

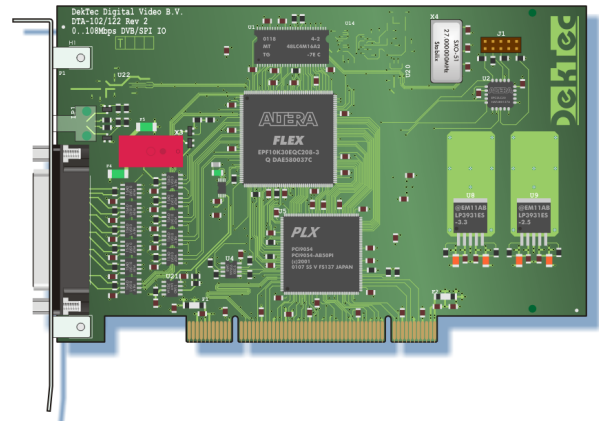
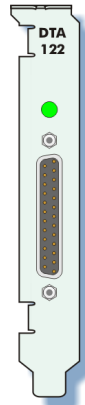


DVB/SPI Input Adapter for PCI Bus

- 0...108 Mbit/s
- 8-MBytes Buffer
- Support for External Signal Adapters

FEATURES

- High-speed Transport-Stream input, compliant to DVB/SPI (Synchronous Parallel Interface) as defined in DVB document A010 rev 1 and EN50083
- Special support for external signal adapters from e.g. ECL or TTL
- Input bit rate 0...108 Mbit/s
- LED indicator shows synchronisation status
- 8-Mbytes on-board buffer
- Optional FEC stripping
- Support for 188-byte and 204-byte packets, or arbitrary packet size in raw mode
- On-board PCI Bus master for off-loading host processor
- DMA burst-mode transfers for optimal usage of PCI Bus
- Periodic interrupt generator
- PCI rev 2.2, 32 bit, 33 MHz
- Industrial temperature range operation
- Scatter/Gather DMA for efficient handling of fragmented host memory



- Comes with free:
 - Windows-2000/XP device driver
 - API for developing custom applications
 - Example grabber code
 - DtGrabber: Windows grabber program

APPLICATIONS

- Universal DVB/ASI input adapter for applications that process MPEG-2 Transport Streams
- MPEG-2 Transport-Stream Analyser
- MPEG-2 Transport-Stream Recorder
- MPEG-2 Transport-Stream Monitor

KEY ATTRIBUTES

Parameter	Value
Physical Layer	DVB/SPI
DVB/SPI Connector	25-pin sub-D
DVB/SPI Clock	0...13.5 MHz
Input Bit Rate	0...108 Mbit/s
Packet Size in Bytes	188 or 204*
Target-Adapter Power Supply	5 V, 2 A
Current Firmware Version	3

* Arbitrary packet size in raw mode

RELATED PRODUCTS

Type	Description
DTA-102	DVB/SPI Output Adapter for PCI Bus
DTA-120	DVB/API Input Adapter for PCI Bus
DTC-320	Transport-Stream Analyser Software
DTC-7X2	Integrated Stream Analyser

Table of Contents

1. General Description	3	5.2.3. PerIntEn – Periodic Interrupt Enable	19
1.1. Typical Application	3	5.2.4. OvflntEn – Input-FIFO Overflow Interrupt Enable	20
1.2. Software	3	5.2.5. SynclntEn – Synchronisation-Error Interrupt Enable	20
1.3. Block Diagram	4	5.2.6. ThrlntEn – Threshold-Crossed Interrupt Enable	20
1.3.1. PCI-9054 Bus Interface.....	4	5.2.7. RateOvflntEn – Rate-Overflow Interrupt Enable	20
1.3.2. Input FIFO	4	5.2.8. EnaPwr – Enable Power.....	20
1.3.3. Register Set	4	5.3. Receive Status (Offset = 08h)	20
1.3.4. Target-Adapter Detection	4	5.3.1. PckSize – Packet Size.....	20
1.4. References.....	4	5.3.2. NumInv – Number of Invalid Bytes	21
1.5. Document Overview	5	5.3.3. SdramSize – SDRAM Size	21
2. External Interfaces	6	5.3.4. PerInt – Periodic Interrupt	21
2.1. Overview.....	6	5.3.5. Ovflnt – Input-FIFO Overflow Interrupt ...	21
2.2. LED Status Indicator	6	5.3.6. Synclnt – Synchronisation-Error Interrupt ..	22
2.3. DVB/SPI Output	6	5.3.7. Thrlnt – Threshold-Crossed Interrupt.....	22
2.3.1. Standard DVB/SPI Mode.....	6	5.3.8. RateOvflnt – Input-Rate Overflow	22
2.3.2. Target-Adapter Mode	7	5.4. FIFO Load (Offset = 18h)	22
2.4. Encoding of CODE Signal	7	5.5. Diagnostics (Offset = 1Ch).....	22
2.5. Target-Adapter Detection	8	5.5.1. LoopBack – Loop-Back Mode	23
3. Streaming Data	9	5.5.2. CdcRateOvf – Clock-Domain-Converter Rate Overflow	23
3.1. DMA vs. Direct Reads.....	9	5.5.3. RfRateOvf – Receive-FIFO Rate Overflow ..	23
3.2. Latencies	9	5.5.4. SdFull – SDRAM Full	23
3.2.1. PCI-Bus Latency	9	5.6. Loop-Back Data (Offset = 20h)	23
3.2.2. Interrupt Latency	10	5.7. Packet Count (Offset = 28h).....	23
3.2.3. Scheduling Latency.....	10	5.8. Clock Count (Offset = 2Ch)	24
3.2.4. Total Latency.....	10	5.9. Valid Count (Offset = 30h).....	24
3.3. Buffer Model	10	5.10. Threshold Control (Offset = 24h)	24
3.3.1. Input FIFO	11	5.10.1. UpdDone – Update Done	25
3.3.2. DMA Buffer	11	5.10.2. CmpA – Output Comparator A	25
3.4. Synchronisation	11	5.10.3. CmpB – Output Comparator B.....	25
3.5. Buffer Management	12	5.10.4. An/B – Update Threshold A or B.....	25
3.5.1. Ping-Pong, DMA-Driven	12	5.10.5. ThreshA – Threshold A.....	26
3.5.2. Start Up.....	13	5.10.6. ThreshB – Threshold B	26
4. Configuration Space	14	5.11. FIFO Data (Offset = 40h...7Ch).....	26
5. Target Address Space	16	6. Vital Product Data	27
5.1. General Control (Offset = 00h)	18	6.1. Serial EEPROM Lay-Out.....	27
5.1.1. PE – Program Enable.....	18	6.2. VPD ID String.....	27
5.1.2. PRE – Protect Register Enable	18	6.3. VPD Read-Only Resources	27
5.1.3. Reset – Software Reset	18	6.4. VPD Read-Write Resources.....	28
5.1.4. AlteraRev – Firmware Revision.....	18	6.5. Reading VPD Data	29
5.1.5. TypeNum – Type Number.....	18	6.6. Writing VPD Data.....	29
5.2. Receive Control (Offset = 04h).....	19		
5.2.1. RxMode – Receive Mode	19		
5.2.2. RxCtrl – Receive Control	19		

Copyright © 2001-2002 by DEKTEC Digital Video B.V.

DEKTEC Digital Video B.V. reserves the right to change products or specifications without notice. Information furnished in this document is believed to be accurate and reliable, but DEKTEC Digital Video assumes no responsibility for any errors that may appear in this material.

1. General Description

The DTA-122 is a PCI adapter card that reads a DVB/SPI-compliant Transport Stream for further processing in software. The transport rate of the incoming MPEG-2 data may have any value between 0 and 108 Mbit/s.

Note

- The DVB/SPI specification limits the maximum bit rate to 108 Mbit/s. However, the DTA-122 accepts data up to 150 Mbit/s.

1.1. Typical Application

The DTA-122 is typically deployed as DVB/SPI input card for MPEG-2 applications running on a PCI-based system. The low cost and high performance of generic computer platforms (e.g. Industrial PC¹) can be leveraged to create cost-effective digital-video solutions.

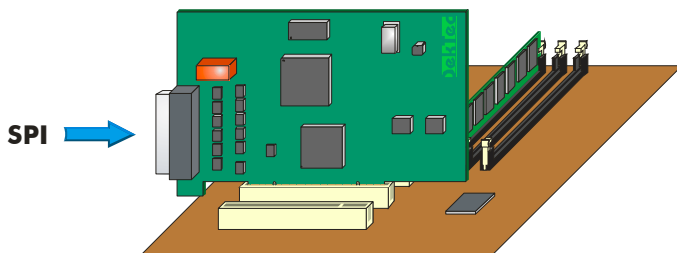


Figure 1. Typical application of the DTA-122 as DVB/SPI input stage in a PC-based digital-video application.

The DVB/SPI input signal makes the DTA-122 an excellent fit for direct connection to a demodulator (cable/satellite/terrestrial.)

The DTA-122 adds versatility by supporting so-called *Target Adapters*. A Target Adapter is a "little box" outside the DTA-122 that converts a custom Transport-Stream interface to a set of signals that can be connected to the DTA-122. The conversion may involve:

- Signal conversion (e.g. TTL to LVDS);
- Transformation of synchronisation signals;
- Custom physical connector.

The DTA-122 supports the use of a Target Adapter in the following ways:

- A special mode reuses two DVB/SPI pins to

¹ Application of the DTA-100 is not limited to the PC: Any platform that supports the PCI bus can be used.

supply power to the Target Adapter. No separate power supply is required!

- Identification of the target adapter. A simple scheme allows the identification of 35 different types of target adapters.
- Decoding of error signalling from target adapter to DTA-122.

1.2. Software

The DTA-122 hardware offering includes the following software:

- WDM device driver for Windows-2000 and Windows-XP;
- DTAPI library.

The WDM driver implements "low-level" operations that require direct access to the DTA-122 hardware, such as initiation and coordination of DMA transfers, handling interrupts and reading Vital Product Data (VPD, §6).

The DTAPI library is a thin layer of user-mode software that packages the driver functions into an easy to use API.

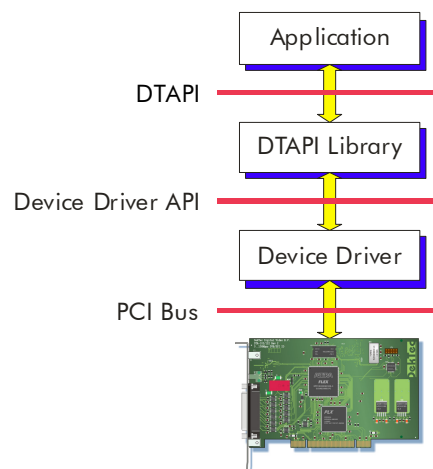


Figure 2. Software stack for DTA-122.

For customers that require a standard streaming solution, DEKTEC can offer a Transport-Stream Analysis and Monitoring application running on Windows-2000/XP. This program provides a convenient user interface for displaying the contents of a Transport Stream. Application features include:

- Full decoding of television signals;
- DVB PSI/SI decoding;
- Measurement of transmission rate and computation of MPEG-2 transport rate.

1.3. Block Diagram

Figure 3 shows a conceptual block diagram of the DTA-122. LVDS receivers interface the DVB/SPI signals to the *Input FIFO*, which buffers incoming transports packets. The *Master-Control* block relays the packets to the PCI bus via the *PCI-9054 Bus Interface*.

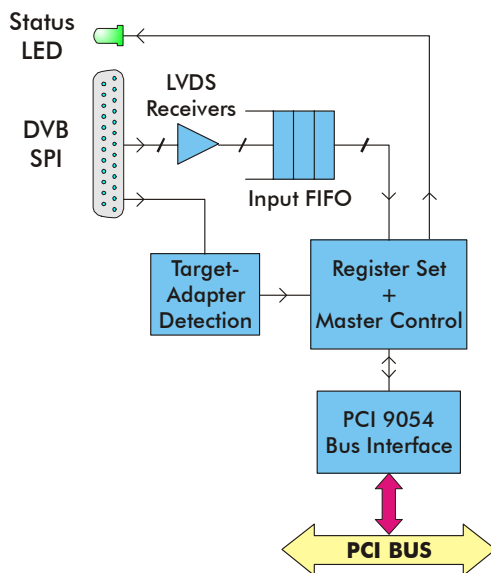


Figure 3. Conceptual block diagram of the DTA-122.

The *Target-Adapter Detection* block detects whether a Target Adapter is connected to the DVB/SPI input and, if so, which particular type.

1.3.1. PCI-9054 Bus Interface

The DTA-122 uses the PCI-9054 IC made by PLX for interfacing to the PCI Bus. The PCI-9054 also implements the DMA functions required for high-speed streaming of transport packets, with minimal host interaction.

The PCI-9054 supports large host buffers with *scatter/gather* DMA mode. Such buffers may get fragmented through allocation / deallocation of memory by OS-components. A *scatter/gather* list glues the buffer together without requiring software intervention.

Whenever appropriate, this specification provides information on the way the PCI-9054's registers should be used on the DTA-122. To obtain more details on the operation of the PCI-9054, please refer to the *PCI 9054 Data Book*.

1.3.2. Input FIFO

The DTA-122 contains a large (8-MBytes) buffer for incoming Transport-Stream data: the Input FIFO. A standard SDRAM is used to implement this buffer economically.

The Input FIFO enables reliable reception of a Transport Stream by providing buffer capacity while the PCI bus is temporarily unavailable, or while the application software is scheduled "out".

1.3.3. Register Set

Next to the registers in the PCI-9054, the DTA-122 contains a number of dedicated registers in PCI Memory Space (refer to §5 for syntax and semantics). With these Registers, the application software can configure and operate the DVB/SPI-specific features of the DTA-122.

1.3.4. Target-Adapter Detection

A target adapter is identified by a resistor to ground. The target-adapter detection circuit can be used to measure the value of the resistor, and thus the type of the target adapter.

When the target-adapter type has been determined, a "window" can be programmed around the resistor value. This way, the software can easily test whether the resistor value has changed to a value outside the window, indicating that the cable is disconnected, or that the target adapter signals an error.

1.4. References

- *Interfaces for CATV / SMATV Headends and Similar Professional Equipment, DVB DOCUMENT A010 rev.1, May 1997* – This is the original DVB document that specifies physical interfaces for the interconnection of signal processing devices for professional digital-television equipment. One of the interfaces described is DVB/SPI. DVB document A010 document has also been issued as CENELEC EN50083-9.
- *ISO/IEC 13818-1, Information technology – Generic coding of moving pictures and associated audio information: Systems, April 27th, 1995, also known as "MPEG-2 Sys-*

- tems” – Specification of the structure of a MPEG-2 Transport Stream.
- *DTAPI: C++ API for DTA-series of Digital-Video PCI-Bus Cards, DEKTEC Digital Video B.V., 2002* – Specification of **DTAPI**: the C++ interface to access the DTA-122 functions at a higher level of abstraction than would be possible using direct device-driver calls.
 - *PCI 9054 Data Book, PLX Technology, V2.1, January 2001* – Specification of the PCI 9054, the chip used on the DTA-122 to interface with the PCI bus. Use this document if you need to program the PCI-9054 directly, e.g. when writing a custom device driver.
The latest version of this document is available on line at <http://www.plxtech.com>.
 - *PCI Local Bus Specification, Revision 2.2, December 18, 1998* – Formal specification of the protocol, electrical, and mechanical features of the PCI bus.

1.5. Document Overview

This specification describes the details relevant for operating the DTA-122. The information herein is primarily intended for device driver writers and for software developers that have to access the DTA-122 directly from a real-time operating system.

The WDM device driver and DTAPI library encapsulate many programming details of the DTA-122. Users of DTAPI may find this document useful for providing background information, but do not need to master each and every detail.

- *Section 1* introduces the main features of the DTA-122.
- *Section 2* describes the physical interfaces of the DTA-122.
- *Section 3* provides a detailed description of synchronisation and buffer-management, in order to stream data efficiently and reliably.
- *Section 4* lists the PCI Configuration-Space registers supported by the DTA-122.

- *Section 5* describes the *operational registers* on the DTA-122. These registers can be used to control and monitor the streaming of digital-video data.
- *Section 6* defines the structure of *Vital Product Data (VPD)* as supported by the DTA-122 and other DEKTEC PCI cards.

2. External Interfaces

2.1. Overview

The DTA-122 supports a DVB/SPI connector with a LED status indicator, as shown in Figure 4 below.

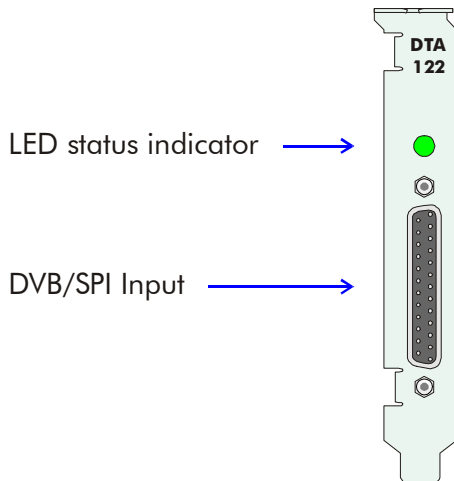


Figure 4. DTA-122 physical interfaces.

2.2. LED Status Indicator

The LED on the PCI-bracket of the DTA-122 is a bi-colour LED (red/green) that indicates the status of the DVB/SPI input signal.

Table 1. LED Indicator – Meaning	
LED Status	Meaning
Short green flashes	No signal (no clock)
Steady green	Valid Transport-Stream
Short orange flashes	No signal, power applied to target adapter
Steady orange	Valid Transport-Stream, power applied to target adapter
Short red flashes	Packet size not 188 or 204
Steady red	Packet size not 188 or 204, power applied to target adapter

Note

- Software may overrule the LED indication, and thus assign a different meaning to the LED Status patterns.

Just after power-up, the LED flashes a few times to indicate that the board is initialising. During this short period, the usual meaning of the LED indications does not apply. Instead, the LED pattern shows the board type and the firmware revision. An example start-up pattern is shown in Figure 5 below.

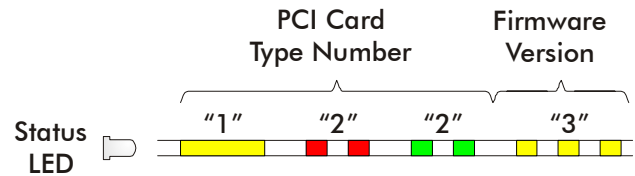


Figure 5. Power-up pattern for DTA-122 with (as example) firmware version 3.

2.3. DVB/SPI Input

The main interface of the DTA-122 is the DVB/SPI input on the 25-pin sub-D connector.

Two modes are supported:

- §2.3.1 Standard DVB/SPI Mode
- §2.3.2 Target-Adapter Mode

The mode is software-selectable with the *Enable-Power* field in the Transmit-Control register (§5.2.7).

2.3.1. Standard DVB/SPI Mode

In *Standard DVB/SPI Mode*, the Transport-Stream input of the DTA-122 conforms to the DVB/SPI specification (refer to §1.4 for references), with one small exception explained below the pin out of the 25-pin sub-D connector in Figure 6.

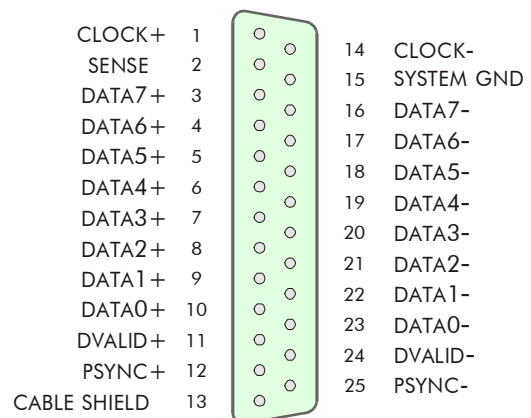


Figure 6. Pin assignment of 25-pin sub-D connector in *Standard DVB/SPI Mode*.

Notes

- The **CABLE-SHIELD** pin is connected to system ground on the DTA-122 board.
- In Figure 6, pin 1 is drawn in the upper-left corner. However, physically the pin is located at the bottom-right side (PCI bracket held in its normal position).

The usage of pin 2 on the DTA-122 is non-conformant. The DVB/SPI specification defines this pin as a **SYSTEM-GND** pin. The DTA-122 uses pin 2 as “**SENSE**” pin, to detect whether a target adapter is attached to the DTA-122. Target-adapter detection is discussed in §2.5.

Note

- The usage on pin 2 of **SENSE** instead of **SYSTEM GND** has one unfortunate side effect. When a DTA-102 is used as the DVB/SPI source for a DTA-122, both the sending DTA-102 and the receiving DTA-122 cannot detect that a cable is connected.

2.3.2. Target-Adapter Mode

The other mode supported by the DTA-122 is *Target-Adapter Mode*. A target adapter is a device that converts a custom Transport-Stream interface to DVB/SPI. In this mode, the DTA-122 supplies power to the target adapter, so that no additional power supply is required.

Target-Adapter Mode is effected when the Enable-Power field in the Transmit-Control register (§5.2.7) is set to ‘1’.

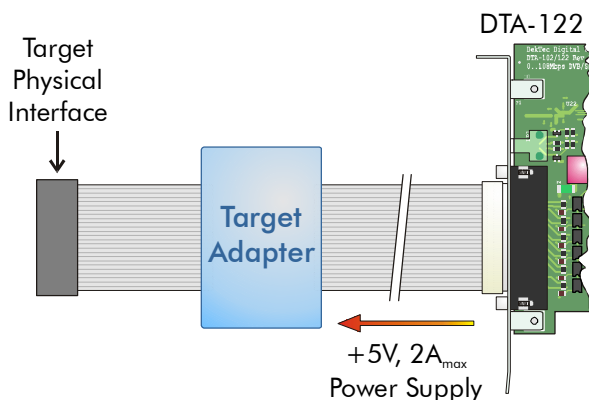


Figure 7. Target adapter connected to the DVB/SPI interface of the DTA-122.

The +5V power-supply is applied on two **POWER** pins, replacing the two pins that origi-

nally carried **PSYNC+/-**. Each power pin is fused and can carry a maximum current of 1A.

To keep all signal information available in Target-Adapter Mode, a special signal **CODE+/-** is introduced. The target adapter should encode **DVALID** and **PSYNC** into a single signal **CODE** (refer to §2.4 for encoding rules).

The altered pin out of the DVB/SPI connector in Target-Adapter Mode is shown in Figure 8.

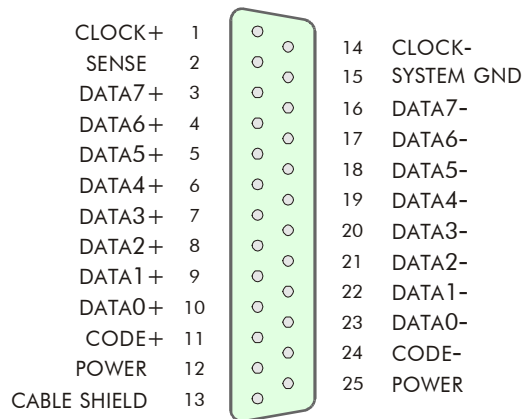


Figure 8. Pin assignment of 25-pin sub-D connector in Target-Adapter Mode.

2.4. Encoding of CODE Signal

In Target-Adapter Mode, both **DVALID** and **PSYNC** are encoded on the **CODE** signal.

The encoding of the signal on the **CODE** pin is listed in Table 2. The signal is synchronous to the rising edge of **CLOCK**. Commands are encoded in a simple serial format. While **CODE** is ‘0’, nothing changes. A ‘1’ is a “start-bit”, followed by a 1-bit command code.

Seq.	Mnem	Definition
0	NO_CHANGE	PSYNC: ‘0’ DVALID: unchanged
10	PULSE_PSYNC	PSYNC: 1-clock cycle pulse DVALID: set to ‘1’.
11	RESET_DVALID	PSYNC: ‘0’ DVALID: reset to ‘0’.

Figure 9 illustrates the relation between **PSYNC** and **DVALID** and the encoded **CODE** signal.

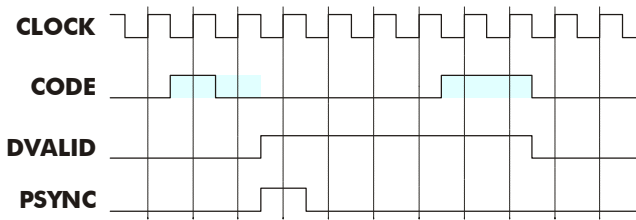


Figure 9. Timing diagram for encoding of **CODE** pin.

The two blue-shaded areas illustrate a **PULSE_PSYNC** and **RESET_DVALID** command respectively.

2.5. Target-Adapter Detection

A target adapter is identified by a resistor connected between **SYSTEM GND** and **SENSE** pin on the DVB/SPI connector.

The DTA-122 target-adapter detection circuitry has been designed to differentiate between 35 different resistor values. To achieve sufficient accuracy, the target-adapter resistor should be made from two 1% resistors in series. Refer to §5.10 for a description of target-adapter-detection circuitry, and how to measure the resistor value in software.

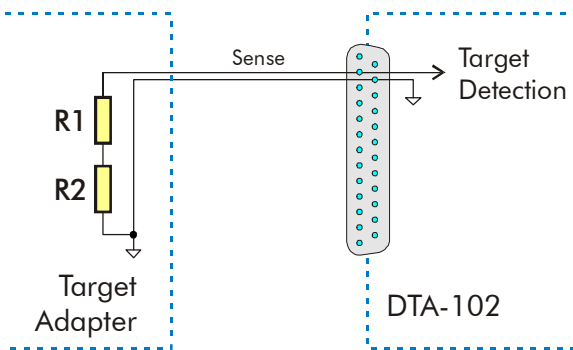


Figure 10. Two 1% resistors identify a target adapter.

Table 3 provides the combination of standard resistors that best approximate the “ideal” resistor value for each target-adapter number.

#	R1	R2	#	R1	R2
1	330	15	19	8200	150
2	560	39	20	8200	1000
3	820	47	21	10000	82
4	1000	150	22	10000	1000
5	1200	270	23	12000	150
6	1500	270	24	10000	3300
7	1800	330	25	12000	2700
8	2200	270	26	15000	1200
9	2700	120	27	18000	0
10	2700	560	28	18000	1800
11	3300	330	29	22000	220
12	3900	220	30	22000	2700
13	4200	390	31	27000	1000
14	4700	390	32	27000	4700
15	5600	56	33	33000	3300
16	5600	680	34	39000	2700
17	6800	100	35	47000	1800
18	6800	820			

Note

- Target-adapter numbers 1...15 (red-shaded cells) are reserved for use by DEKTEC. Target-adapter numbers 16...35 can be used for proprietary target adapters.

3. Streaming Data

The primary application of the DTA-122 is reading MPEG-2 Transport Packets for processing in a software application. The trickiest part of streaming data is achieving *real-time* operation, or at least – for streams without time stamps – achieving *efficient* operation. A well-balanced buffering scheme is required to compensate for hardware and software latencies.

This section first argues why DMA-based streaming offers superior performance compared to direct reading from the PCI Bus, followed by a description of hardware and software latencies that need to be overcome for real-time streaming. Then the buffer model adopted on the DTA-122 is described. Finally, synchronisation and buffer management are covered.

Using these techniques, §3.5 describes an approach for real-time operation of the DTA-122 in a relatively simple and robust way. This scheme has been implemented in the drivers that come with the DTA-122.

Note

- Of course, other approaches to real-time streaming with the DTA-122 may be feasible as well.

3.1. DMA vs. Direct Reads

The DVB/SPI Transport Stream that enters the DTA-122 is first buffered in the Input FIFO. Then, the transport packets need to be transferred to a buffer in host memory. The host can do this in one of two ways:

1. The host² directly reads data from the DTA-122's Input FIFO.
2. The DMA controller on-board of the DTA-122 transfers the data from the Input FIFO to a DMA Buffer in host memory.

In the first method, the host processor executes a long series of read instructions from the PCI Bus. Effectively, this slows down the processor speed to the PCI-Bus rate. Theoretically the

² This is: the application program or driver running on the host.

read-from-PCI instructions could be interleaved with other instructions, but in practice, this is awkward.

The second method allows the processor to read from main memory, which is at least an order of a magnitude faster than reading from the PCI Bus. The DMA cycles from PCI Bus to main memory– used to transfer the data from the DTA-122 – are invisible to the host processor.

To sum up, the “direct-read” method is simple, but ties the processor to the PCI timing. The DMA method is more complex, yet much faster as a consequence of the vast difference in speed between reading from main memory and reading from the PCI Bus.

The remainder of this section elaborates techniques to transfer data using DMA.

3.2. Latencies

MPEG-2 streams are very sensitive to the loss of even a single packet. Avoiding Input FIFO overflow, which leads to packet loss, is essential for flawless operation in most MPEG-2 applications.

This section describes the hardware- and software- latencies that need to be taken into account when streaming MPEG-2 packets with the DTA-122.

3.2.1. PCI-Bus Latency

The DTA-122 shares the PCI Bus with other bus masters that also compete for PCI cycles. The DTA-122 DMA Controller may have to wait a certain amount of time – the *PCI-Bus latency* – before it can acquire the bus and begin a DMA transfer.

Under normal conditions, the maximum duration of PCI latencies is in the order of a few microseconds. On a heavily loaded PCI Bus, latencies can be longer, but practical experience indicates that in all but pathological cases (see note) 2 ms can be safely taken as the absolute maximum PCI-Bus latency.

Note

- Cases are known in which PCI latencies be-

come unbounded³. If real-time streaming is required, it is essential to check the host system for such adversary conditions.

3.2.2. Interrupt Latency

Interrupt latency is the time between a hardware device raising an interrupt and software actually servicing the interrupt.

The DTA-122 hardware/software synchronisation methods (as described below in §3.4) rely on interrupts to signal certain hardware conditions to the software. The maximum interrupt latency has to be taken into account.

3.2.3. Scheduling Latency

The host CPU cannot dedicate all of its time to processing packets coming from the DTA-122: Other threads need processor cycles as well. *Scheduling latency* is the maximum time – in the worst-case scenario – the CPU still has to spend on other jobs, before it can service the DTA-122.

Scheduling latency is hard to grasp. It depends on many factors, like operating system, other software running on the host, relative priorities of threads and other hardware to be serviced.

3.2.4. Total Latency

Adding all up, the total latency is the sum of PCI-Bus-, interrupt- and scheduling- latencies. The total latency is the maximum time elapsing between an event that indicates that data should be transferred, and the time that data is actually copied to host memory.

The principal technique to avoid Input-FIFO Overflow is to ensure that the total latency is less than the time to completely fill the Input FIFO.

Thanks to the large size of the Input FIFO (8 MB), the maximum total latency that can be tolerated is relatively long. At the maximum input rate (108 Mbps), the maximum latency is still 621 ms.

3.3. Buffer Model

Incoming Transport-Stream data is buffered in a cascade of two buffers:

1. The *Input FIFO*, located on the DTA-122.
The DMA Controller on the DTA-122 transfers data bytes from the Input FIFO to the DMA Buffer.
2. The *DMA Buffer*, located in host memory.
The host processor reads data directly from this buffer.

The DMA Buffer should be divided in multiple (sub-)buffers, to avoid contention between host processor and DMA controller⁴. The DMA Controller writes to one DMA Buffer, while the host processes packets from another DMA Buffer.

A scheme with two DMA Buffers is elaborated in §3.5 below. Of course, advanced buffering schemes with more than two DMA Buffers may also be used⁵.

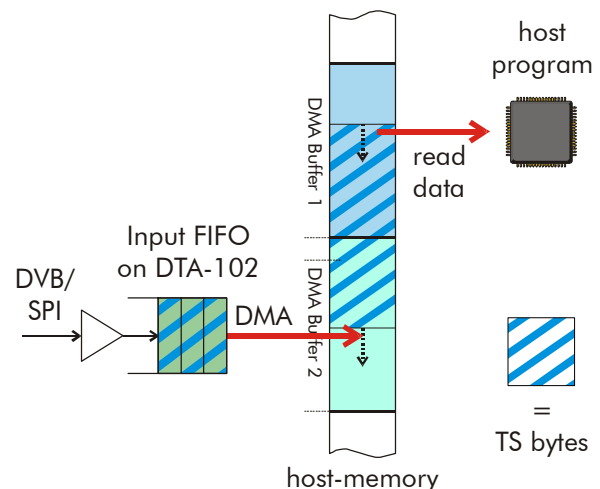


Figure 11. Double-buffer model. The Input FIFO is implemented in hardware on the DTA-122. The two DMA Buffers are located in host memory.

Figure 11 shows a snapshot of the double-buffer model in action. The buffer is split in DMA Buffer 1 and DMA Buffer 2, together forming one circular buffer. The blue-hatched

³ E.g. the host CPU writes continuously to a frame buffer on the PCI-Bus for a long period of time.

⁴ It is possible to use a single DMA Buffer, but then CPU read operations from the buffer and DMA write operations to the buffer should be mutually exclusive in time.

⁵ This specification does not provide further details on configurations with more than two DMA Buffers.

area represents Transport-Stream data that still has to be processed by the host program.

MPEG-2 data in DMA Buffer 1 is processed by the program running on the host. At the same time, the DMA Controller transfers new packets from the Input FIFO on the DTA-122 to DMA Buffer 2.

When DMA Buffer 1 has been read empty, and DMA Buffer 2 has been filled completely, the function of both DMA Buffers is swapped.

3.3.1. Input FIFO

Transport packets entering the DTA-122 are first buffered in the *Input FIFO*.

The Input FIFO is implemented on the DTA-122 with an 8-MByte SDRAM. For all practical purposes, the Input FIFO can be considered a large conventional FIFO that can buffer 8-Mbytes of packet data.

3.3.2. DMA Buffer

A DMA Buffer is an array of bytes allocated in the application's address space. The start- and end- address of a DMA Buffer must be aligned on 4-byte boundaries. The first byte that enters the DTA-122 is stored at relative address 0, the second byte at address 1, etc.

Obviously, the DMA Buffer may not be virtual memory that is swapped out to disk. Either non-paged memory should be used, or the driver should ensure that the pages are locked into physical memory whenever the DTA-122's DMA Controller may write to them.

A DMA Buffer maps to a contiguous address range in virtual-address⁶ space. The DMA Buffer needs not be contiguous in physical-address space: Memory pages may be *scattered* over physical memory⁷. The PCI-9054 Scatter/Gather DMA mode can be used to transfer such a scattered DMA Buffer in one go, without requiring processor intervention to glue pages together.

⁶ It is not a strict requirement that the DMA Buffer is contiguous in virtual address space. Nonetheless, application programmers will find it very convenient.

⁷ This means that the DMA Buffer may be allocated from a fragmented memory pool.

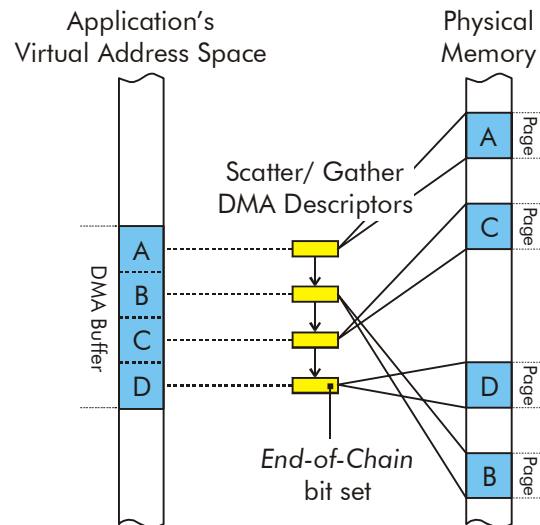


Figure 12. The DMA Buffer appears contiguous to the application, while pages are scattered over physical memory. Scatter/gather DMA allows writing to many pages in the DMA Buffer in one go without processor intervention.

Scatter/Gather DMA uses a list of *Scatter/Gather DMA Descriptors* stored in (non-paged) host memory. This so-called *scatter/gather list* can be built at the same time as the DMA Buffer is allocated.

The last descriptor in the list shall have its *End-of-Chain* bit set. The corresponding interrupt can be enabled, so that the driver is alerted when the DMA Buffer has been filled.

The descriptor syntax and the way to initialise and operate scatter/gather DMA are described in the PCI-9054 data book. Note that the DTA-122 uses demand-mode DMA. This implies that DMA channel 0 shall be used for performing the DMA transfers.

Note

- The scatter/gather mechanism incurs a little overhead per descriptor. Therefore, scatter/gather buffers should not be made too small, as this will lead to degraded performance. Buffers with the size of a memory page are fully acceptable.

3.4. Synchronisation

The DTA-122 reads Transport Packets, while the host processor processes them. Obviously,

packet reception must be synchronised to packet processing, or discontinuities will occur.

The DTA-122 hardware is tailored for *DMA-Driven* synchronisation: The host software locks packet processing to the completion of DMA transfers. Refer to §3.5.1 for how this works in combination with buffer management.

DMA transfers from Input FIFO to DMA buffer can and should be initiated before the Input FIFO contains sufficient data to complete the DMA transfer. In this manner, the DMA-done interrupt can be used to synchronise the host software to the incoming DVB/SPI Transport Stream: The DMA-done interrupt fairly accurately represents the moment in time that another buffer load with Transport Packets has been read by the DTA-122.

DMA-driven synchronisation works reliably because the DTA-122 hardware implements *demand-mode* DMA: DMA transfers are requested on the PCI Bus only as long as the DTA-122's Input FIFO has data available. When the FIFO becomes empty, the DMA process stalls. When new data enters the Input FIFO again, DMA resumes.

In other words: The Input FIFO cannot underflow in DMA-driven operation. The handshaking hardware prevents this from happening.

3.5. Buffer Management

This section discusses how to manage DMA Buffers such that synchronisation of packet reception by the DTA-122 and packet processing by the host is achieved.

3.5.1. Ping-Pong, DMA-Driven

As explained in §3.3, efficient streaming of data to the DTA-122 requires at least two DMA Buffers. The DMA Controller on the DTA-122 writes packets from the Input FIFO to one buffer. At the same time, the host program reads and processes packets from the other buffer. When both DMA is done and the packets in the other buffer have been processed completely, the DMA Buffers swap function. This process continues ad infinitum.

Note

- An advanced buffer-management scheme

may use more than two DMA Buffers. However, for the majority of applications a double-buffering will suffice.

The use of two buffers that swap function after each cycle – also known as *Ping-Pong* buffering – is illustrated in Figure 13.

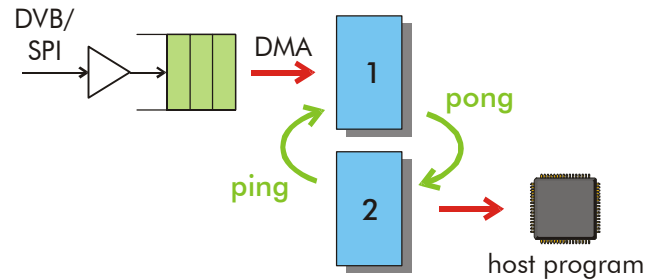


Figure 13. Ping-Pong buffering. The DMA Controller writes packets from the Input FIFO into one buffer, while the host reads packets from the other buffer. When both are finished, the “Ping-Pong” swap is executed.

DMA-Driven flow control in a double-buffering scheme is illustrated in the message-sequence chart shown in Figure 14.

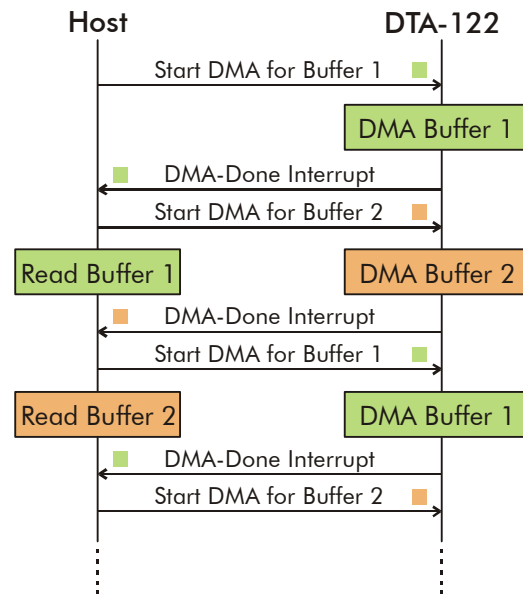


Figure 14. Ping-Pong buffer management using DMA-Driven flow control. After the host has read a buffer and the DMA write to the other buffer is done, the buffers swap function.

The DMA-Done Interrupt is the handshake signal. It triggers the host to read and process

new packets, and to initiate a new DMA transfer.

Note

- While waiting for the DMA-Done Interrupt, it is opportune for a device driver to sleep the process and give another process a chance to run.

It is instructive to ponder on the limiting factor in the Ping-Pong process: Host or DMA?

DMA has to be the limiting factor. The effective average rate of DMA is equal to the receive rate. The host has to be able to read and process packets faster than packets enter the DTA-122, otherwise long-term operation cannot be sustained. So, on average the host finishes reading a buffer before DMA to the other buffer is done.

3.5.2. Start Up

The following procedure is recommended to start-up the Ping-Pong process:

1. Allocate DMA Buffers, allocate and initialise Scatter/Gather DMA Descriptors.
2. Reset the DTA-122. This will clear the Input FIFO and reset the Receive-Control field (§5.2.2) to **Idle**.
3. Initiate the first DMA transfer from Input FIFO to DMA Buffer.
As the Input FIFO is still in idle mode, no data will actually be transferred yet.
4. Set the Receive-Control field to **Rcv**.
At this time, Transport Packets are allowed to enter the Input FIFO. Shortly after, the first data will be transferred to the DMA Buffer.
5. Normal Ping-Pong Operation has been entered.

4. Configuration Space

The DTA-122 acts as a single logical PCI Bus device. It implements the configuration registers required for identifying the device, control PCI Bus functions, and provide PCI Bus status.

Table 5 displays the address map of registers defined in configuration space:

- Black fields indicate configuration registers

supported by the DTA-122.

- Red-text cells represent registers supported by the PCI bridge chip, but not used for operating the DTA-122.
- Grey-text cells represent registers defined in the PCI Local Bus Revision 2.2 specification, but not supported on the DTA-122.

Table 4. Configuration Space – Address Map

Address Offset	Byte			
	3	2	1	0
00h	Device ID		Vendor ID	
04h	Status Register		Command Register	
08h	Class Code			Revision ID
0Ch	BIST	Header Type	Latency Timer	Cache Line Size
10h	PCI Base Address 0; used for memory-mapped configuration registers (PCI 9054)			
14h	PCI Base Address 1; not used			
18h	PCI Base Address 2; used for memory-mapped operational registers ⁸			
1Ch	PCI Base Address 3; not used			
20h	PCI Base Address 4; not used			
24h	PCI Base Address 5; not used			
28h	Card Bus CIS Pointer; not supported			
2Ch	Subsystem ID		Subsystem Vendor ID	
30h	Expansion ROM Base Address Register; not used			
34h	Reserved			Next_Cap = 40h
38h	Reserved			
3Ch	Maximum Latency	Minimum Grant	Interrupt Pin	Interrupt Line
40h	Power Management Capabilities; not used		Next_Cap = 48h	Capability ID = 01h
44h	Power Management Register; not used			
48h	Hot Swap; not used		Next_Cap = 4Ch	Capability ID = 06h
4Ch	VPD; used for manufacturing / service		Next_Cap = 00h	Capability ID = 03h
50h	VPD; used for manufacturing / service			

Table 5 shows a short description of the registers in configuration space.

⁸ Operational registers are mapped in "Local Address Space 0" of the PCI 9054.

Table 5. Configuration Space – Register Overview

Register	Bits	RW*	Value	Short Description
Vendor ID	16	R	10B5h	Identifies PLX as manufacturer of the PCI interface chip.
Device ID	16	R	9054h	Identifies the PCI interface chip (PCI 9054).
Command Register	16	RW	-	Provides coarse control on the ability to generate and respond to PCI cycles.
Status Register	16	RWC	-	Status of PCI-Bus relevant events.
Revision ID	8	R	0	Revision number of your DTA-122.
Class Code	24	R	FF0000h	Generic function of the DTA-122.
Cache Line Size	8	R	16	System cache line size in units of 32-bit words.
Latency Timer	8	RW	-	Amount of time in PCI-Bus-clock units that the DTA-122 may retain ownership of the PCI Bus.
Header Type	8	R	0	Specifies layout of configuration addresses 10h through 3Fh and single / multiple functions.
BIST	8	R	0	PCI Built-In Self Test (BIST).
PCI Base Address 0	32	RW	-	Memory attributes and base memory address for memory accesses to PCI 9054 registers
PCI Base Address 2	32	RW	-	Memory attributes and base memory address for memory accesses to <i>Local Address Space 0</i> , which is used to access the DTA-122's operational registers (Refer to Table 6).
Subsystem Vendor ID	16	R	14B4h	Identifies the manufacturer of the DTA-122. Subsystem Vendor ID and Subsystem Device ID are leased from Philips BE.
Subsystem Device ID	16	R	D116h	Identifies the PCI card as a DTA-122.
Interrupt Line	8	RW	-	Interrupt line routing information.
Interrupt Pin	8	R	01h	Interrupt pin used by the DTA-122.
Minimum Grant	8	R	10h	Length of time (in 250-ns units) the DTA-122 would like to retain master ship of the PCI Bus.
Maximum Latency	8	R	1Ah	Frequency in which the DTA-122 would like to gain access to the PCI Bus.

5. Target Address Space

The DTA-122's operational registers are mapped in Local-Address Space 0 of the PCI 9054. The PCI Base address of these registers is specified in BAR2. All accesses to the operational registers shall be 32-bit transfers.

Address Offset	Byte							
	3	2	1	0				
00h	0 0 0 0	0 0 0 0	General Control					
04h	0 0 0 0	0 0 0 0	0 0	Receive Control				
08h	0 0 0 0	0 0 0 0	0 0 0 0	0	Receive Status			
0Ch ... 14h	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
18h	0 0 0 0	0 0 0 0	FIFO Load					
1Ch	Diagnostics							
20h	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	Loop-Back Data	
24h	0 0 0 0	0 0 0 0	Threshold Control					
28h	Packet Count							
2Ch	Clock Count							
30h	Valid Count							
34h ... 3Ch	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
40h ... 7Ch	FIFO Data							

Register	Field	Bit Pos	#	RWC*	Short Description
General Control	PE	0	1	RW	Serial EEPROM Program Enable
	PRE	1	1	RW	Serial EEPROM Protect Register Enable
	Reset	2	1	W	Soft Reset of DTA-122 circuitry
	<i>reserved</i>	7...3	5	R	Not used
	AlteraRev	15...8	8	R	Firmware Revision
	TypeNum	23...16	8	R	Type Number: 122 for DTA-122
Receive Control	RxMode	1...0	2	RW	Receive Mode: St188/St204/StRaw
	<i>reserved</i>	4...2	3	R	Reserved for expansion of RxMode
	RxCtrl	5	1	RW	Receive Control
	<i>reserved</i>	7...6	2	R	Reserved for expansion of RxCtrl
	PerIntEn	8	1	RW	Periodic-Interrupt Enable
	OvfIntEn	9	1	RW	Input-FIFO Overflow Interrupt Enable
	SyncIntEn	10	1	RW	Synchronisation-Error Interrupt Enable
	<i>reserved</i>	11	1	R	Not used
ThrIntEn	12	1	RW	Threshold-Crossed Interrupt Enable	

Table 7. Operational Registers – Register Overview

Register	Field	Bit Pos	#	RWC*	Short Description
	RateOvfInt En	13	1	RW	Input-Rate Overflow Interrupt Enable
	<i>reserved</i>	16...14	3	R	Not used
	EnaPwr	17	1	RW	Enable Power to Target Adapter
	<i>reserved</i>	18	1	R	Not used
	LedControl	19	1	RW	Take over LED Control
	LedGreen	20	1	RW	State of Green LED (if LedControl is set)
	LedRed	21	1	RW	State of Red LED (if LedControl is set)
Receive Status	PckSize	1...0	2	R	Size of incoming packets: Rx188/Rx204/RxInv
	NumInv	3...2	1	R	Number of invalid bytes per packet
	SdramSize	7...4	4	R	SDRAM Size = Size of Input FIFO
	PerInt	8	1	RC	Periodic Interrupt
	OvfInt	9	1	RC	Input-FIFO Overflow Interrupt
	SyncInt	10	1	RC	Synchronisation-Error Interrupt
	<i>reserved</i>	11	1	R	Not used
	ThrInt	12	1	RC	Threshold-Crossed Interrupt
	RateOvfInt	13	1	RC	Input-Rate Overflow Interrupt
	RxClockDet	14	1	R	Receive-Clock Detected
FIFO Load	FifoLoad	23...0	24**	R	Current Load of Input FIFO in #bytes
Diagnostics	<i>reserved</i>	7...0	8	R	Not used
	LoopBack	8	1	RW	Loop-back mode
	CdcRateOvf	9	1	R	Clock-domain-converter Rate Overflow
	RfRateOvf	10	1	R	Receive-FIFO Rate Overflow
	SdFull	11	1	R	SDRAM Full
Loop-Back Data	LbData	7..0	8	W	Write data to Input FIFO
Packet Count	PckCnt	31...0	32	R	Sample of Packet Counter
Clock Count	ClkCnt	31...0	32	R	Sample of DVB/SPI Clock Counter
Valid Count	ValidCnt	31...0	32	R	Sample of DVB/SPI Valid-byte Counter
Threshold Control	UpdDone	0	1	RC	Update of threshold done
	CmpB	1	1	R	Output of comparator B
	CmpA	2	1	R	Output of comparator A
	An/B	3	1	RW	Update threshold A ('0') or threshold B ('1')
	<i>reserved</i>	7...4	4	R	Not used
	ThreshA	15...8	8	RW	Threshold A
	ThreshB	23...15	8	RW	Threshold B
FIFO Data	FifoData	31...0	4x8	R	Transport-Stream data: 4 bytes at a time

* R=Readable, W=Writeable, C=Clearable (clear when a '1' is written to bit position).

** Number of bits depends on SdramSize. Shown size (24-bits) is valid for 8-Mbyte SDRAM.

5.1. General Control (Offset = 00h)

Table 8. General-Control Register – Format

Bit	Mnem	Description
0	PE	Serial EEPROM Program Enable
1	PRE	Serial EEPROM Protect Register Enable
2	Reset	Soft Reset
7...3	<i>reserved</i>	
15...8	AlteraRev	Firmware Revision
23...16	TypeNum	Value=122
Bits 31...24 of this register are tied to '0'.		

The General-Control register has a common structure for all DTA-1xx PCI adapter cards. The register contains control bits that are independent of the specific functions of the board.

5.1.1. PE – Program Enable

The Program-Enable field directly controls the **PE** signal of the serial EEPROM. The EEPROM can only be programmed if this bit is set to '1'. In normal operation, **PE** should remain '0'.

5.1.2. PRE – Protect Register Enable

The Protect-Register-Enable field directly controls the **PRE** signal of the serial EEPROM. It enables the write-protection mechanism in the EEPROM. In normal operation, this field should remain '0'.

Warning

- Issuing a write-protection command to the serial EEPROM is an irreversible operation. Incautious use of the **PRE** bit may destroy the Vital-Product Data read/write capability!

5.1.3. Reset – Software Reset

Writing a '1' to the Reset bit issues a "soft" reset to the DTA-122. The following fields and logic circuitry are affected:

- RxCtrl in the Receive-Control register is reset to **Idle**.
- LedControl in the Receive-Control register is cleared.
- Interrupt-Status flags in the Receive-Status register are cleared, except PerInt.

- The entire contents of the Input FIFO are cleared.
- The FIFO-Load register is reset to zero.
- A number of internal state machines are reset.

Other fields in the operational registers are not affected, notably:

- Receive Mode (RxMode) in the Receive-Control register.
- Loop-Back Mode in the Diagnostics register.
- Interrupt-Enable bits in the Transmit-Control register: interrupts that were enabled remain enabled.
- PE and PRE in the General-Control register.

This behaviour is by design, so that the data-processing pipeline in the DTA-122 can be reset without compromising other processes running on the PCI card.

The Reset bit is write-only. The write operation triggers the reset action: it is not required to reset the bit to '0' again. The next time a '1' is written to the Reset bit, the board will be reset again.

5.1.4. AlteraRev – Firmware Revision

This read-only field identifies the current revision level of the firmware programmed into the Altera FPGA on-board of the DTA-122.

Note

- The Firmware Revision level is independent of the DTA-122 board revision (which can be read from VPD).

5.1.5. TypeNum – Type Number

The Type-Number field identifies the board in a straightforward way. For the DTA-122, the field's value is fixed to **122**.

Next to this field, the board's type number is also encoded in the Vital Product Data (VPD), which is the primary source of descriptive data. The purpose of the Type-Number field is to provide a convenient way for device drivers to distinguish between different kinds of DTA-1xx boards at start-up.

5.2. Receive Control (Offset = 04h)

The Receive-Control register contains a number of fields that allow the device driver to control receive-specific functions of the DTA-122.

Table 9. Receive-Control Register – Format

Bit	Mnem	Description
1...0	RxMode	Receive Mode
4...2	<i>reserved</i>	
5	RxCtrl	Receive Control
7...6	<i>reserved</i>	
8	PerIntEn	Periodic-Interrupt Enable
9	OvfIntEn	Input-FIFO-Overflow-Interrupt Enable
10	SyncIntEn	Synchronisation-Error-Interrupt Enable
11	<i>reserved</i>	
12	ThrIntEn	Threshold-Crossed Interrupt Enable
13	RateOvfIntEn	Input-Rate-Overflow-Interrupt Enable
16...13	<i>reserved</i>	
17	EnaPwr	Enable Power
18	<i>reserved</i>	
19	LedControl	Take over LED Control
20	LedGreen	State of Green LED
21	LedRed	State of Red LED
Bits 31...22 of this register are tied to '0'.		

5.2.1. RxMode – Receive Mode

Receive Mode is a 2-bit field that controls the processing applied to incoming packets.

Table 10. Receive Mode – Values

Value	Mode	Definition
00	St188	Store 188-byte packets.
01	St204	Store 204-byte packets.
11	StRaw	No notion of packets. All incoming data bytes are stored in the Input FIFO.

The default Receive Mode is **St188**. Packets are stored as 188-byte packets. If the input con-

tains 204-byte packets, the 16 trailing bytes are dropped, irrespective of their content.

In Receive Mode **St204**, the DTA-122 always stores 204-byte packets. If the input contains 188-byte packets, 16 zero bytes are appended.

In modes St188 and St204, packet synchronisation is based on the **PSYNC** signal, not on the value of the first byte of the packet: The value of **DATA** at a **PSYNC** pulse is stored in the input buffer, even if the value is not 0x47. Incoming packets must have a length of either 188 or 204 bytes. Packets with another length are dropped.

In Receive Mode **StRaw**, the DTA-122 ignores **PSYNC**. All data bytes for which **DVALID** is true are stored in the Input FIFO, irrespective of packet length.

5.2.2. RxCtrl – Receive Control

The Receive-Control field controls storage of data into the Input FIFO.

Table 11. Receive Control – Values

Value	Mnem	Definition
0	Idle	No new data is stored in Input FIFO.
1	Rcv	Store incoming data into Input FIFO.

After a power-up condition, Receive Control is initialised to **Idle**. Whenever Receive Control is set to **Idle**, the input circuitry is “disconnected” from the Input FIFO and no new data can be stored in the Input FIFO.

When Receive Control is set to **Rcv**, actual storage of transport packets in the Input FIFO begins.

5.2.3. PerIntEn – Periodic Interrupt Enable

Writing a '1' to this bit enables⁹ the *Periodic Interrupt* (§5.3.4).

⁹ To actually enable the Periodic Interrupt on the PCI Bus, the Local-Interrupt-Input-Enable bit in the PCI9054's Interrupt Control/Status register must also be set to '1'.

5.2.4. OvfIntEn – Input-FIFO Overflow Interrupt Enable

Writing a '1' to this bit enables⁹ the Input-FIFO-Overflow Interrupt (§5.3.5).

5.2.5. SyncIntEn – Synchronisation-Error Interrupt Enable

Writing a '1' to this bit enables⁹ the Synchronisation-Error Interrupt (§5.3.6).

5.2.6. ThrIntEn – Threshold-Crossed Interrupt Enable

Writing a '1' to this bit enables⁹ the Threshold-Crossed Interrupt (§5.3.7).

5.2.7. RateOvfIntEn – Rate-Overflow Interrupt Enable

Writing a '1' to this bit enables⁹ the Rate-Overflow Interrupt (§5.3.8).

5.2.8. EnaPwr – Enable Power

Writing a '1' to this field closes the relay so that +5V power is applied to the **PSYNC** input. At the same time the **DVALID** input changes to **CODE**, which encodes both **DVALID** and **PSYNC** in one signal. Refer to §2.4 for a description.

5.2.9. LedControl – Take over LED Control

When this field is '0', the state of the bi-colour LED indicator on the PCI bracket is determined by the hardware, as described in §2.2.

When this field is '1', the hardware is disconnected from the LED indicator. Instead, the LED is controlled directly by fields `LedGreen` and `LedRed`.

This bit is reset to '0' upon a hardware- or software reset.

5.2.10. LedGreen – State of Green LED

If `LedControl` is '1', this field controls the **green** colour of the bi-colour LED next to the connector on the PCI bracket.

5.2.11. LedRed – State of Red LED

If `LedControl` is '1', this field controls the **red** colour of the bi-colour LED next to the connector on the PCI bracket.

5.3. Receive Status (Offset = 08h)

The Receive-Status register contains a number of fields that allow the device driver to read status information from the DTA-122.

Table 12. Receive-Status Register – Format

Bit	Mnem	Description
1...0	PckSize	Size of incoming packets
3...2	NumInv	#Invalid bytes per packet
7...4	SdramSize	SDRAM Size
<i>Interrupt status flags</i>		
8	PerInt	Periodic Interrupt
9	OvfInt	Input-FIFO Overflow
10	SyncInt	Synchronisation-Error
11	<i>reserved</i>	
12	ThrInt	Threshold-Crossed
13	RateOvfInt	Input-Rate Overflow
14	RxClkDet	Receive-Clock Detected
Bits 31...15 of this register are tied to '0'.		

The interrupt-status flags (bit 8...13) in this register share common behaviour:

- An interrupt-status flag is set when the corresponding condition occurs. The flag remains set until it is explicitly cleared.
- Writing a '1' to the flag clears the interrupt-status flag, and also clears the PCI interrupt¹⁰ (unless another interrupt condition is pending).
- The interrupt-status flag only leads to an interrupt if the corresponding interrupt-enable bit in the Receive-Control register is set, and interrupts in the PCI 9054 have been enabled.
- The operation of the interrupt-status bits is independent from the state of the interrupt-enable bit: If the interrupt-enable bit is '0', the interrupt-status flag still latches the corresponding condition.

5.3.1. PckSize – Packet Size

Packet Size is a 2-bit field that indicates the length of the packets currently being received

¹⁰ No write action to a PCI-9054 register is required.

on the Transport-Stream input. The Packet-Size field is updated every packet.

Value	Mode	Definition
00	RxInv	Invalid packet size.
01	-	<i>reserved</i>
10	Rx188	Receiving 188-byte packets.
11	Rx204	Receiving 204-byte packets.

Packet-Size values **Rx188** and **Rx204** indicate that the DTA-122 receives correctly formatted packets with 188 or 204 valid bytes respectively.

Packet-Size value **RxInv** indicates that the DTA-122 is currently out of synchronisation with the input Transport Stream. Faulty packets are not stored in the input FIFO, except if the Receive Mode is **StRaw**, in which case every valid byte is stored.

Notes

- If input packets consist of 188 bytes with DVALID=1, followed by 16 bytes with DVALID=0, Packet Size will indicate **Rx188**, because the Packet-Size detection logic only considers valid bytes.
- Packet Size and Receive Mode (number of bytes stored per packet; §5.2.1) are *independent* of each other. For example, if Packet Size reads **Rx188**, Receive Mode may very well be set to **St204**. In this case, the DTA-122 receives 188-byte packets and stores them with 16 zero bytes appended to each packet.
- In Receive Mode **StRaw**, the Packet-Size field still provides a useful and valid value. For example, when 204-byte packets enter the DTA-122, Packet Size will indicate **Rx204**. If the input consists of e.g. 133-byte packets, Packet Size is **RxInv**, yet all valid bytes are stored in the Input FIFO as Receive Mode is **StRaw**.

5.3.2. NumInv – Number of Invalid Bytes

NumInv is a 2-bit field that provides an indication on the number of “invalid” bytes per packet. An invalid byte is a clock period in

which the **DAVLID** input level is ‘0’. The values supported for NumInv are listed in Table 14.

Value	#bytes	Definition
00	0	No invalid bytes.
01	16	16 invalid bytes per packet.
10	Other	Other number of invalid bytes per packet.
11	-	<i>reserved</i>

The DTA-122 does not provide information on the *location* of the invalid bytes. So, if the NumInv field indicates 16 invalid bytes per packet, the invalid bytes likely, but not necessarily positioned at the end of the packet.

This field is updated every packet.

5.3.3. SdramSize – SDRAM Size

SDRAM Size is a static read-only field that indicates the size of the SDRAM on-board of the DTA-122. The SDRAM size determines the maximum size of the Input FIFO.

Value	Size	Comment
0000	8 MB	Minimum supported size.
0001	16 MB	May be supported in future revisions of the DTA-122.
0010	32 MB	May be supported in future revisions of the DTA-122.
other	<i>reserved</i>	

5.3.4. PerInt – Periodic Interrupt

When set to ‘1’, this bit indicates that the *Periodic Interrupt* is pending. The Periodic Interrupt is generated automatically every 2²¹ clock cycles of the on-board 27 MHz reference clock. This corresponds to approximately once every 77.7 ms, or 12.87 times per second.

5.3.5. Ovflnt – Input-FIFO Overflow Interrupt

When set to ‘1’, this bit indicates that an overflow condition has occurred for the Input FIFO: The data in the Input FIFO could not be trans-

ferred fast enough to a system buffer in host memory (or to another PCI agent).

5.3.6. *Synclnt* – Synchronisation-Error Interrupt

When set to '1', this bit indicates that a synchronisation error has been detected in the packet-synchronising logic on the DTA-122.

In Receive Mode **StRaw**, no notion of packet size exists and therefore synchronisation-error checking is disabled.

5.3.7. *Thrlnt* – Threshold-Crossed Interrupt

When set to '1', this bit indicates that one of the threshold comparators has changed value. Thresholds are used in the target-adaptor detection process, refer to §5.10.

5.3.8. *RateOvfInt* – Input-Rate Overflow

When set to '1', this bit indicates that transport packets are entering the system faster than the DTA-122 can process the data. The maximum processing rate of the DTA-122 circuitry is at least 150 Mbit/s, so as long as the input rate remains below 150 Mbit/s, Input-Rate Overflow cannot occur¹¹.

Input-Rate Overflow (*RateOvfInt*) is a different kind of overflow condition than Input-FIFO Overflow (*OvfInt*):

- Input-Rate Overflow indicates that the maximum input rate is exceeded: the DTA-102 input circuitry cannot handle the incoming data rate.
- Input-FIFO Overflow indicates that the Input FIFO is *full*: the application program has read the Input FIFO too late.

5.3.9. *RxCikDet* – Receive Clock Detected

When set to '1', this bit indicates that the receive clock input on the DVB/SPI connector is toggling.

5.4. FIFO Load (Offset = 18h)

The FIFO-Load register contains the current load of the DTA-122's Input FIFO, expressed in number of bytes. Table 16 below is valid for an SDRAM size of 8 Mbytes.

Bit	Mnem	Definition
23...0	<i>FifoLoad</i>	Current FIFO load.
Bits 31...24 of this register are tied to '0'.		

While the DTA-122 is streaming data, the value read from this register is volatile. The value may change with every transmitted byte and with every DMA transfer.

Note

- The actual number of bytes buffered on the PCI card may be slightly higher than FIFO Load due to words residing in pipeline registers.

The maximum value of the FIFO-Load register is (approximately) the size of the Input FIFO, which is the SDRAM size plus 960 bytes.

The use of the FIFO-Load register in flow-control algorithms is optional. It can be used for enhancing robustness by checking at specific moments in time whether the FIFO Load is contained within a certain expected range.

5.5. Diagnostics (Offset = 1Ch)

The Diagnostics register contains a number of special fields that can be used for validation and testing of the DTA-122. In normal operation, this register should not be touched. It is recommended to clear the Diagnostics register to all zeros in the device-driver's initialisation routine¹³.

¹¹ DVB/SPI specifies a maximum input clock frequency of 13.5 MHz, corresponding to 108 Mbit/s. Nevertheless, the DTA-122 will work correctly up to 150 Mbit/s.

¹² Assuming an SDRAM-size of 8 MB. If the SDRAM is larger, more significant bits are included.

¹³ Issuing a soft reset through the General-Control register will also clear the Diagnostics register.

Table 17. Diagnostics Register – Format

Bit	Mnem	Definition
7..0	<i>reserved</i>	
8	LoopBack	Loop-back mode
9	CdcRateOvf	Clock-Domain-Converter Rate Overflow
10	RfRateOvf	Receive FIFO Rate Overflow
11	SdFull	SDRAM Full
31..12	<i>reserved</i>	

5.5.1. LoopBack – Loop-Back Mode

Writing a '1' to this bit disconnects the DVB/SPI input circuitry from the Input FIFO. This enables software to write a test pattern to the input FIFO through the Loop-Back-Data register (LbData).

In normal operation, this field should be set to '0'. Loop-Back Mode can be used in the manufacturing test to check data-path and memory integrity.

The Loop-Back-Mode field is not cleared by a software reset.

5.5.2. CdcRateOvf – Clock-Domain-Converter Rate Overflow

The `CdcRateOvf` flag indicates whether the clock-domain converter is in an overflow condition. The clock-domain converter is the first stage in the processing pipeline of the DTA-122. This type of overflow may occur if the input bit rate is (much) higher than 150 Mbps. This condition will result in the loss of data.

`CdcRateOvf` is one of the two conditions that can set the Rate-Overflow interrupt. The other condition is `RfRateOvf`.

5.5.3. RfRateOvf – Receive-FIFO Rate Overflow

The `RfRateOvf` flag indicates whether the rate at which data bytes enter the system exceeds the maximum write rate for the SDRAM. This type of overflow may occur if the input bit

rate is higher than 150 Mbps. This condition will result in the loss of data.

`RfRateOvf` is one of the two conditions that can set the Rate-Overflow interrupt. The other condition is `CdcRateOvf`.

5.5.4. SdFull – SDRAM Full

The `SdFull` flag indicates whether the SDRAM on-board the DTA-122 is full. If this flag is set, further writing to the SDRAM is inhibited (DVB/SPI input data will be dropped.)

5.6. Loop-Back Data (Offset = 20h)

The Loop-Back Data register can be used to write 8-bit test data to the Input FIFO in loop-back mode.

Table 18. Loop-Back Data Register – Format

Bit	Mnem	Definition
7...0	SfDataNxt	Write data to Input FIFO
Bits 31...8 of this register are tied to '0'.		

The purpose of this register is to enable diagnostics data-path-integrity and memory-test software.

Note

- Loop-Back Mode must be '1' for meaningful use of Loop-Back Data.
If Loop-Back Mode is '0', writes to Loop-Back Data have no effect.

5.7. Packet Count (Offset = 28h)

The Packet-Count register can be used to estimate the transport rate of the incoming DVB/SPI Transport Stream.

Table 19. Packet Count – Format

Bit	Mnem	Definition
31...0	PckCnt	Sample of packet counter.

The Packet-Count register contains a *sample* from a free-running counter that is incremented with every packet that enters the DTA-122.

The packet counter is sampled halfway between periodic interrupts (§5.3.8). Hence, the time interval between two samples is 2^{21} clock cycles of the on-board 27-MHz reference clock. The register is designed to be read just after each periodic interrupt, e.g. in an interrupt-service routine.

Note

- The absolute value of the Packet-Count register has no significance. Just the difference between two successive samples is relevant.

5.8. Clock Count (Offset = 2Ch)

The Clock-Count register provides a sample from a free-running counter that is incremented at every positive edge of the clock pin on the DVB/SPI connector.

Bit	Mnem	Definition
31..0	ClkCnt	Sample of DVB/SPI clock-pulse counter.

The Clock-Count register enables software measurement of the DVB/SPI clock frequency.

The operation of the Clock-Count register is similar to that of the Packet-Count register: The DTA-122 takes samples halfway between periodic interrupts and the software should read the register just after the periodic interrupt. The absolute value of the Clock-Count register has no significance.

Notes

- The DVB/SPI clock is a *byte* clock. The bit clock is a factor 8 higher.
- The clock counter is incremented for valid *and invalid* DVB/SPI bytes. The number of valid bytes can be obtained from the Valid-Count register, which is sampled at the same time as this register.

5.9. Valid Count (Offset = 30h)

The Valid-Count register contains a sample of a free-running counter that is incremented for every *valid* byte received at the DVB/SPI input.

Bit	Mnem	Definition
31..0	ValidCnt	Sample of valid-byte counter.

The Valid Count may be useful for special-purpose measurements. The register is not directly suitable for obtaining the transport rate, because “extra” bytes may have been inserted at the end of a transport packet. Use the Packet-Count register (§5.7) to measure the MPEG-2 transport rate.

Notes

- The Valid-Count register, just like the Clock-Count register, is expressed in *byte* units.

5.10. Threshold Control (Offset = 24h)

The Threshold-Control register contains a number of control- and status- fields to manage *target-adaptor detection*.

Bit	Mnem	Definition
0	UpdDone	Update of threshold done
1	CmpB	Output of comparator B
2	CmpA	Output of comparator A
3	An/B	Update A ('0') or B ('1')
7..4	<i>reserved</i>	Not used
15..8	ThreshA	Threshold A
23..15	ThreshB	Threshold B
Bits 31...24 of this register are tied to '0'.		

Each target adapter is identified by a specific resistor located in the target adapter, as described in §2.5. The DTA-122 hardware, with the help of software, measures the value of the external resistor, so that the target-adaptor type can be determined automatically.

Three special cases can be detected:

- If the resistor is temporarily shorted, the target adapter signals an error condition to the DTA-122. Target adapters may optionally

implement this feature with an NPN transistor parallel to the resistor.

- If the measured resistor value is close to 0Ω right from the start, then the DTA-122 is connected to a standard DVB/SPI sink.
- If the measured resistor value is ∞ (infinite), nothing is connected to the DTA-122.

A schematic diagram of the target-adapter detection circuitry, and the fields in the Threshold-Control register, are shown in Figure 15.

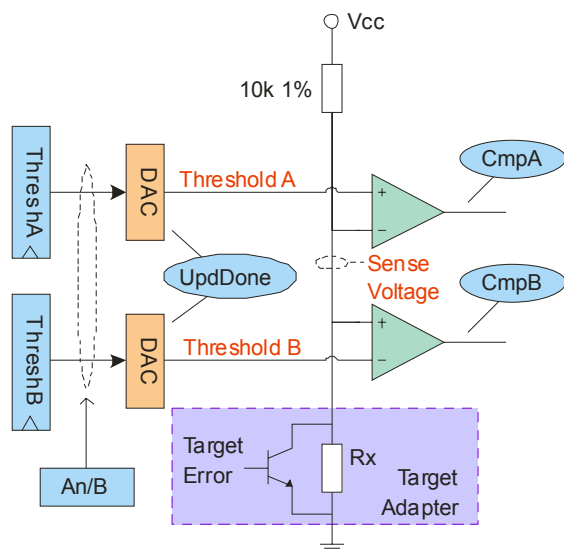


Figure 15. Target-adapter detection. Text on light-blue background refers to fields in the Threshold-Control register.

The target-adapter resistor is converted to the sense voltage using a resistor divider with a 10k 1% resistor to +5 Volts. The sense voltage is compared to two thresholds, which are set by two D/A Converters: Threshold A and Threshold B. The result of the comparison can be read through status flags CmpA and CmpB. The voltage can be determined by executing a binary search with one of the DACs and observing the comparator's output.

Once the resistor value is known, the state of the target adapter can be monitored by setting a window around the voltage induced over the target-adapter resistor. One DAC is set to the lower end of the window, the other to the higher end. By observing the comparator outputs, the software can be alerted to two events:

- The high end of the window is crossed. This event indicates that the cable to the target adapter is disconnected.
- The low end of the window is crossed. This event indicates that the target adapter is signalling an error.

5.10.1. UpdDone – Update Done

This status flag indicates that the value in one of the DACs (selected by the An/B field) has been updated. Waiting for this status flag is necessary because – after writing to the Threshold Register – it takes some time to transmit the value from the Threshold Register to the DAC.

After Update-Done has become '1', it is a requirement for the driver to wait another 10 μ s before reading the comparator output. The DAC and comparator need this period to settle to a stable value.

5.10.2. CmpA – Output Comparator A

This status flag holds the result of comparing the output of D/A Converter A and the sense voltage. If CmpA is '1', the sense voltage is lower than the DAC voltage.

5.10.3. CmpB – Output Comparator B

This status flag holds the result of comparing the output of D/A Converter B and the sense voltage. If CmpB is '1', the sense voltage is higher than the DAC voltage.

5.10.4. An/B – Update Threshold A or B

The An/B field determines whether D/A Converter A (An/B is '0') or D/A Converter B (An/b is '1') is updated.

Due to hardware limitations, the two D/A Converters cannot be updated in one go. To write both values, follow these steps:

1. Write the value for DAC A in field ThreshA while, in the same write operation, An/B is '0'.
2. Wait until field UpdDone is '1'.
3. Write the value for DAC B in field ThreshB while, in the same write operation, An/B is '1'.

4. Wait until field UpdDone is '1'.
5. Wait 10 μ s to let the DACs and comparators settle.

5.10.5. ThreshA – Threshold A

Value for the D/A Converter that generates Threshold A.

The value in this field is transferred to D/A Converter A when a value is written to the Threshold-Control register with An/B is '0'.

5.10.6. ThreshB – Threshold B

Value for the D/A Converter that generates Threshold B.

The value in this field is transferred to D/A Converter B when a value is written to the Threshold-Control register with An/B is '1'.

5.11. FIFO Data (Offset = 40h...7Ch)

The FIFO-Data register is connected to the output side of the DTA-122's Input FIFO. It's the main register for receiving MPEG-2 data with the DTA-122.

Table 23. FIFO-Data Register – Format

Bit	Mnem	Definition
7...0	FifoData	First received byte
15...8	FifoData	Second byte
23...16	FifoData	Third byte
31...24	FifoData	Last received byte

The suggested way to operate the DTA-122 is to program (in the PCI-9054) DMA transfers with the FIFO-Data register as constant source address¹⁴.

Instead of using DMA, an application program may also choose to read directly from the FIFO-Data register. In certain circumstances this may be a simple, convenient way to read data, e.g. for diagnostic purposes. However, performance will suffer, mainly because the host processor is required to read each indi-

vidual word. The recommended way to operate the DTA-122 is using DMA transfers.

Notes

- The FIFO-Data register appears at 16 consecutive word addresses in the Operational-Registers Memory Map. In principle, this enables read bursts on the PCI bus¹⁵, so that applications that choose not to use DMA can still achieve reasonable performance.

¹⁴ Local Address in the PCI-9054's gather/scatter DMA descriptor should be set to 00000040h and Local Addressing Mode to '1' (hold Local Address bus constant).

¹⁵ "In principle", because in many PCI system architectures, read bursts do not materialise. Reading may become very slow, in the order of 1 word per 700ns.

6. Vital Product Data

Vital Product Data (VPD) is information stored in a PCI device to uniquely identify the hardware and, potentially, software elements of the device. *PCI Local Bus Specification Rev2.2* defines both a standard storage structure and access method for VPD.

The DTA-122 uses VPD to store the serial number, revision level, etc. The sections below list all supported fields. The VPD is stored in the serial EEPROM on-board of the DTA-122. The VPD can be accessed through the VPD-function support built in the PCI-9054.

The DEKTEC DTA-series of PCI cards share the same layout of the serial EEPROM, so that the VPD data can be accessed in a uniform way for each board.

6.1. Serial EEPROM Lay-Out

Figure 16 below shows the memory map of the serial EEPROM and the positioning of VPD elements within the EEPROM memory. Note that addresses are *byte* addresses, whereas the PCI-9054 specification sometimes uses word (16-bit) or long word (32-bit) addresses.

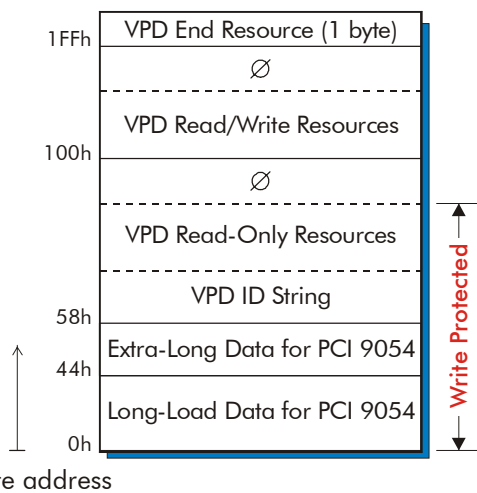


Figure 16. Memory Map of the Serial EEPROM.

The 88 bytes at address 0h...57h are loaded into the PCI-9054 upon power-on reset, to define the initial value of the register set. This part of the EEPROM data is not related to VPD.

The other 424 bytes in the serial EEPROM, starting at address 58h, are dedicated to VPD.

First, the VPD Identification String and the VPD Read-Only Resources are stored sequentially, followed by zero-byte stuffing. The VPD Read/Write Resources are located at address 100h, also followed by zero-byte stuffing. The EEPROM is "closed" by a VPD End resource at address 1FFh.

The VPD Read-Only and Read/Write Resources each consist of a two-character key (e.g. "PN" for Part Number) and a length-prefixed string defining the actual resource.

The first part of the serial EEPROM (register-data for PCI-9054, VPD ID String and VPD Read-Only Resources) is programmed in the factory and write-protected to avoid accidental modification. The VPD Read/Write Resources are not write-protected and may be modified by the end user, e.g. for storing the customer's system-asset identifier. DEKTEC utilises VPD Read/Write Resources to store software licenses.

6.2. VPD ID String

The VPD ID-String Resource contains the name of the board in ASCII characters. The content of this resource is fixed for all incarnations of the DTA-122.

Addr ¹⁶	Value	Item
58h	82h	ID-String tag.
59h	21h	Length LSB.
5Ah	00h	Length MSB.
5Bh	"DTA-122 DVB/SPI Input 0..108 Mbps"	ID-String data ¹⁷ .

6.3. VPD Read-Only Resources

The VPD Read-Only Resources describe permanent hardware characteristics of the PCI card. This VPD section is stored in the EEPROM just behind the VPD ID-String section.

¹⁶ Byte address in serial EEPROM.

¹⁷ No trailing zero ('\0') character!

The following VPD read-only resources are supported on the DTA-122:

- **PN = Part Number**
The PN Resource is fixed to "DTA-122".
- **EC = Engineering Change Level**
The EC Resource identifies the hardware revision of the board, e.g. "Rev 2".
- **MN = Manufacture ID**
The MN Resource is a 2-digit code¹⁸ identifying the manufacturer of the board.
- **CL = Customer ID**
The CL Resource is a 6-digit code¹⁸ identifying the (initial) customer of the board.
- **SN = Serial Number**
The SN Resource holds a unique serial number. For the DTA-122, this number begins with "4122", followed by a sequence number in 6 or more digits.
- **GC = Guard Code**
The GC Resource is a coded string used in DEKTEC license management.
- **PD = Production Date**
The PD Resource keeps the production date of this DTA-122 instance, e.g. "2001.11".
- **XT = Crystal Accuracy**
The XT Resource lists the accuracy of the 27-MHz crystal oscillator as a string, e.g. "100ppm".

Table 25 below shows an example of the contents of the VPD Read-Only Resources section. DEKTEC reserves the right to append private descriptors in the reserved part of the read-only area.

Addr	Value	Item
07Ch	90h	VPD-R tag.
07Dh	81h	Length LSB.
07Eh	00h	Length MSB.
07Fh	"PN"	VPD keyword.
081h	7	Field length.
082h	"DTA-122"	Part number.

¹⁸ DEKTEC internal code.

Table 25. VPD Read-Only Resources – Syntax

Addr	Value	Item
089h	"EC"	VPD keyword.
08Bh	5	Field length.
08Ch	"Rev 2"	Engineering-Change level.
091h	"MN"	VPD keyword.
093h	2	Field length.
094h	"03"	Manufacture ID.
096h	"SN"	VPD keyword.
098h	10	Field length.
099h	"4122000005"	Serial number.
0A3h	"CL"	VPD keyword.
0A5h	6	Field length.
0A6h	"300004"	Customer ID
0ACh	"GC"	VPD keyword.
0AEh	10	Field length.
0AFh	"TZ)/*L*QNZ"	Guard Code
0B9h	"PD"	VPD keyword.
0BBh	7	Field length.
0BCh	"2001.11"	Production Date.
0C3h	"XT"	VPD keyword.
0C5h	14	Field length.
0C6h	"5ppm@25C;15 ppm"	Crystal Accuracy.
0D4h	"RV"	VPD keyword.
0D6h	29h	Field length.
0D7h	CCh	Checksum.
0D8h	40 x 00h	Reserved.
100h		Read-Write section

The length of the VPD Read-Only Resources section is tuned such that the VPD Read/Write Resources section starts at byte address 100h.

6.4. VPD Read-Write Resources

The VPD read/write section can hold 255 data bytes that can be updated dynamically from software. Potential usage includes diverse applications such as software keying, system-asset identification and storage of fault codes for inspection by service personnel.

Every byte in the serial EEPROM can be rewritten about 10⁶ times. Therefore, the VPD Read/Write Resource cannot be used for data that is updated a lot, e.g. every second. It is recommended to use the Read/Write section only for data that has a near-static nature.

The following standard tags are defined in *PCI Local Bus Specification Rev2.2*.

- **Vx = Vendor Specific**
This is a DEKTEC-specific item, e.g. a software license. The second character (x) of the keyword can be 0 through 9 and A through Z.
- **Yx = System Specific**
This is a system-specific item. The second character (x) of the keyword can be 0 through 9 and B through Z.
- **YA = Asset Tag Identifier**
The resource contains the system-asset identifier provided by the system owner.
- **RW = Remaining Read/Write Area**
This descriptor is used to identify the unused portion of the read/write space.

The data bytes are stored in the serial EEPROM at address 100h up to 1FEh inclusive. The byte at address 1FFh is used to store the VPD-End tag. Table 26 below shows an example of the syntax of the VPD-Read/Write-Resources section.

Addr	Value	Item
100h	91h	VPD-W tag.
101h	FCh	Length LSB.
122h	00h	Length MSB.
103h	"V3"	VPD keyword.
105h	16	Field length.
106h	"3rS=;2kl`MD(#ac"	License string.
116h	"YA"	VPD keyword.
118h	25	Field length.
119h	"DVB/SPI Test Analyser #11"	System-asset identifier.

Addr	Value	Item
132h	"RW"	VPD keyword.
134h	202	Field length.
135h	202 x 00h	Reserved.
1FFh	78h	VPD End tag.

6.5. Reading VPD Data

VPD Resources can be read 4 bytes at a time with the procedure described below. The hardware does not support any form of parsing VPD data, this is the job of the device driver.

1. Ensure that the read address is 32-bit aligned: the two least-significant address bits shall be zero.
2. Write the address to the 16-bit PCI-9054 register **PVPDAD** at PCI offset 4Eh in PCI-Configuration Space. Set the **VPD-Address** field to the read-address field (last two bits 0) and, in the same operation, set the **F-flag** field to '0', signalling a read operation.
3. Poll the **F-flag** in a loop until it becomes '1'. This indicates that the VPD read data is actually available.
4. Read the 32-bit PCI-9054 register **VPDDATA** at PCI offset 50h to obtain the requested 4 VPD data bytes.
5. Repeat steps 1..4 for all VPD words to be read.

6.6. Writing VPD Data

The VPD Read/Write Resources section can be written 4 bytes at a time with the procedure described below.

1. Ensure that the write address is 32-bit aligned: the two least-significant bits shall be zero.
2. Enable programming of the serial EEPROM by writing a '1' to PE in the General Control register (§5.1.1).

3. Change the *Serial EEPROM Write-Protected Address Boundary* register in the PCI 9054 (register **PROT_AREA** at PCI-offset 0Eh¹⁹) to a value less or equal than the write address divided by four.
The division by four is required because **PROT_AREA** contains a 7-bit field that points to a 32-bit "long-word" address.
4. Write the desired data (32-bits!) to PCI-9054 register **VPDDATA**.
5. Write the destination address to the 16-bit PCI-9054 register **PVPDAD** at PCI offset 4Eh in *PCI-Configuration Space*.
Set the **VPD-Address** field to the write address (last two bits 0) and, in the same operation, set the **F-flag** to '1', which signals a write operation.
6. Poll the **F-flag** until it changes to '0' to ensure that the write operation has completed.
7. Repeat steps 1..4 for all VPD words to be written.
8. For safety, change **PROT_AREA** back to 7Fh, and:
9. Disable programming of the serial EEPROM by writing a '0' to **PE** in the General Control register.

Note

- It is the responsibility of the device driver to maintain integrity of the VPD Resources.
For example, if a VPD Resource to be rewritten does not start at a 32-bit boundary, then the host should first read the original 32-bit VPD word, AND/OR-in the new data, and write the resulting 32-bit word back.

¹⁹ In PCI-memory space.