("columnWidths", _tmp);
}
gbpanel19.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
{
  double _tmp[] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0};
  gbpanel19.set("rowWeights", _tmp);
}
{
  double _tmp[] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
  gbpanel19.set("columnWeights", _tmp);
}

textfield131 = new TextFieldShadow();
textfield131.set("name", "textfield131");
gbpanel19.add(textfield131);
```

```
textfield131.set("GBConstraints", new GBConstraints("x=1;y=0;fill=horizontal"));
textfield131.set("text", "f34e");
textfield131.set("numColumns", new Integer(4));

textfield132 = new TextFieldShadow();
textfield132.set("name", "textfield132");
gbpanel19.add(textfield132);
textfield132.set("GBConstraints", new GBConstraints("x=3;y=0;fill=horizontal"));
textfield132.set("text", "e617");
textfield132.set("numColumns", new Integer(4));

textfield133 = new TextFieldShadow();
textfield133.set("name", "textfield133");
gbpanel19.add(textfield133);
textfield133.set("GBConstraints", new GBConstraints("x=5;y=0;fill=horizontal"));
textfield133.set("text", "673f");
textfield133.set("numColumns", new Integer(4));

textfield134 = new TextFieldShadow();
textfield134.set("name", "textfield134");
gbpanel19.add(textfield134);
textfield134.set("GBConstraints", new GBConstraints("x=7;y=0;fill=horizontal"));
textfield134.set("text", "6e3h");
textfield134.set("numColumns", new Integer(4));

label19 = new LabelShadow();
label19.set("name", "label19");
gbpanel19.add(label19);
label19.set("GBConstraints", new GBConstraints("x=0;y=0"));
label19.set("text", "code");

label20 = new LabelShadow();
label20.set("name", "label20");
gbpanel19.add(label20);
label20.set("GBConstraints", new GBConstraints("x=2;y=0"));
label20.set("text", "target Id");

label21 = new LabelShadow();
label21.set("name", "label21");
gbpanel19.add(label21);
label21.set("GBConstraints", new GBConstraints("x=4;y=0"));
```

```
label21.set("text", "command");

label22 = new LabelShadow();
label22.set("name", "label22");
gbpanel19.add(label22);
label22.set("GBConstraints", new GBConstraints("x=6;y=0"));
label22.set("text", "source Id");

labelbar2 = new LabelBarShadow();
labelbar2.set("name", "labelbar2");
gbpanel19.add(labelbar2);
labelbar2.set("GBConstraints", new GBConstraints("x=0;y=1;width=8;fill=horizontal"));

labelbar3 = new LabelBarShadow();
labelbar3.set("name", "labelbar3");
gbpanel19.add(labelbar3);
labelbar3.set("GBConstraints", new GBConstraints("x=0;y=3;width=8;fill=horizontal"));

label23 = new LabelShadow();
label23.set("name", "label23");
gbpanel19.add(label23);
label23.set("GBConstraints", new GBConstraints("x=0;y=2"));
label23.set("text", "control");

textfield135 = new TextFieldShadow();
textfield135.set("name", "textfield135");
gbpanel19.add(textfield135);
textfield135.set("GBConstraints", new GBConstraints("x=1;y=2;fill=horizontal"));
textfield135.set("text", "7ef4");
textfield135.set("numColumns", new Integer(4));

label24 = new LabelShadow();
label24.set("name", "label24");
gbpanel19.add(label24);
label24.set("GBConstraints", new GBConstraints("x=2;y=2"));
label24.set("text", "add.Offset 15..00");

textfield136 = new TextFieldShadow();
textfield136.set("name", "textfield136");
gbpanel19.add(textfield136);
textfield136.set("GBConstraints", new GBConstraints("x=3;y=2;fill=horizontal"));
```

```
textfield136.set("text", "ab23");
textfield136.set("numColumns", new Integer(4));

label25 = new LabelShadow();
label25.set("name", "label25");
gbpanel19.add(label25);
label25.set("GBConstraints", new GBConstraints("x=4;y=2"));
label25.set("text", "add.Offset 31..18");

textfield137 = new TextFieldShadow();
textfield137.set("name", "textfield137");
gbpanel19.add(textfield137);
textfield137.set("GBConstraints", new GBConstraints("x=5;y=2;fill=horizontal"));
textfield137.set("text", "a9da");
textfield137.set("numColumns", new Integer(4));

label26 = new LabelShadow();
label26.set("name", "label26");
gbpanel19.add(label26);
label26.set("GBConstraints", new GBConstraints("x=6;y=2"));
label26.set("text", "add.Offset 48..32");

textfield138 = new TextFieldShadow();
textfield138.set("name", "textfield138");
gbpanel19.add(textfield138);
textfield138.set("GBConstraints", new GBConstraints("x=7;y=2;fill=horizontal"));
textfield138.set("text", "cd92");
textfield138.set("numColumns", new Integer(4));

labelbar4 = new LabelBarShadow();
labelbar4.set("name", "labelbar4");
gbpanel19.add(labelbar4);
labelbar4.set("GBConstraints", new GBConstraints("x=0;y=5;width=8;fill=horizontal"));

label38 = new LabelShadow();
label38.set("name", "label38");
gbpanel19.add(label38);
label38.set("GBConstraints", new GBConstraints("x=0;y=4"));
label38.set("text", "ext 00&01");

label39 = new LabelShadow();
```

```
label39.set("name", "label39");
gbpanel19.add(label39);
label39.set("GBConstraints", new GBConstraints("x=0;y=6"));
label39.set("text", "ext 08&09");

textfield140 = new TextFieldShadow();
textfield140.set("name", "textfield140");
gbpanel19.add(textfield140);
textfield140.set("GBConstraints", new GBConstraints("x=1;y=4;fill=horizontal"));
textfield140.set("text", "####");
textfield140.set("numColumns", new Integer(4));

textfield141 = new TextFieldShadow();
textfield141.set("name", "textfield141");
gbpanel19.add(textfield141);
textfield141.set("GBConstraints", new GBConstraints("x=1;y=6;fill=horizontal"));
textfield141.set("text", "####");
textfield141.set("numColumns", new Integer(4));

textfield142 = new TextFieldShadow();
textfield142.set("name", "textfield142");
gbpanel19.add(textfield142);
textfield142.set("GBConstraints", new GBConstraints("x=3;y=4;fill=horizontal"));
textfield142.set("text", "####");
textfield142.set("numColumns", new Integer(4));

textfield143 = new TextFieldShadow();
textfield143.set("name", "textfield143");
gbpanel19.add(textfield143);
textfield143.set("GBConstraints", new GBConstraints("x=3;y=6;fill=horizontal"));
textfield143.set("text", "####");
textfield143.set("numColumns", new Integer(4));

textfield144 = new TextFieldShadow();
textfield144.set("name", "textfield144");
gbpanel19.add(textfield144);
textfield144.set("GBConstraints", new GBConstraints("x=5;y=4;fill=horizontal"));
textfield144.set("text", "####");
textfield144.set("numColumns", new Integer(4));

textfield145 = new TextFieldShadow();
```

```
textfield145.set("name", "textfield145");
gbpanel19.add(textfield145);
textfield145.set("GBConstraints", new GBConstraints("x=5;y=6;fill=horizontal"));
textfield145.set("text", "####");
textfield145.set("numColumns", new Integer(4));

textfield146 = new TextFieldShadow();
textfield146.set("name", "textfield146");
gbpanel19.add(textfield146);
textfield146.set("GBConstraints", new GBConstraints("x=7;y=4;fill=horizontal"));
textfield146.set("text", "####");
textfield146.set("numColumns", new Integer(4));

textfield147 = new TextFieldShadow();
textfield147.set("name", "textfield147");
gbpanel19.add(textfield147);
textfield147.set("GBConstraints", new GBConstraints("x=7;y=6;fill=horizontal"));
textfield147.set("text", "####");
textfield147.set("numColumns", new Integer(4));

label42 = new LabelShadow();
label42.set("name", "label42");
gbpanel19.add(label42);
label42.set("GBConstraints", new GBConstraints("x=2;y=4"));
label42.set("text", "ext 02&03");

label43 = new LabelShadow();
label43.set("name", "label43");
gbpanel19.add(label43);
label43.set("GBConstraints", new GBConstraints("x=2;y=6"));
label43.set("text", "labelext 10&11");

label44 = new LabelShadow();
label44.set("name", "label44");
gbpanel19.add(label44);
label44.set("GBConstraints", new GBConstraints("x=4;y=4"));
label44.set("text", "ext 04&05");

label45 = new LabelShadow();
label45.set("name", "label45");
gbpanel19.add(label45);
```

```
label45.set("GBConstraints", new GBConstraints("x=4;y=6"));
label45.set("text", "ext 12&13");

label46 = new LabelShadow();
label46.set("name", "label46");
gbpanel19.add(label46);
label46.set("GBConstraints", new GBConstraints("x=6;y=4"));
label46.set("text", "ext 06&07");

label47 = new LabelShadow();
label47.set("name", "label47");
gbpanel19.add(label47);
label47.set("GBConstraints", new GBConstraints("x=6;y=6"));
label47.set("text", "ext 14&15");

gbpanel23 = new GBPanelShadow();
gbpanel23.set("name", "gbpanel23");
gbpanel20.add(gbpanel23);
{
  int _tmp[] = {14,14};
  gbpanel23.set("rowHeights", _tmp);
}
gbpanel23.set("GBConstraints", new GBConstraints("x=2;y=19;width=9;fill=both"));
{
  int _tmp[] = {14,14,14,14,14,14,14,14};
  gbpanel23.set("columnWidths", _tmp);
}
gbpanel23.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("east"));
{
  double _tmp[] = {0.0,0.0};
  gbpanel23.set("rowWeights", _tmp);
}
{
  double _tmp[] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};
  gbpanel23.set("columnWeights", _tmp);
}

labelbar5 = new LabelBarShadow();
labelbar5.set("name", "labelbar5");
gbpanel23.add(labelbar5);
labelbar5.set("GBConstraints", new GBConstraints("x=0;y=0;width=8;fill=horizontal"));
```

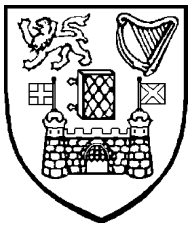
```
textfield148 = new TextFieldShadow();
textfield148.set("name", "textfield148");
gbpanel23.add(textfield148);
textfield148.set("GBConstraints", new GBConstraints("x=1;y=1;fill=horizontal"));
textfield148.set("text", "5ef1");
textfield148.set("numColumns", new Integer(4));

textfield149 = new TextFieldShadow();
textfield149.set("name", "textfield149");
gbpanel23.add(textfield149);
textfield149.set("GBConstraints", new GBConstraints("x=3;y=1;fill=horizontal"));
textfield149.set("text", "4ef1Pcket");
textfield149.set("numColumns", new Integer(4));

textfield150 = new TextFieldShadow();
textfield150.set("name", "textfield150");
gbpanel23.add(textfield150);
textfield150.set("GBConstraints", new GBConstraints("x=5;y=1;fill=horizontal"));
textfield150.set("text", "####");
textfield150.set("numColumns", new Integer(4));

textfield151 = new TextFieldShadow();
textfield151.set("name", "textfield151");
gbpanel23.add(textfield151);
textfield151.set("GBConstraints", new GBConstraints("x=7;y=1;fill=horizontal"));
textfield151.set("text", "eac5");
textfield151.set("numColumns", new Integer(4));

label48 = new LabelShadow();
label48.set("name", "label48");
gbpanel23.add(label48);
label48.set("GBConstraints", new GBConstraints("x=0;y=1"));
label48.set("text", " CRC ");

label49 = new LabelShadow();
label49.set("name", "label49");
gbpanel23.add(label49);
label49.set("GBConstraints", new GBConstraints("x=2;y=1"));
label49.set("text", "post");
```

```
label50 = new LabelShadow();
label50.set("name", "label50");
gbpanel23.add(label50);
label50.set("GBConstraints", new GBConstraints("x=4;y=1"));
label50.set("text", "reserved");

label51 = new LabelShadow();
label51.set("name", "label51");
gbpanel23.add(label51);
label51.set("GBConstraints", new GBConstraints("x=6;y=1"));
label51.set("text", " parity ");

gbpanel22 = new GBPanelShadow();
gbpanel22.set("name", "gbpanel22");
gbpanel20.add(gbpanel22);
{
  int _tmp[] = {14,14,14,14,14,14};
  gbpanel22.set("rowHeights", _tmp);
}
gbpanel22.set("GBConstraints", new GBConstraints("x=0;y=2;height=9;fill=both"));
{
  int _tmp[] = {14,14};
  gbpanel22.set("columnWidths", _tmp);
}
{
  double _tmp[] = {0.0,0.0,0.0,0.0,0.0,0.0};
  gbpanel22.set("rowWeights", _tmp);
}
{
  double _tmp[] = {0.0,0.0};
  gbpanel22.set("columnWeights", _tmp);
}

bscipacket = new ButtonShadow();
bscipacket.set("name", "bscipacket");
gbpanel22.add(bscipacket);
bscipacket.set("GBConstraints", new GBConstraints("x=1;y=2"));
bscipacket.set("text", "Update Packet");
bscipacket.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));
bscipacket.set("font", convert("java.awt.Font", "name=Dialog;style=plain;size=12"));
```

```
button4 = new ButtonShadow();
button4.set("name", "button4");
gbpanel22.add(button4);
button4.set("GBConstraints", new GBConstraints("x=1;y=4"));
button4.set("text", "Next Packet   ");
button4.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));

button5 = new ButtonShadow();
button5.set("name", "button5");
gbpanel22.add(button5);
button5.set("GBConstraints", new GBConstraints("x=1;y=5"));
button5.set("text", "Previous Packet");
button5.set("anchor", new sunsoft.jws.visual.rt.type.AnchorEnum("west"));

tfPacketID = new TextFieldShadow();
tfPacketID.set("name", "tfPacketID");
gbpanel22.add(tfPacketID);
tfPacketID.set("GBConstraints", new GBConstraints("x=1;y=1;fill=horizontal"));
tfPacketID.set("text", "23736");

label1 = new LabelShadow();
label1.set("name", "label1");
gbpanel22.add(label1);
label1.set("GBConstraints", new GBConstraints("x=1;y=0"));
label1.set("text", "Packet ID");

gbpanel7 = new GBPanelShadow();
gbpanel7.set("name", "gbpanel7");
tabbedfolder1.add(gbpanel7);
{
  int _tmp[] = {14};
  gbpanel7.set("rowHeights", _tmp);
}
{
  int _tmp[] = {14};
  gbpanel7.set("columnWidths", _tmp);
}
gbpanel7.set("layoutName", "SCI Simulation");
gbpanel7.set("visible", Boolean.FALSE);
{
  double _tmp[] = {0.0};
```

```
  gbpanel7.set("rowWeights", _tmp);
}
{
  double _tmp[] = {0.0};
  gbpanel7.set("columnWeights", _tmp);
}

gbpanel8 = new GBPanelShadow();
gbpanel8.set("name", "gbpanel8");
gbpanel1.add(gbpanel8);
{
  int _tmp[] = {14};
  gbpanel8.set("rowHeights", _tmp);
}
gbpanel8.set("GBConstraints", new GBConstraints("x=0;y=1;fill=both"));
{
  int _tmp[] = {14,14,14,14,14};
  gbpanel8.set("columnWidths", _tmp);
}
{
  double _tmp[] = {0.0};
  gbpanel8.set("rowWeights", _tmp);
}
{
  double _tmp[] = {0.0,0.0,0.0,0.0,0.0};
  gbpanel8.set("columnWeights", _tmp);
}

button1 = new ButtonShadow();
button1.set("name", "button1");
gbpanel8.add(button1);
button1.set("GBConstraints", new GBConstraints("x=3;y=0"));
button1.set("text", "Exit");

button2 = new ButtonShadow();
button2.set("name", "button2");
gbpanel8.add(button2);
button2.set("GBConstraints", new GBConstraints("x=4;y=0"));
button2.set("text", "Reset");

hostTextField = new TextFieldShadow();
```

```
    hostTextField.set("name", "hostTextField");
    gbpanel8.add(hostTextField);
    hostTextField.set("GBConstraints", new GBConstraints("x=2;y=0;fill=horizontal"));
    hostTextField.set("text", "CAGraidserver.cs.tcd.ie");
    hostTextField.set("numColumns", new Integer(30));

    databasehost = new LabelShadow();
    databasehost.set("name", "databasehost");
    gbpanel8.add(databasehost);
    databasehost.set("GBConstraints", new GBConstraints("x=1;y=0"));
    databasehost.set("text", "Database Host:");
  }

  // methods from lib/visual/gen/methods.java

  /**
   * Converts a string to the specified type.
   */
  private Object convert(String type, String value) {
    return(Converter.getConverter(type).convertFromString(value));
  }
}
```



Deep Trace DT200.1

Document History

Revision History

Revision 1 : Tuesday, 11 May 1999