DATA SHEET 7751 Encryption Processor

Hi/fn® supplies two of the Internet's most important raw materials: compression and encryption. Hi/fn is also the world's first company to put both on a single chip, creating a processor that performs compression and encryption at a faster speed than a conventional CPU alone could handle, and for much less than the cost of a Pentium or comparable processor.

Hi/fn, Inc.
750 University Avenue
Los Gatos, CA 95032
info@hifn.com
http://www.hifn.com
Tel: 408-399-3500
Fax: 408-399-3501

Hi/fn Applications Support Hotline:
408-399-3544

Disclaimer

Hi/fn reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

Hi/fn warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with Hi/fn's standard warranty. Testing and other quality control techniques are utilized to the extent Hi/fn deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

HI/FN SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of Hi/fn products in such critical applications is understood to be fully at the risk of the customer. Questions concerning potential risk applications should be directed to Hi/fn through a local sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

Hi/fn does not warrant that its products are free from infringement of any patents, copyrights or other proprietary rights of third parties. In no event shall Hi/fn be liable for any special, incidental or consequential damages arising from infringement or alleged infringement of any patents, copyrights or other third party intellectual property rights.

"Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals," must be validated for each customer application by customer's technical experts.

The use of this product may require a license from Motorola. A license agreement for the right to use Motorola patents may be obtained through Hi/fn or directly from Motorola.

Table of Contents

Figures

THIS PAGE INTENTIONALLY BLANK

1 Product Description

The Hi/fn® 7751 is a high-performance data encryption processor that may be used in a variety of data communication applications. This product implements data compression, data encryption, and data authentication algorithms including LZS®, MPPC, DES, Triple-DES, RC4, SHA-1, and MD5. These algorithms are found in many of the communication standards in use today.

The 7751 has a PCI Revision 2.1 compliant interface, allowing for direct connection to a PCI-based system. An external Serial EEPROM provides loadable PCI configuration space. Note: External EEPROM *required* to enable encryption and authentication processing units.

The 7751 will compress, encrypt, and authenticate fast enough to support up to twelve full duplex T1 (nine E1) communication links with a single chip, assuming a 2:1 compression ratio. Each data line may be channelized with a separate compression and encryption context.

Features

- 32 Bit, 33MHz PCI 2.1 compliant bus interface
- Pin Compatible with Hi/fn 9751 Compression Processor
- 5V or 3.3V PCI Support
- Supports DES, Triple-DES, RC4, LZS, MPPC, SHA-1, and MD5 algorithms
- No CPU intervention required between operations (no latency)
- Simple operation
- To facilitate exportability, device requires unlock procedure to enable encryption functions. Unlock procedure requires external EEPROM.
- Supports multiple sessions
- Descriptor based DMA engine (supports data scatter/gather)
- 3.3V operation with 5V tolerant I/O's

Part Number	Package
7751 PT6	144-pin TQFP

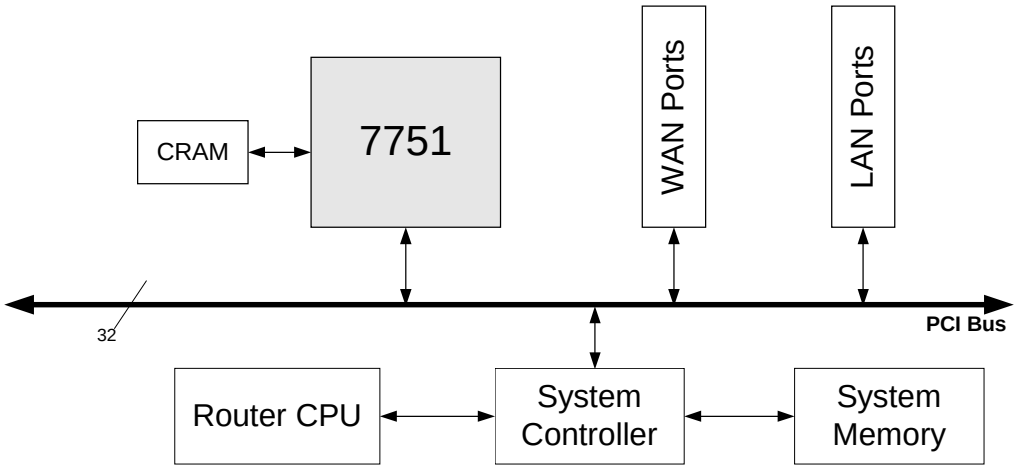


Figure 1. Example System Concept

2 Product Overview

The 7751 is a data compression, authentication, and encryption processor with a DMA bus master and a PCI interface.

Figure 2 shows the internal block diagram for the 7751. The 7751 consists of a PCI bus interface, two register sets, two FIFO's, and the processing units.

The PCI bus interface is a PCI 2.1 compliant interface which can be configured to control the amount of bus bandwidth consumed. There are no other bus interfaces in this device.

The two internal register sets are used for configuration and use of the 7751. The first register set contains the DMA Interface registers, which are used to configure and control DMA transfers in the 7751. The second register set contains the processing unit registers, which are used to configure and control the processing units in the 7751.

The 7751 has two internal FIFO's, the Source FIFO and the Dest FIFO, which are used to buffer data transfers to and from the 7751. Each FIFO is 64 bytes.

The 7751 has four processing units: a compression processing unit, an encryption processing unit, an authentication processing unit, and a padding processing unit. The order of the processing units is programmable, and will vary depending on whether the operation being performed is an encode or decode operation.

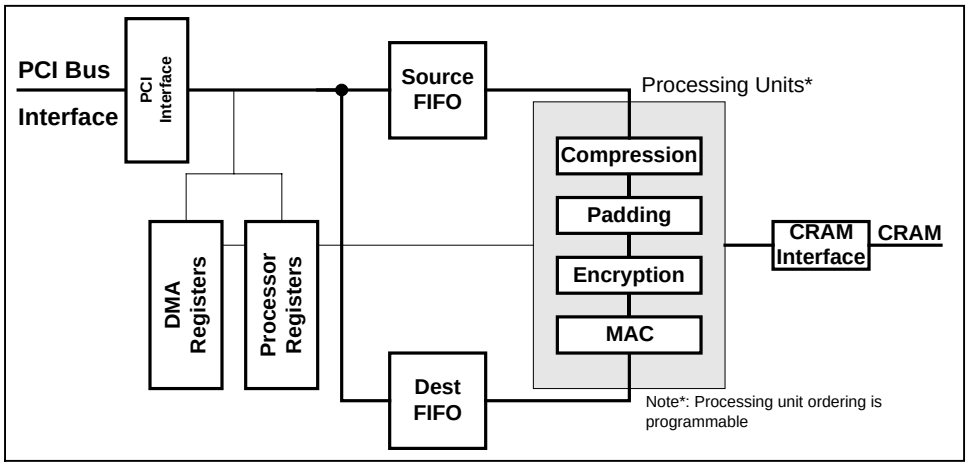


Figure 2. Internal Block Diagram

2.1 Command and Data Pipeline

The 7751 transfers commands, source context, and source data from system memory through the PCI bus interface. As the 7751 processes the command, it transfers destination data and result information to system memory from the device through the same PCI bus interface. Commands, context, data,and results are all formatted into structures that are passed to the chip or read from the chip via the PCI interface. The formats of these structures are defined later in this document.

The local RAM, called Context RAM, is used to store compression history information, and may optionally be used to store security information.

2.1.1 PCI Interface

The 7751 PCI interface is fully compliant with the PCI Local Bus Specification, revision 2.1. The PCI interface contains the configuration space the system uses during power on reset to configure the 7751. The 7751 supports PCI bus target and initiator cycles. The 7751's PCI interface is a memory-mapped target that the system uses for register accesses. The 7751's PCI interface is an initiator used by the DMA bus master to transfer data to and from system memory. The 7751 will attempt to transfer in 64-byte bursts.

The 7751 has the ability to place an upper limit on the amount of PCI bus bandwidth used during operations. This allows data transfers to and from the 7751 to be controlled to prevent the device from creating large data bursts on the PCI bus. This is accomplished by using a Polling Frequency counter that controls 7751 access to the PCI bus. The initial value in the Polling Frequency counter is loaded into the DMA Configuration register, which is described later in this document.

2.1.2 Bus Master DMA Controller

The 7751 bus master DMA controller utilizes four descriptor rings to transfer blocks of data between system memory and 7751. Two rings facilitate transfer of commands and source data from system memory into the device, and the other two rings facilitate transfer of destination data, and results to system memory out of the device. The bus master polls for valid descriptors that point to the system memory locations that the data is to be transferred. The descriptors also contain attributes describing the blocks of data to be transferred. The inbound DMA rings support gathering (using multiple descriptors for command blocks and data blocks). The outbound DMA rings support scattering (using multiple descriptors for data blocks and results blocks).

The 7751 has the ability to place an upper limit on the amount of PCI bus bandwidth that is consumed during DMA operations. This allows data transfers to and from the 7751 to be controlled to prevent the device from creating large data bursts on the PCI bus. This is accomplished by using two counters that control 7751 access to the PCI bus. The Polling Frequency controls how often 7751 pursues valid descriptors after it encounters a valid descriptor. The Invalid Poll Scalar counter controls how often 7751 looks for a valid descriptor after it encounters an invalid descriptor. Generally, the Polling Frequency is set to a high bus utilization, and the Invalid Poll Scalar is set to a low bus utilization. The initial values for these counters are loaded into the DMA Configuration register.

2.1.3 Data Flow

The 7751's DMA controller transfers data from system memory into the Source FIFO. Initially, the Source FIFO will be in the *command phase*, where the Source FIFO will accept a command structure. The command structure sent to the 7751 is defined in the Command Structure section. The command structure will be sent in the command block, which is pointed to by the command descriptor.

After the command structure has been written, the Source FIFO may enter the *context phase*. The context phase is optional and will be passed based on information sent in the Command Structure. The context information will be sent in the Source Data block. If multiple descriptors are utilized for context and source

data, then the system must set the LAST bit in the descriptor that points to the source data. Following the context information, the Source Data block will contain the actual source data to be processed by the 7751.

After data is processed by the engines, it enters the Dest FIFO. The first information to enter the Dest FIFO is the destination data. The destination data is the processed data from the command. The 7751 DMA controller transfers the destination data from the 7751 to system memory in the destination data block.

After the destination data has been transferred from the Dest FIFO, the Dest FIFO enters the *result phase*. While in the result phase, the Dest FIFO will transfer a result structure. The result structure is pointed to by the Result Descriptor. After the result structure has been transferred from the Dest FIFO, the FIFO will again enter the destination data phase, and the whole process repeats itself.

2.2 Operation

Once the descriptor rings are enabled, the 7751 will begin polling the command descriptor waiting for the VALID bit to be set. It is important to ensure that the system is ready to transfer data before setting the valid bit in the descriptor table. Upon reading a valid descriptor, the DMA transfer of the command block will begin. Byte alignment is supported on the command and source data rings, but not on the destination data and result rings.

When the entire command block has been transferred, the 7751 begins polling the source data descriptor and waits for the VALID bit to be set. The SOURCE LENGTH field in the source data descriptor must be the same value as The TOTAL SOURCE COUNT given in the Base Command structure. Upon reading a valid descriptor, the DMA transfer of the source data block will begin. When the last byte of the source data block has been transferred, the 7751 begins polling the command descriptor. This enables the system to pipeline commands to the 7751 for higher throughput operation.

Any Source Context information required for the command must be passed at the end of the Command block or the beginning of the Source Data block

As soon as the first bytes of source data enter the processing units, the 7751 begins its operation(s) and begins to fill the destination FIFO with destination data. As the destination FIFO fills, the 7751 begins to poll the destination data descriptor waiting for the VALID bit to be set. If the destination data descriptor is not ready, the current command will stall until the destination data descriptor becomes valid.

After the destination data block has been transferred, the result descriptor ring will be polled to see if it is valid, and the result information will be transferred.

The 7751 checks the Processing Unit Status register to determine which ring(s) to poll.

2.3 Processing Units

The 7751 consists of four processing units: Compression, Padding, Encryption, and MAC. The order of processing is determined by whether the operation is an encode or a decode. In addition, the location of the authentication (MAC) processing unit may be programmed to support processing various protocols.

Each processing unit performs the function that it has been programmed to do, as specified by the command issued. Although each processing unit has unique characteristics, all of them share several attributes.

Each processing unit may be enabled or disabled. A disabled processing unit simply passes data forward to the next processing unit without altering the data or modifying the context.

Each processing unit has two counters. The Header Counter determines how may bytes the processing unit will pass through (starting with the first byte of source data) before it begins processing the data. This counter is useful to skip header fields in many communication protocols.

The Source Counter determines how many bytes to process by that processing unit. The Source Counter may be programmed to start counting from the first byte to be processed (after skipping the header bytes), or it may be programmed to start counting after the last byte processed by the previous processing unit (as defined by the previous processing unit's source counter). This flexibility takes into account the variable output size produced by the compression and padding units.

Once the source count of a processing unit reaches zero, any remaining bytes in the input data stream are passed through that processing unit without altering the data or modifying the context.

In addition to the independent Header counters and Source Counters within each processing unit, there is one other counter, called the Total Source Counter. The Total Source Counter determines when a command is completed. Once the Total Source Counter exhausts its count, the command will complete, and the next command will begin to be processed.

2.4 Example Data Block

The 7751 processes blocks of data. The command associated with this block of data controls each processing unit by specifying what part of the block each unit must work on.

Through careful selection of the Header Count and Source Count values in each processing unit, relatively complex patterns of compression, encryption, and authentication may be performed.

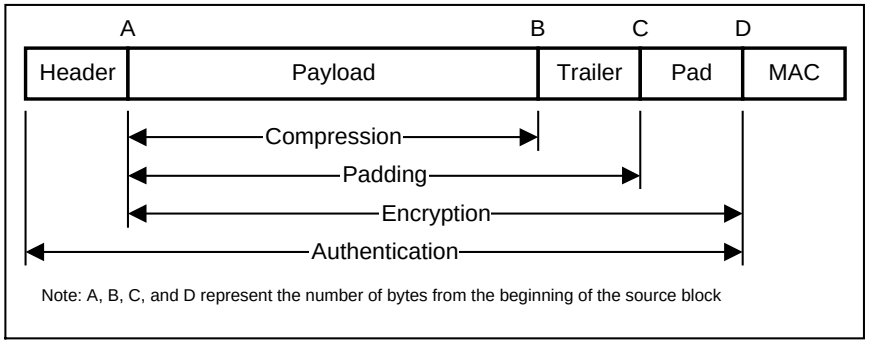


Figure 3. Example Data Block

The following simple example will help demonstrate how the 7751 processes a block of data.

Consider an encode operation being performed on the block of data shown in Figure 3. Although the location of the MAC processing unit is programmable, in this example the ordering of the processing units will be compression, padding, encryption, and MAC.

The header, payload, and trailer are the parts of this block of data that will enter the 7751 (the padding and MAC are added to the data by the 7751). As such, the total source count value is "C", as shown in Figure 4.

Command Structure Field	Length
Base	
Total Source	C
Compression	
Header Cnt	A
Source Cnt	B-A
Pad	
Header Cnt*	A
Count Mode	1
Source Cnt	C-B
Encrypt	
Header Cnt	A
Count Mode	1
Source Cnt	0
MAC	
Header Cnt	0
Count Mode	1
Source Cnt	0

Note *: Pad Header Count is taken from the Encryption Command Structure.

Figure 4. Command Length

Since the header of this data block is not to be compressed, the compression processing unit must skip over this header. The Header Cnt, "A", represents the number of bytes of data the compression processing unit will pass through with out compressing. Once the Header Cnt is passed through, the compression processing unit will compress Source Cnt bytes, "B-A". Note that "B" represents the number of bytes (passed in the command) that the compression engine will process. The value "B" returned in the results represents the number of compressed bytes.

The padding, encryption, and MAC processing units work in a similar fashion. The processing units will pass-through the number of bytes in the Header Cnt, then process Source Cnt bytes of data. Since the Count Mode is set to "1", these engines will begin to process data after the last byte exits the compression engine.

Figure 4 summarizes the count values for the different processing units in this example.

2.5 Destination Data Overrun

There are two types of destination overruns: processing unit-based overruns and descriptor-based overruns. The 7751 may be configured to respond to either or both types of destination overruns. Typically the detection of only one type of destination overrun is enabled. However, the detection of both types of destination overruns may be enabled if desired. Please see section 2.5 below.

2.5.1 Processing Unit-Based Destination Overrun

A processing unit based destination overrun condition occurs when the command produced more data than specified in the TOTAL DEST COUNT field of the Base Command Structure, and the IGNORE DEST COUNT bit was set to zero. If the Destination Counter attempts to decrement below zero, the command continues to process incoming data, but no additional data is output to the Destination FIFO. This does not cause the command to terminate, and the Destination Counter will remain at zero. After all source data in the command is processed, results will be produced as normal.

When a destination overrun condition occurs, the DEST OVERRUN bit in the Processing Unit's Status and Interrupt Status registers will be set. Additionally, the DEST OVERRUN bit in the Base Result Structure will be set when the results are produced.

Bit 9 in the Processing Unit Interrupt Enable register is defined as DEST OVERRUN INTERRUPT ENABLE. This bits may be used to generate error detection when a destination overrun occurs. To utilize this, the Engine IRQ (bit 0) in the DMA Interrupt Enable register must be also set to enable processing unit interrupts.

Note that the DEST OVERRUN bit will only be set momentarily in the Processing Unit Status Register, and may not be detectable. However, the DEST OVERRUN bit in the Processing Unit Interrupt Status Register will remain set until cleared. Clearing bits in the Processing Unit Interrupt Status and DMA Status and Control registers is described in the coresponding register descriptions in the 7751 data sheet.

Furthermore, the DEST OVERRUN condition will prevent the current encryption and MAC context from being saved in the context RAM. Compression context is always saved in local RAM. This effect on packet processing is described in the following section.

2.5.2 Processing Unit-Based Destination Overrun Effect On Same Sessions

Only the RC4 key and DES Initialization Vector (IV) change while data is processed. The other encryption and MAC keys remain constant during processing. Upon completion of a command (that does not cause a Destination Overrun), the internal RC4 key or DES IV is written to the Context RAM, as well as being stored in the encryption engine. When a command causes a destination overrun to occur, the internal RC4 key or DES IV is **not** written to the Context RAM, but it **is** stored in the encryption engine.

The effect of this is that when a destination overrun occurs, the RC4 key and DES IV stored in the encryption engine is not valid. If the command following a destination overrun is from a different session number, the engines will retrieve the appropriate context information from the context memory, which will overwrite the invalid

encryption information in the engine, and packet processing proceeds. However, if the command following a destination overrun is from the same session number, the engines will not retrieve the encryption information from the context memory - they will use the incorrect encryption information contained in the engines.

For security protocols that use MAC, DES, and 3DES keys, and also utilize new IVs for each packet (such as IPSec), the Destination Overrun condition does not affect the packet processing operation. However, the following note for setting new keys should be followed.

Note: To ensure that a new MAC, DES, 3DES, and RC4 keys and DES and 3DES IVs are not prevented from being stored to context RAM in the event of destination overflow, it is recommended that when setting a new key, perform a "dummy" encode (encryption and MAC) on zero bytes of source data to insure that Destination Overrun will not occur.

For security protocols that use RC4 or DES IVs that chain between packets, the Destination Overrun condition affects packet processing operation. When the destination overrun condition occurs, to restore the RC4 engine to the current key, either pass the key again in the next operation in the current session, or perform an RC4 operation with another session and then resume with an operation in the current session. Performing an RC4 operation in another session causes the 7751 to re-read the current RC4 key from context RAM when an RC4 operation in the current session is performed.

The same operations are used to restore the IV: either the last 8 bytes of ciphertext from the previous packet are passed with the next operation in the current session, or an operation in another session is performed, and then operation is resumed in the current session.

Programming the 7751 to process packets with RC4 (or chained DES IVs) can be done in one of two ways. The first method is to set the NEW KEY (or NEW IV) bit if the next command to use the engine has the same session number as the command causing the destination overrun. The second method is to interleave commands by using the engine with another session number before sending a command with the same session number that encountered the Destination Overrun. The new session number will cause the engine to retrieve the appropriate context information from the context memory, overwriting the invalid context information.

Please note that these solutions squelch transmission of the packet that expanded. Also, please note that certain protocols may not allow restoration of previously used keys or IVs. Also note that reprocessing the expanded packet in the session can be another solution for protocols that use RC4 or chained DES IVs; however, this will interrupt pipelined operations.

2.5.3 Descriptor-Based Destination Overrun

If bit 4 of the DMA Configuration register is set to one, then the CPU controls the LAST bit in the Destination Data descriptor, and the 7751 is configured to respond to descriptor-based destination overruns. These types of overruns occur when the amount of data transferred exceeds the DESTINATION DATA LENGTH field of the Destination Data descriptor (or the sum total of bytes if multiple destination data descriptors are used).

When a descriptor-based destination overrun occurs, the OVER bit in Destination Data Ring field of the DMA Status and Control register is set. Additionally, the OVER bit in the Destination descriptor and the DEST OVER bit in the Result descriptor are set when the 7751 writes to those descriptors.

Bit 25 in the DMA Interrupt Enable register is defined as the OVER INTERRUPT ENABLE. This bit may be used to generate error detection when a destination data overrun occurs.

2.6 Descriptor-Based Result Overrun

Additionally, there can be result overruns. These are analogous to descriptor-based data overruns, except for results.

If bit 4 of the DMA Configuration register is set to one, then the CPU controls the LAST bit in the Result descriptor, and the 7751 is configured to respond to descriptor-based result overruns. These types of overruns occur when the amount of the amount of results transferred exceeds the RESULT LENGTH field of the Results descriptor (or the sum total of bytes if multiple destination data descriptors are used).

A results overrun condition occurs when the results generated by the 7751 exceed the Results Length field in the results descriptor. When a result overrun occurs, the OVER bit in the Result Ring field of the DMA Status and Control register is set. Additionally, the OVER bit in the Result descriptor is set when the 7751 writes to this descriptor.

Bit 17 in the DMA Interrupt Enable register is defined as the OVER INTERRUPT ENABLE. This bit may be used to generate error detection when a result overrun occurs.

2.7 Descriptor-Based Overrun Effect On Descriptor Chains

When a descriptor based data overrun (or results overrun) occurs, the DMA engine discards any extra data (or results) for the current command. It then writes a one to the OVER bit and a zero VALID bit in the corresponding descriptor, and thereafter it starts placing the data (or results) for the next operation in the next valid descriptor.

2.8 Context RAM

The 7751 utilizes external RAM to maintain context information for each half-duplex channel of data. Either SRAM or EDO DRAM may be used for this context storage. At least 12ns SRAM or 60ns EDO DRAM are needed for peak performance. The PCLK speed may be adjusted to accommodate slower RAM speeds, but this will affect encoding and decoding performance.

When using SRAM, the compression RAM address signals are tied directly to the address bus of the compression RAM. The 7751 supports a maximum of 2 Mbytes of SRAM.

If DRAM is used, it must be EDO DRAM. When using DRAM, CADDR11-CADDR0 are attached to the address bus of the Context RAM. CADDR12 attaches to the RAS# signal. CADDR13 attaches to the LCAS# signal. CADDR14 attaches to the UCAS#

signal. CADDR15 attaches to the WE# signal. CADDR16 becomes the OE# signal. The 7751 supports a maximum of 32 Mbytes of EDO DRAM.

The 7751 encoding and decoding performance is affected by the setting of the DRAM bit due to the memory speed differences. Figure 5 shows the difference in 7751 performance with SRAM and DRAM.

Context RAM	Encode (Mbps)	Decode (Mbps)
12ns SRAM	64	120
60ns EDO DRAM	20	40

Figure 5. 7751 Context RAM performance

Compression uses significantly more RAM than if encryption is used alone. If desired, a single compression context may be used with a variable number of smaller encryption contexts.

When using LZS compression, memory is configured in such a way as to support a single full-duplex operation in a single 16 Kbyte block of memory. This includes support for two independent encryption contexts (encode and decode). For example, both an encode and decode operation may use the same session number. When using MPPC compression, each half-duplex context requires a full 16 Kbytes of memory, requiring 32Kbytes for full-duplex. That is, each encode and decode operation require a separate session number.

If only a single compression context is used, each additional half-duplex encryption context requires only 128 or 512 bytes depending upon the encryption algorithm used. See the Processing Unit Configuration register description for more details.

2.9 Clock Interface

There are three clock domains in the 7751: PCI_CLK, PCLK and ECLK. The PCLK and PCI_CLK signals are external inputs, and ECLK is an internal signal generated from one of the two input clock signals. The relationship between these three clocks is shown in Figure 6.

PCLK is an input which may be used to drive the internal ECLK signal. If the PCLK signal is not used to drive ECLK, it should be tied high or low. The maximum input frequency for the PCLK signal is 66MHz if the PLL is disabled, and 33MHz if the PLL is enabled. See Figure 7 for a summary of how the PLL, PCLK, and ECLK are related.

PCI_CLK is an input which is used to drive the PCI bus interface. The PCI_CLK signal may also be used to drive the PLL and ECLK , as determined by the CLKSEL signal. The maximum input frequency for the PCI_CLK signal is 33MHz. For the clock signal driving the PLL (if enabled) the minimum input frequency is 20MHz, and the maximum is 33MHz. If the PLL is disabled, the minimum clock speed will be determined by the compression RAM requirements.

ECLK is an internal signal which drives the clocks for all processing units, as well as the compression ram interface. The ECLK signal is driven by either PCLK or PCI_CLK, as determined by the CLKSEL signal. ECLK frequency must not surpass 33MHz. See Figure 6 and Figure 7 for a summary of how PCI_CLK, PCLK and ECLK are related.

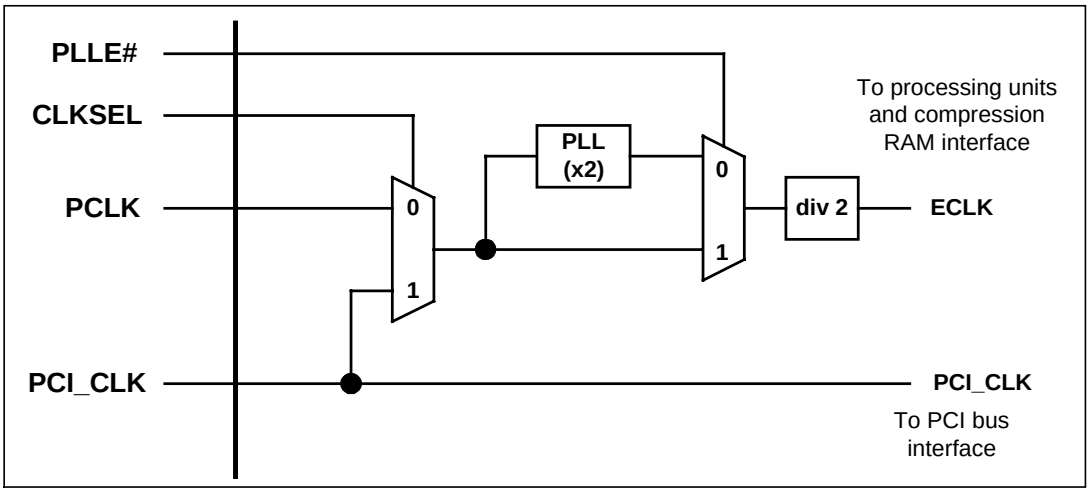


Figure 6. Clock Relationships

PLL	PCLK	ECLK
Disabled	0-66MHz	PCLK/2 (0-33MHz)
Enabled	20-33MHz	PCLK (20-33)

Figure 7. PLL/PCLK/ECLK relationship

2.10 PLL

The PLL is used to double the input value of either the PCLK or the PCI_CLK signal. This is useful if a 33MHz clock signal is readily available, and peak 7751 performance is desired.

2.10.1 Disabled

When the PLL is disabled, the ECLK frequency will be one-half the frequency of the clock that is driving the 7751. Disabling the PLL is done by connecting the PLL pins as described in Figure 8. If the PLL is disabled, the 7751 external component pins may be either no connects, or may be connected to the external PLL components.

Pin	Connection
LF	No Connect/PLL
RO	No Connect/PLL
AGS	No Connect/PLL
PLLE#	Vdd
AVdd	Vdd
AGND	GND

Note: "PLL" means the pin may be connected to the external 7751 PLL components.

Figure 8. PLL disable

2.10.2 Enabled

If the 7751 PLL is used, several external components are required. The schematic of these external components is shown in Figure 9, with the appropriate component values shown in Figure 10. Actual resistor values (including tolerance) must be within 25% of the value listed in Figure 10. The actual capacitor value, C1, must be within 20% of the value listed in Figure 10. Figure 10 shows typical values for C2 and C3, which are used as by-pass capacitors.

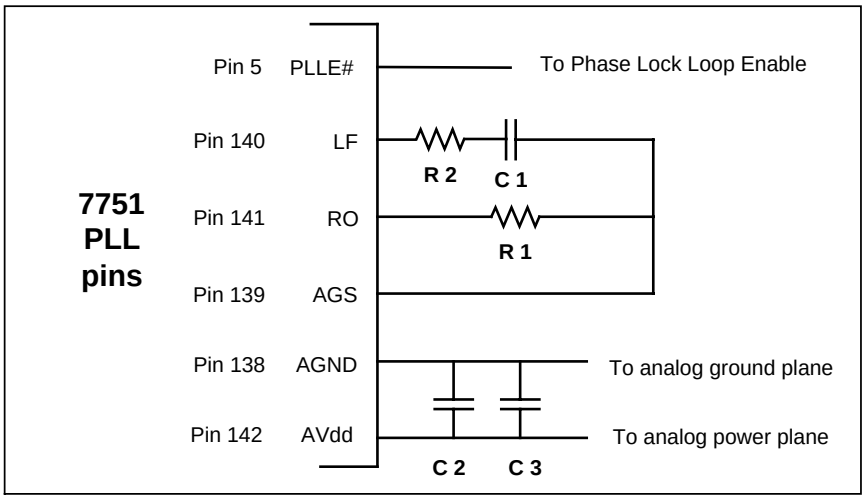


Figure 9. PLL pin connection

Discrete	Frequency	Value	Symbol
R1	33MHz	3	KΩ
R1	25 MHz	18	KΩ
R1	20MHz	45	KΩ
R2	any	0	KΩ
C1	any	1,500	PF
C2	any	0.1	µF
C3	any	10	µF

Figure 10. PLL external components

If the PLL is enabled, the input range for the signal driving ECLK (which will be PCLK or PCI_CLOCK, depending on the CLKSEL signal) must be between 20MHz and 33MHz. 7751 performance will scale with the value of ECLK.

2.11 Endianness

The BIG ENDIAN bit in the Processing Unit Configuration register affects the Command and Result Structures differently than it affects the Context, Source Data, and Dest Data Structures. The Command and Result Structures are affected by the BIG ENDIAN bit at the word (16-bit) level. The Source Context, Source Data, and Dest Data Structures are affected by the BIG ENDIAN bit at the byte (8-bit) level.

2.11.1 Command and Result Structures

The 7751 BIG ENDIAN bit operates on the Command and Result Structures in a 16-bit (word) level. The BIG ENDIAN bit affects the Command and Result Structures in the following way: Note that bits within the 16-bit words are not affected by the BIG ENDIAN bit.

Big Endian
In big endian Mode, bits 31-16 will be treated as the first two-byte word of the Command or Result Structure. That is, bits 31-16 will be the first sent to the Command, and the first produced by the Result. Bits 15-0 will represent the second two-byte word sent to the Command and the second produced by the Result. In big endian Mode, word "A" in Figure 11 represents the first word of the Command/Result Structure, and word "B" represents the second word of the Structure.

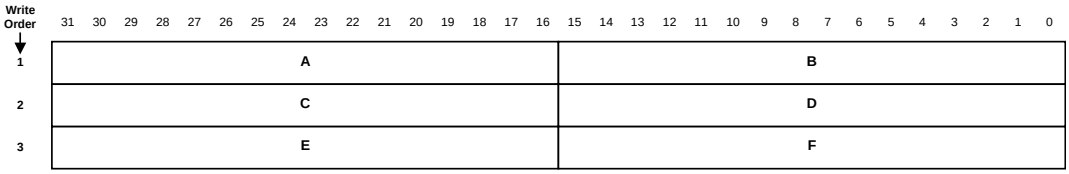


Figure 11. Command/Result Structure Words

Little Endian
In little endian mode, bits 15-0 will be treated as the first two-byte word, and bits 31-16 will be treated as the second two-byte word. In little endian Mode, word "B" in Figure 11 represents the first word of the Command /Result Structure, and word "A" represents the second word of the Structure.

2.11.2 Source Context, Source Data, and Dest Data

The 7751 big endian bit operates on the Source Context, Source Data, and Dest Data Structures in an 8-bit (byte) level. Note that the first byte represents the most significant byte of a key or IV.

Big Endian
When configured for 32-bit big endian Mode, the byte residing on bits 31-24 (byte A in Figure 12) will enter the engine as the first byte, and the byte residing on bits 7-0 (byte D in Figure 12) will enter the engine as the fourth byte.. In 16-bit big endian Mode, the byte residing on bits 15-8 (byte C in Figure 12) will enter the engine first, and the byte residing on bits 7-0 (byte D in Figure 12) will enter the engine second.

Little Endian
When configured for little endian Mode, the byte residing on bits 7-0 (byte D in Figure 12) will enter the engine as the first byte, and the byte residing on bits 31-24 (byte A in Figure 12) will enter the engine as the fourth byte. In 16-bit little endian Mode, the byte residing on bits 7-0 (byte D in Figure 12) will enter the engine as the first byte, and the byte residing on bits 15-8 (byte C in Figure 12) will enter the engine as the second byte..

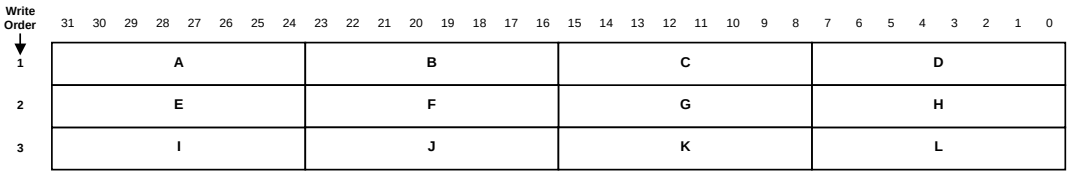


Figure 12. Source Context/Source Data/Dest Data Structure bytes

3 7751 Engine Performance

Figure 13 summarizes the measured performance of the individual 7751 engines. Performance of 7751 when multiple engines are used (e.g. the compression, MAC, and encryption engines are all engaged) can be approximated by considering the 7751 throughput of the slowest engine. The MAC and encryption engine speeds are accelerated (effectively multipled) by the actual compression ratio achieved by the compression engine. For example, if the achieved compression ratio is 2:1, then the MAC and encryption engine speeds are effectively doubled, and the compression engine would be the slowest engine. All engine speeds are given in Mbps. These performance numbers were obtained by measuring 1500 byte packet processing through the individual engines.

	Speed (Mbps)	
Engine	**Encode**	**Decode**
LZS	64	115
MPPC	53	103
SHA HMAC	80	80
MD5 HMAC	96	96
DES	164	164
3-DES	83	83
RC4	122	122

Figure 13. Individual Engine Performance

Figure 14 shows the measured 7751 packet processing performance for IPSec and PPTP.

		Speed (Mbps)	
Protocol	**Engine**	**Encode**	**Decode**
IPSec	LZS/SHA HMAC/DES	62	110
	LZS/SHA HMAC/3DES	62	110
PPTP	MPPC/RC4	51	101

Figure 14. Packet processing speed

4 Application Overview

The 7751 must be initialized properly before normal operation can begin. The following describes initializing the 7751.

4.1 PCI Initialization Overview

The overall flow of the initialization process is as follows:

1) Configure the following items in the PCI configuration space:
   Read Base Address Register 0 (points to Processing Unit registers)
   Read Base Address Register 1 (points to DMA registers)
   Set Master Latency Timer
   Set RETRY TIMEOUT and TRDY TIMEOUT fields to zero
   Write 0xFFFF to PCI Status Register to clear any outstanding error conditions
   Set Memory Access Enable (bit 1) and Bus Master Enable (bit 2) in PCI command Register to one
   Set Cacheline size to enable Memory Read Multiple commands (value depends on target memory controller)
2) Initialize the four DMA address registers.
3) Set the MSTRESET# and DMARESET# bits in the DMA Configuration register to zero and then to one. Ensure that the POLLING FREQUENCY and INVALID POLL SCALAR fields in the DMA Configuration register are set to zero.
4) Perform Unlock operation (see AN-0004-00).
5) Write a one to the RESET bit in the Processing Unit Control register, and poll for it to return a zero.
6) Set the following bits in the Processing Unit Configuration register to:
   select RAM type (also DRAM size and refresh rate, if DRAM),
   configure Endianness,
   set width of the data bus to 32 bits,
   Set the COMPRESSION CONFIGURATION and ENCRYPTION CONFIGURATION bits in the Processing Unit Configuration register (to configure context RAM).
7) Write 0x0400 to the FIFO Configuration register.
8) Invalidate all descriptors in all four descriptor rings.
9) Set the four Ring Control fields of the DMA Status and Control register to 0b01 (‘Enable’).
10) Clear all interrupts by writing a 0xFFFF to the Interrupt Status register.
11) Enable the desired interrupts by writing to the Interrupt Enable register.
12) The host sets the polling frequency in the DMA Configuration register to start initiator cycles.

4.2 DMA Register Setup

There are seven registers pointed to by BAR1. The first four registers are the Ring Address Registers for each of the four types of DMA descriptors: command, source data, destination data, and result. The host CPU, which sets up the descriptor tables in system memory, must place the starting memory address of each of the descriptors into the respective 7751 Ring Address register. A complete description of the descriptor formats can be found in Section 13.

Once the Ring Address registers have been configured, the descriptor rings must be enabled. This is accomplished by setting the appropriate bits in the DMA Status and Control Register. Any interrupts may be enabled at this time by configuring the DMA Interrupt Enable Register. The polling frequency and Invalid Scalar values should also be written to the DMA Configuration Register at this time.

4.3 Processing Unit Register Setup

The Processing Unit registers should be setup as desired by the user. Reserved bits must be written as zeros and ignored when read. Bits labeled "1" must be written as 1 and ignored when read.

See the processing unit register descriptions for details on how these registers should be set.

5 PCI Configuration Space

This block provides the first 66 bytes of Type 0, version 2.1, Configuration Space Header (CSH) to support software-driven “Plug-and-Play” initialization and configuration. This includes Command, Status, and two Base Address Registers (BAR0 and BAR1). These two BARs are used to configure memory-mapped address spaces for the 7751. Each BAR sets the base address for the register seet. The addressable range required by the 7751 is 4K for each BAR. Only BAR0 and BAR1 are used by the 7751. Address 0x00000000 is not a valid BAR address, as specified in PCI v2.1.

BAR0 is used for the Processing Unit registers, and BAR1 is used for the DMA registers.

Figure 15 shows the PCI commands supported by the 7751. The system writes the number of bus transfers its memory controller supports into the cacheline size register in the PCI configuration space. The 7751 supports Read Multiple PCI bus commands by using the Cache Line size PCI configuration register setting as a threshold to switch between Read and Read-multiple bus operations. Read-Multiple may allow up to 64 byte bursts for system controllers that only support smaller burst sizes. A cacheline size of 0x00 inhibits 7751 generation of Read-Multiple PCI bus commands.

CBE[3:0]	Command Types	PCI Master	PCI Slave
0110	Memory Read	Yes	Yes
0111	Memory Write	Yes	Yes
1010	Configuration Read	No	Yes
1011	Configuration Write	No	Yes
1100	Memory Read Multiple	Yes	Yes
1110	Memory Read Line	Yes	Yes

Figure 15. Command Types supported

Figure 16 gives a summary of the 7751 PCI Configuration Space. Figure 17 shows the default values for the PCI Configuration Space fields, as well as which of the fields may be optionally programmed at start up by an external EEPROM.

Figure 18 summarizes the PCI Configuration Space Command field, and Figure 19 summarizes the PCI Space Status field.

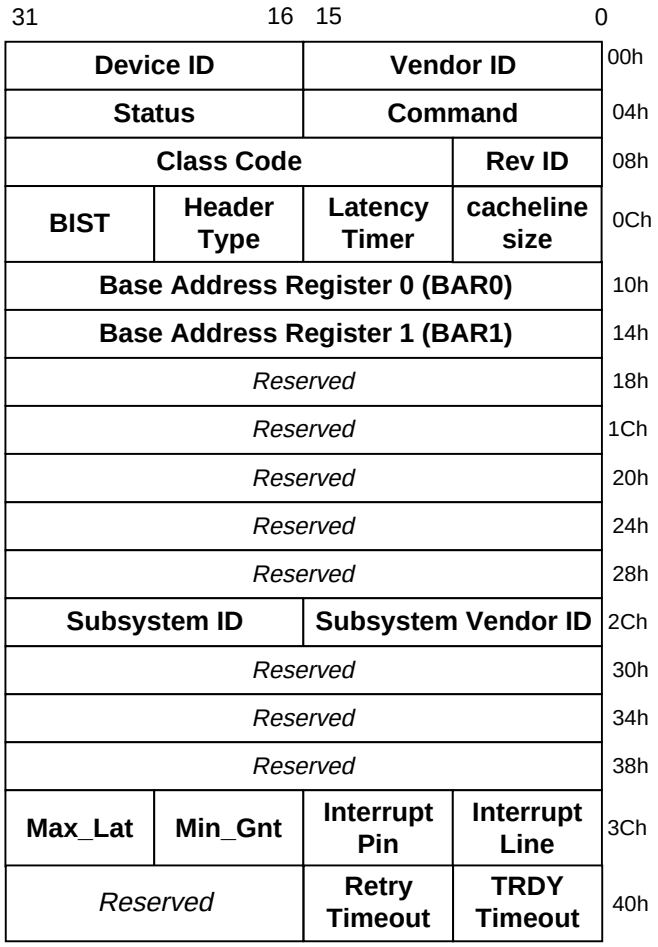


Figure 16. PCI Configuration Space

Note that the addresses for the PCI Configuration Space and the EEPROM Space are different.

PCI Configuration Space Field	Default Value (Hex)	Read/Write Access	PCI Configuration Address (Hex)	EEPROM Loadable
Device ID	0005	Read	02-03	Yes
Vendor ID	13A3	Read	00-01	Yes
Status	0280	Read	06-07	No
Command	0000	Read / Write	04-05	No
Class Code	0B4000	Read	09-0B	Yes
Revision ID	01	Read	08	Yes
BIST	0	Read	0F	Yes
Header Type	00	Read	0E	Yes
Master Lat Timer	00	Read / Write	0D	No
Cacheline Size	00	Read / Write	0C	No
BAR0	00000000	Read / Write	10-13	No
BAR1	00000000	Read / Write	14-17	No
Subsystem ID	0000	Read	2E-2F	Yes
Subsys Vendor ID	0000	Read	2C-2D	Yes
Max_Lat	00	Read	3F	Yes
Min_Gnt	00	Read	3E	Yes
Interrupt Pin	01	Read	3D	Yes
Interrupt Line	00	Read / Write	3C	No
Retry Timeout	80	Read / Write	41	No
TRDY Timeout	80	Read / Write	40	No

Figure 17. Default PCI Configuration Space Field Values

Note: the RETRY TIMEOUT field and the TRDY TIMEOUT field must be set to zero upon initialization and after a reset.

Bit	Description	Default	Type
15:10	RESERVED	0	RO
9	Fast Back-to-Back Mstr Enable	0	CMND
8	System Error Enable	0	CMND
7	RESERVED	0	RO
6	Parity Error Enable	0	CMND
5	RESERVED	0	RO
4	MWINV	0	CMND
3	RESERVED	0	RO
2	Bus Master Enable	0	CMND
1	Memory Access Enable	0	CMND
0	I/O Access Enable	0	CMND

RO = Read Only

Figure 18. PCI Command Register

Bit	Description	Default	Type
15	Detect Parity Error	0	Status
14	Signaled System Error	0	Status
13	Received Master Abort Status	0	Status
12	Received Target Abort Status	0	Status
11	Signaled Target Abort Status	0	Status
10:9	Device Select Timing	01	RO
8	Data Parity Detected	0	Status
7	Fast Back-to-Back Capable Status Flag	1	RO
6	RESERVED	0	RO
5	66MHz-Capable Status Flag	0	Status
4:0	RESERVED	0	RO

RO = Read Only

Figure 19. PCI Status Register

5.1.1 Use of EEPROM

Some of the fields in the PCI Configuration Space may be configured to load automatically from an external 64x16 EEPROM. If the EEPROM# signal is tied high, these fields will load on power-up to their default values. If the EEPROM# signal is tied low, some fields in the PCI Configuration Space will be loaded on power-up from the external EEPROM. Figure 17 summarizes which PCI configuration registers will be loaded by EEPROM, as well as the default PCI configuration values if EEPROM is not used. Figure 20 shows which EEPROM addresses are utilized to set the PCI configuration values.

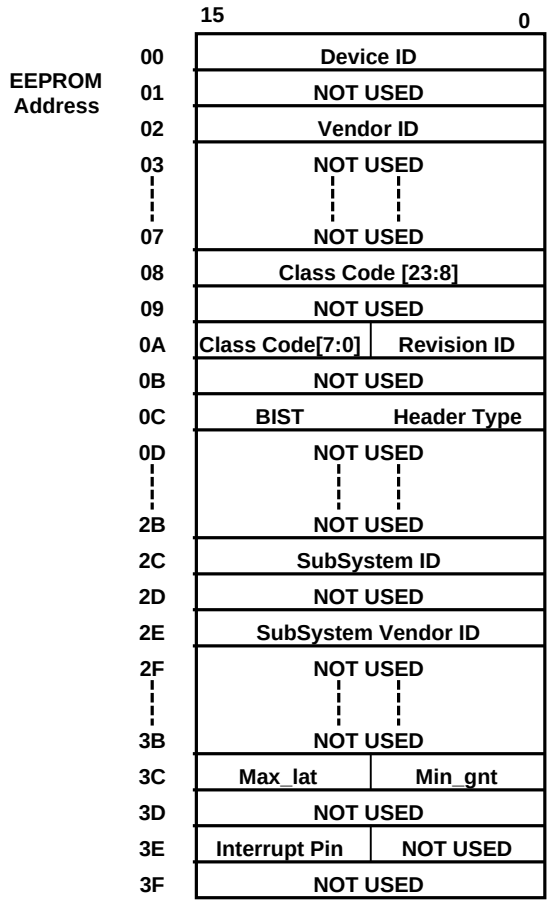


Figure 20. EEPROM Addressing

5.1.2 Enabling Encryption

The 7751 powers on with encryption disabled. In order to enable encryption capabilities, an unlock software mechanism must be followed, utilizing a key that is both programmed into the EEPROM and resident in the software mechanism. Thus the serial EEPROM is required to enable the encryption capabilities of the 7751.

There are 3 security levels that can be enabled via this unlock procedure. Figure 21 summarizes these security levels, along with the different 7751 Chip IDs and algorithms supported. Details of this unlock software mechanism are available from Hi/fn.

Security Level	7751 Chip ID	Algorithm Support
Security Level 0 (default)	0x30XX	LZS, MPPC
Security Level 1	0x10XX	LZS, MPPC, MD5, SHA, DES
Security Level 2	0x11XX	LZS, MPPC, MD5, SHA, DES, 3-DES, RC4

Figure 21. 7751 Security Levels

6 Memory Overview

The relationship between the Ring Address registers, the descriptor formats, and the data blocks is shown in Figure 22.

There are four Ring Address registers in the 7751: command, source data, destination data, and result. The Ring Address registers are programmed from the system with the address of the appropriate descriptors. The descriptors, in turn, point to locations in memory where the data blocks are located. The data blocks will contain commands, source data, destination data, or results as defined by the specific descriptor. Note that multiple descriptors may be joined to form descriptor rings.

For the destination data and result descriptors, the least two significant bits in the Length and Pointer values must be set to zero for 4-byte alignment. This is not necessary for the command and source data descriptors.

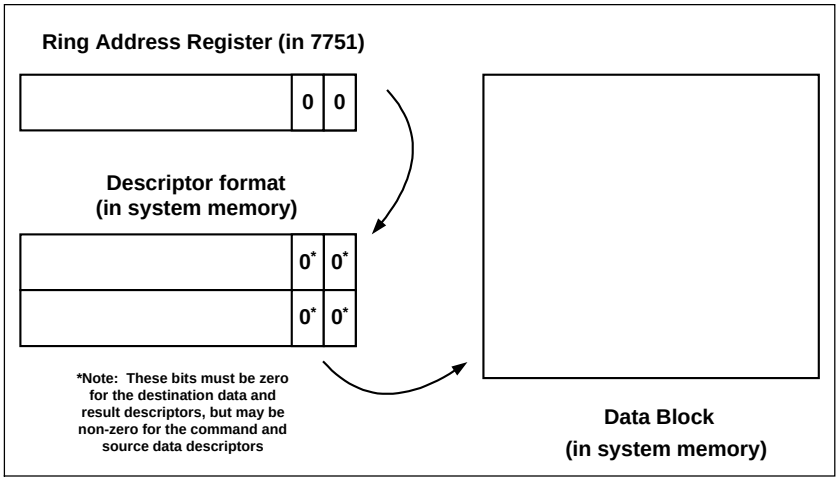


Figure 22. Memory overview

7 7751 Pin Connections

Pin Number(s)	Signal Name	Type	Description
PCI Interface			
20-21, 23-25, 27-29, 33, 35-39, 41-42, 59-60, 62-63, 65-68, 71-73, 75-77, 79-80	AD[31:0]	I/O	**PCI Address/Data Bus:** 32-bit multiplexed PCI address and data lines. During the first clock of the transaction, ADDR[31:0] contains a physical address (32 bits). During subsequent clock cycles, ADDR[31:0] contains data.
8	PCI_CLK	I	**PCI Clock:** This signal is the PCI clock which comes from the PCI bus. The PCI clock range is 33MHz or less.
17	GNT#	I	**Bus Grant:** Indicates to the 7751 that access to the PCI bus is granted.
45	FRAME#	I/O	**Frame:** Asserted by the 7751 to indicate the beginning and duration of a master transaction. While FRAME# is asserted, data transfer continues. FRAME# deasserts to indicate that the next data phase is the final data phase transaction. FRAME# is monitored by the 7751 when it acts as a target.
46	IRDY#	I/O	**Initiator Ready:** Indicates the bus master's ability to complete the current data phase of the transaction. A data phase is completed on any clock when both TRDY# and IRDY# are asserted. Wait cycles are inserted until TRDY# and IRDY# are asserted together.
47	TRDY#	I/O	**Target Ready:** Indicates the target agent's ability to complete the current data phase of the transaction. A data phase is completed on any clock when both TRDY# and IRDY# are asserted. Wait cycles are inserted until TRDY# and IRDY# are asserted together.
49	DEVSEL#	I/O	**Device Select:** Asserted by the target of the current access. When the 7751 is a bus master, it expects the target to assert DEVSEL# within 5 bus cycles, confirming the access. If the target does not assert DEVSEL# within the required bus cycles, the 7751 aborts the cycle. As a target, when the 7751 recognizes its transaction, it asserts DEVSEL# two cycles after the assertion of FRAME#.
50	STOP#	I/O	**Stop:** Indicates that the current target is requesting the bus master to stop the current transaction. As a master, the 7751 responds to the assertion of STOP# by disconnecting, retrying or aborting. As a target, the 7751 asserts STOP# to retry or disconnect.
53	PERR#	I/O	**Parity Error:** Asserted when a data parity error is detected.
54	SERR#	O	**System Error:** Asserted when a serious system error (not necessarily a PCI error) is detected. The 7751 asserts the SERR# two cycles after the failing address. This output is an open drain driver.
55	PAR	I/O	**Parity:** Calculated by the 7751 as an even parity bit for the ADDR[31:0] and CBE#[3:0] lines.

19	REQ#	O	**Bus Request:** Asserted by the 7751 to indicate to the PCI bus arbiter that it requires use of the PCI bus.
16	RESET#	I	**PCI Reset:** Resets the 7751 to its initial state. This signal must be asserted for at least 10 PCI_CLK clock cycles. When in the reset state, all PCI output pins are put into tri-state and all open drain signals are floated.
15	INT#	O	**Interrupt Request:** Asserted by the 7751 when one of the interrupt sources is asserted. The output is an open drain driver.
31, 43, 58, 70	CBE#[3:0]	I/O	**PCI Bus Command/Byte Enable:** During the address phase of the transaction, CBE#[3:0] provide the PCI bus command. During the data phase, these signals provide the byte enables.
32	IDSEL	I	**Initialization Device Request:** Asserted by the system host to act as a chip select during PCI configuration read and write transactions.
EEPROM Interface			
6	EEPROM#	I	**EEPROM:** If this signal is low, the PCI Configuration Space will be loaded from EEPROM. If this signal is high, the PCI Configuration Space will contain the default values. For additional details, see sections 5.1.1 and 5.1.2.
10	EEPROM_DO	O4	**EEPROM_DO:** Data Out Signal for the EEPROM. Connects to the Data In signal of the EEPROM. This pin is always active (i.e. it is never tri-stated).
11	EEPROM_CS	O4	**EEPROM_CS:** Chip Select Signal for the EEPROM. Connects to the Chip Select signal of the EEPROM. This pin is always active (i.e. it is never tri-stated).
12	EEPROM_SK	O4	**EEPROM_SK:** Serial Clock Connection signal for the EEPROM. Connects to the serial clock signal of the EEPROM. This pin is always active (i.e. it is never tri-stated).
13	EEPROM_DI	I	**EEPROM_DI:** Data In Signal for the EEPROM. Connects to the Data Out signal of the EEPROM.
Context RAM Interface			
103, 101-99, 97-95, 93-91, 88-86, 84-82	CDATA 15-0	I/O8	**Context RAM Data:** 16-bit bi-directional data bus for the Context RAM. 16-bit Context RAM must be used.
129-128, 126-124, 122-121, 119-117, 115-112, 110-108, 106-104	CADDR 19-0	O8	**Context RAM Address:** These output signals select the address of the Context RAM. Either SRAM or DRAM may be used. See the Context RAM section (section 2.8)for details on how to connect these signals to the Context RAM. (Note: This bus addresses 16-bit words)
130	WE#	O8	**Write Enable:** Active low output. If SRAM is used, this signal will become active during each write access to the Context RAM. If DRAM is used, this signal must be left unconnected.

132	OE#	O8	**Output Enable:** Active low output. If SRAM is used, this signal will become active during each read access to the Context RAM. If DRAM is used, this signal must be left unconnected.
134	LB#	O4	**Lower Byte Enable:** Active low output. If SRAM is used, this signal will become active during each access to the lower byte of the Context RAM. If DRAM is used, this signal must be left unconnected.
133	UB#	O4	**Upper Byte Enable:** Active low output. If SRAM is used, this signal will become active during each access to the upper byte of the Context RAM. If DRAM is used, this signal must be left unconnected.
PLL Interface			
140	LF	A	**Loop Filter:** Analog pin that connects to the external PLL filter. See the PLL section for additional information on the PLL and the external components required.
141	RO	A	**Resistor Output:** Analog pin that sets the VCO center frequency. See the PLL Section for additional information on setting the center frequency.
139	AGS	A	**Analog Ground Sense:** Analog pin that connects to the external PLL filter. See the PLL Section for additional information on the PLL.
5	PLLE#	I	**PLL Enable:** Active low input. If this signal is high (inactive), the PLL will be disabled. The PCLK (or PCI_CLK) signal will drive the internal logic directly. If this signal is low (active), the PLL will be enabled. The PCLK (or PCI_CLK) frequency will be doubled before driving the internal logic.
142	AVCC		**Analog 3.3V:** Analog 3.3V connection.
138	AGND		**Analog Ground:** Analog ground connection.
Miscellaneous Signals			
3	PCLK	I	**PCLK:** Clock input for processing unit clock.
2	CLKSEL	I	**CLKSEL:** Selects whether the PCLK or PCI_CLK will drive ECLK. If this signal is set to 0, PCLK will drive ECLK, and if this signal is set to 1, PCI_CLK will drive ECLK.
1	TEST (GND)		**TEST(GND):** Test signal. Tie to GND for normal operation.
9, 18, 26, 40, 48, 51, 57, 61, 74, 81, 90, 94, 102, 111, 120, 127, 135, 143, 144	VCC		+3.3 Volts
4, 7, 14, 22, 30, 34, 44, 52, 56, 64, 69, 78, 85, 89, 98, 107, 116, 123, 131, 136, 137	GND		Ground

I=input, O=output; I/O=bidirectional, A=analog
For output and I/O pins, the number following the O represents the output drive capability. For example, O4 implies a 4mA output drive capability, O8 implies an 8mA output drive capability, etc.

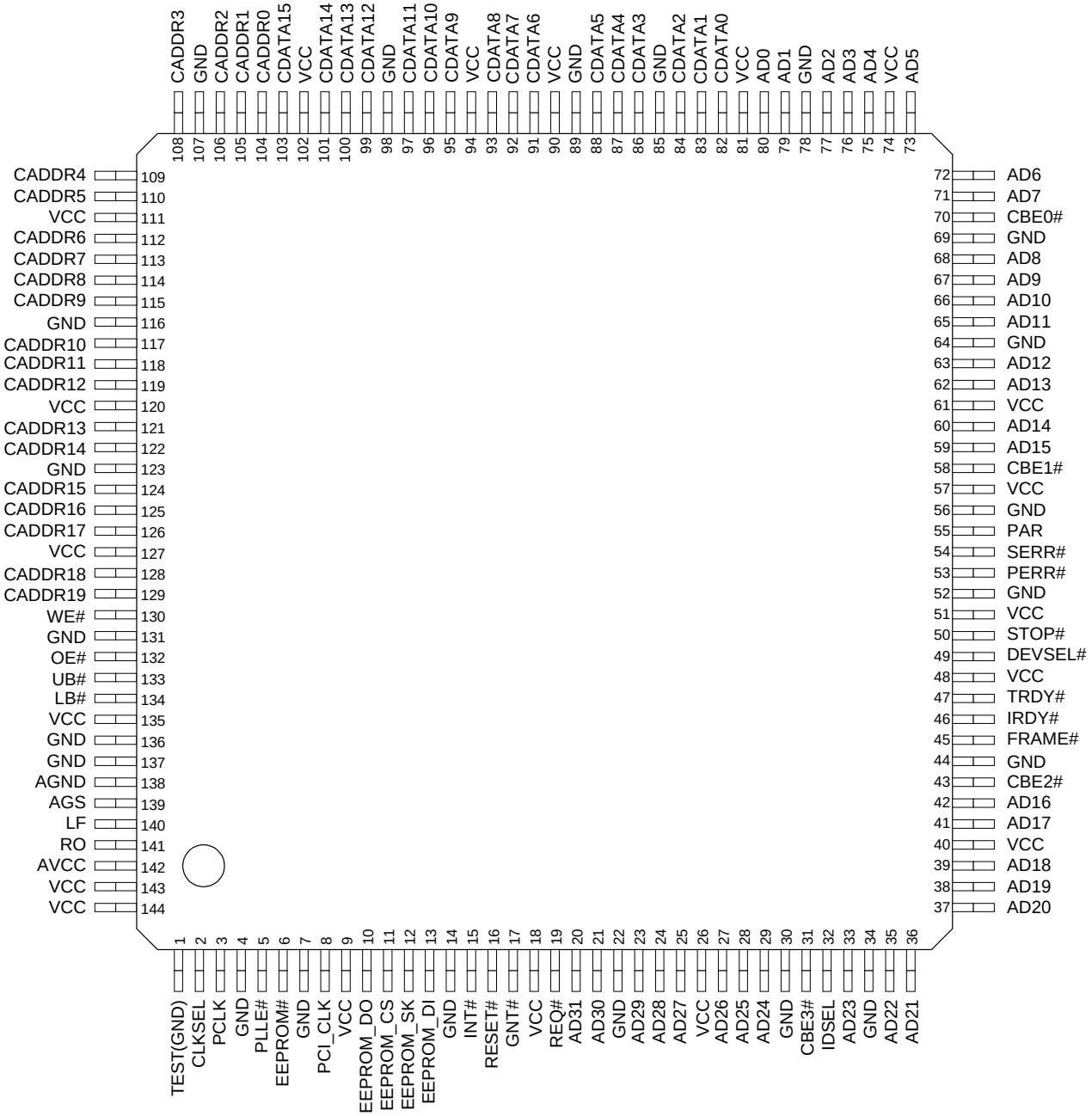


Figure 23. 7751 144-pin TQFP pinout

8 Register Overview

There are two sets of registers in the 7751. The DMA interface registers are used to configure and control DMA transfers. The processing unit registers are used to configure and control the internal processing engines.

The DMA interface registers are accessed at memory location BASE REG1 + offset. The valid offsets for DMA interface register accesses are defined below. All other offsets are reserved, and should not be accessed.

The processing unit registers are accessed at memory location BASE REG0 + offset. The valid offsets for processing unit register accesses are defined below. All other offsets are reserved, and should not be accessed.

8.1 DMA Interface Registers

Name	PCI Address
DMA Command Ring Address	BASE REG1 +0Ch
DMA Source Ring Address	BASE REG1 +1Ch
DMA Result Ring Address	BASE REG1 +2Ch
DMA Destination Ring Address	BASE REG1 +3Ch
DMA Status and Control	BASE REG1 +40h
DMA Interrupt Enable	BASE REG1 +44h
DMA Configuration	BASE REG1 +48h
Revision ID	BASE REG1 +98h

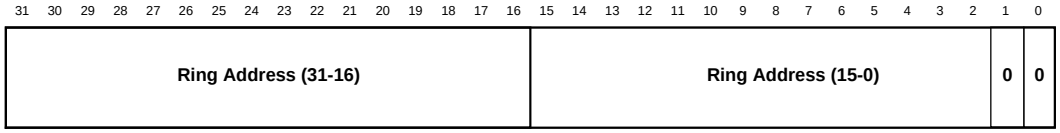
8.2 Processing Unit Registers

Name	PCI Address
Processing Unit Control	BASE REG0 +04h
Processing Unit Interrupt Status	BASE REG0 +08h
Processing Unit Configuration	BASE REG0 +0Ch
Processing Unit Interrupt Enable	BASE REG0 +10h
Processing Unit Status	BASE REG0 +14h
FIFO Status	BASE REG0 +18h
FIFO Configuration	BASE REG0 +1Ch

9 DMA Interface Registers

Reserved bits must be written as zeros and ignored when read. Bits labeled "1" must be written as 1 and ignored when read. Bits labeled "0" must be written as 0 and ignored when read.

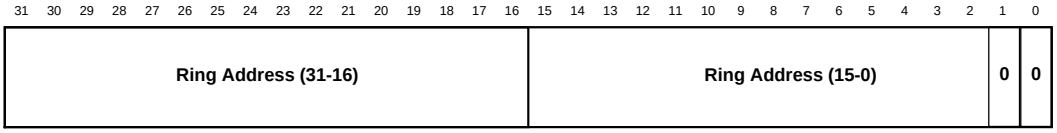
9.1 DMA Command Ring Address Register (31-0)



The Command Ring Address register is the address register for the command descriptor ring. This register must be initialized by the host to the address of the first descriptor of the command descriptor ring. During processing, this register will be updated to contain a pointer to the currently active command descriptor. Bits 0 and 1 of this address register must be set to 0 to allow for 4-byte alignment.

The POLLING FREQUENCY field in the DMA Configuration register must be set to zero before this register can be re-initialized.

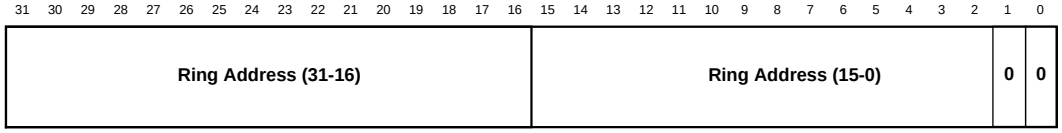
9.2 DMA Source Data Ring Address Register (31-0)



The Source Data Ring Address register is the address register for the source descriptor ring. This register must be initialized by the host to the address of the first descriptor of the source descriptor ring. During processing, this register will be updated to contain a pointer to the currently active source descriptor. Bits 0 and 1 of this address register must be set to 0 to allow for 4-byte alignment.

The POLLING FREQUENCY field in the DMA Configuration register must be set to zero before this register can be re-initialized.

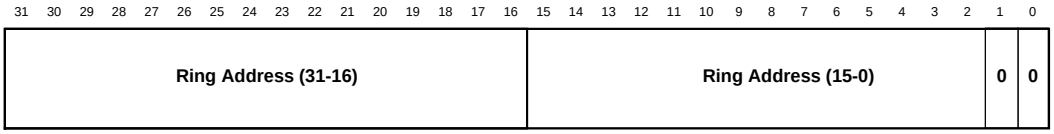
9.3 DMA Result Ring Address Register (31-0)



The Result Ring Address register is the address register for the result descriptor ring. This register must be initialized by the host to the address of the first descriptor of the result descriptor ring. During processing, this register will be updated to contain a pointer to the currently active result descriptor. Bits 0 and 1 of this address register must be set to 0 to allow for 4-byte alignment.

The POLLING FREQUENCY field in the DMA Configuration register must be set to zero before this register can be re-initialized.

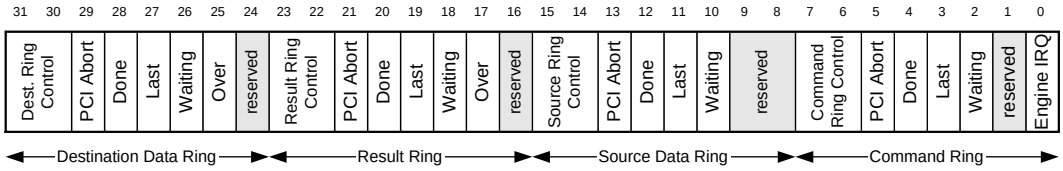
9.4 DMA Destination Data Ring Address Register (31-0)



The Destination Data Ring Address register is the address register for the destination descriptor ring. This register must be initialized by the host to the address of the first descriptor of the destination descriptor ring. During processing, this register will be updated to contain a pointer to the currently active destination descriptor. Bits 0 and 1 of this address register must be set to 0 to allow for 4-byte alignment.

The POLLING FREQUENCY field in the DMA Configuration register must be set to zero before this register can be re-initialized.

9.5 DMA Status and Control Register



The bits in the DMA Status and Control Register (other than the control fields) are set when the corresponding event occurs. They are cleared by writing a one to the bit position, or by a PCI Bus master reset.

Interrupt status bits are logically ANDed with the corresponding DMA Interrupt Enable Register bits to generate interrupts on desired events. Software may poll the status bits independent of the interrupt enable conditions. The individual status bits are defined as follows:

9.5.1 Engine IRQ

This bit is the interrupt directly from the Processing Unit Interrupt Status Register. See the Processing Unit Interrupt Status register for further details.

Note: The bit in the Processing Unit Interrupt Status register which caused the interrupt must be cleared **before** clearing this bit.

9.5.2 Waiting

This bit will be set to one if the next descriptor to be processed has its VALID bit set to zero.

9.5.3 Last

This bit will be set to one after transferring the data for a descriptor whose LAST bit was set.

9.5.4 Done

This bit will be set to one after transferring the data for any descriptor.

9.5.5 PCI Abort

This bit will be set to one if a Master-Abort, Target-Abort, System Error, or Parity Error occurred during a 7751 Initiator cycle. If these errors occur on any descriptor ring, it is considered a fatal error and will cause the 7751 to stop generating Initiator cycles on all descriptor rings until this bit is cleared. Also, the Ring Control fields for each descriptor are disabled (i.e. set to 0x01).

9.5.6 Ring Control

The Ring Control field controls the polling of descriptors. The frequency of polling is set in the DMA Configuration Register.

Control modes are applied independently to the Descriptor Rings. Modes are entered by writing one of the following values to the RING CONTROL field:

00 = NOP: This mode has no effect and can be used when writing to other fields in the register.

01 = Disable: After the current descriptor is processed, the descriptor ring will be disabled.

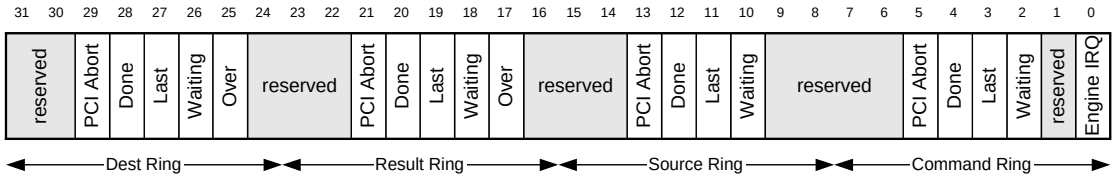
10 = Enable: After all registers and descriptor rings have been initialized by software, setting this mode will enable descriptor ring processing by the DMA engine.

11 = reserved: This mode is reserved.

9.5.7 Over

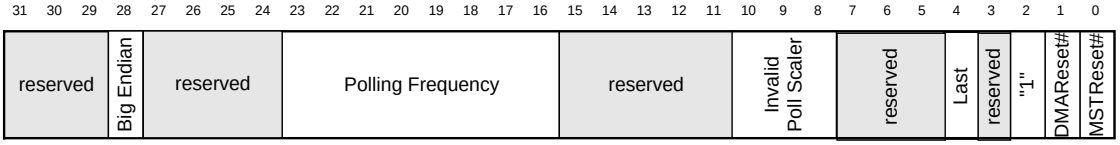
These bits signify that the 7751 produced more destination or result data than was allocated in the destination or result descriptors (when the CPU is in control of the descriptor LAST flag). This bit is set under the same conditions that set the Over bits in the Destination Data or Result descriptors. This bit is only significant if the Last bit (bit 4) in the DMA Configuration register was set to one. See sections 2.5, 2.6, and 2.7 for functional details on overrun conditions.

9.6 DMA Interrupt Enable Register



The DMA Interrupt Enable Register selects the events that will generate an interrupt to the IRQ# signal. The bits in the DMA Interrupt Enable register correspond to the associated bits in the DMA Status and Control Register. Enable bits which are set to one will enable the interrupt signal when the associated bit in the DMA Status and Control register becomes set. This register will be cleared when either the MSTRESET# or DMARESET# bit in the DMA Configuration register is set.

9.7 DMA Configuration Register



9.7.1 MSTReset#

Setting this bit to zero will reset the descriptor engines and the processing unit registers. Setting this bit is equivalent to setting the DMARESET# bit in this register AND the RESET bit in the Processing Unit Control register.

When setting the MSTRESET# bit from one to zero, the poll frequency must first be set to zero. Then the system must wait 1ms to ensure that queued DMA operations are not interrupted in the middle of their processing. Then the MSTRESET# bit may be set to zero.

Before setting the MSTRESET# bit from zero to one, the system must first poll for the RESET bit in the Processing Unit Control register to be set to one. Then the MSTRESET# bit may be set to one.

After the MSTRESET# bit has been set to a one and before writing to any other register, the system must first poll for the RESET bit in the Processing Unit Control register to be set to zero. Then the 7751 is ready for use.

9.7.2 DMAReset#

Setting this bit to zero will reset the DMA descriptor engines.

When setting the DMARESET# bit to zero, the poll frequency must first be set to zero. Then the system must wait 1ms to ensure that queued DMA operations are not interrupted in the middle of their processing. Then the DMARESET# bit may be set to zero.

9.7.3 Last

This bit sets whether the LAST bit in the Dest Data Descriptor and Result Descriptor is controlled by the system or the 7751. The system will always control the LAST bit in the Command Descriptor and Source Data Descriptor.

If this bit is set to zero, the 7751 will control the LAST bit in the Dest Data and Result Descriptors. The 7751 will set this bit when the current descriptor represents the last descriptor in a command.

If this bit is set to one, the host CPU will control the LAST bit in the Dest Data and Result Descriptors. The 7751 will read this bit to determine when a descriptor represents the last descriptor in a buffer. If the 7751 finishes a command but has not detected a LAST bit , it will search forward through the ring until it finds the LAST bit set in a descriptor.

9.7.4 Invalid Poll Scalar

This field selects the frequency with which the 7751 will poll an invalid descriptor, until it becomes valid. This field is useful to control the amount of PCI bus bandwidth used when there is no current DMA activity. Figure 24 shows the cycles between polls for different Invalid Poll Scalar values.

The Invalid Poll Scalar value is only used when the 7751 has already detected an invalid descriptor. However, if the polling frequency value is greater than the invalid poll scalar (polling frequency is slower than Invalid Poll Scalar frequency), the poll frequency will be used. When the polled descriptor is valid, only the polling frequency is used to set 7751 PCI bus requests.

InvalidPoll Scalar value	PCI_CLK Cycles Between polls of an invalid descriptor
000	256
001	512
010	768
011	1024
100	1280
101	1536
110	1792
111	2048

Figure 24. Invalid poll scalar values

9.7.5 Polling Frequency

This field sets the polling rate with which the 7751 will access the PCI bus. This field is useful to control the amount of PCI bus bandwidth used when there is current DMA activity. This setting is in multiples of 16 PCI clock cycles. A value of 00h disables the polling function and is used for diagnostic purposes only. The table below illustrates the maximum amount of PCI bandwidth (out of a possible 132 MB/s) the 7751 will be allowed to use. Figure 25 shows the bandwidth used for different Polling frequency values. Note that the values in Figure 25 are examples, and actual bandwidth may vary.

Avg. Burst Size	Polling Freq	Bandwidth Used
64	11111111	0.52 MB / s
64	10000000	1.04 MB / s
64	01000000	2.08 MB / s
64	00100000	4.17 MB / s
64	00010000	8.33 MB / s
64	00001000	16.7MB / s
64	00000100	33 MB / s
64	00000010	50 MB / s
64	00000001	50 MB / s
64	00000000	Disabled

Figure 25. Polling frequency values

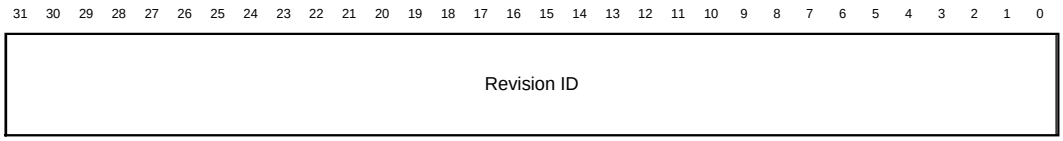
9.7.6 Big Endian

This bit selects the byte order of descriptors and data buffers read and written while 7751 performs initiator accesses. When this bit is set to one, the 7751 accesses memory in big endian byte order. When this bit is set to zero, the 7751 accesses memory in little endian byte order.

Target (register) accesses are not affected by the setting of this bit.

This bit should be used instead of the BIG ENDIAN bit in the Processing Unit Configuration register, when interfacing 7751 in a big endian memory system.

9.8 Revision ID Register

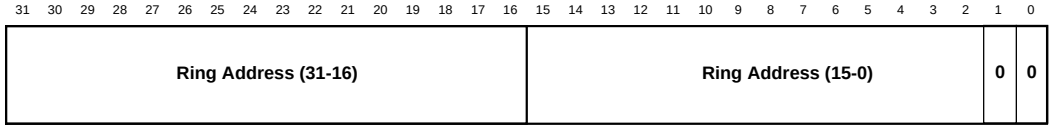


9.8.1 Revision ID

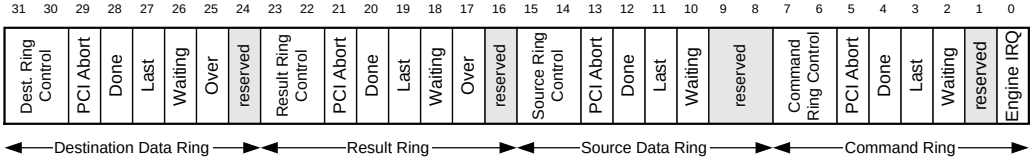
This register may only be read. This register will return the revision number for the 7751. The revision number is 0x00000001. The Chip ID (located in the processing unit register space) represents the identification number for the 7751 device. The Revision ID represents the actual revision number for the 7751.

10 DMA Register Summary

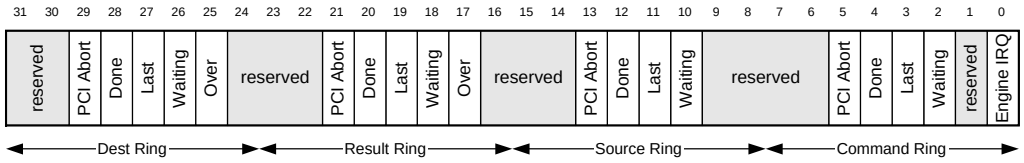
DMA Ring Address registers



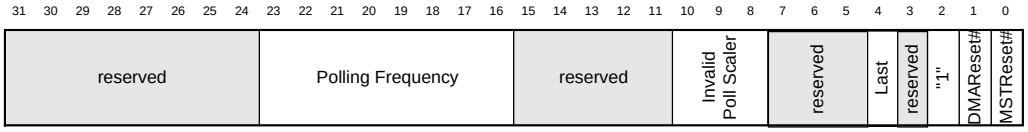
DMA Status and Control register



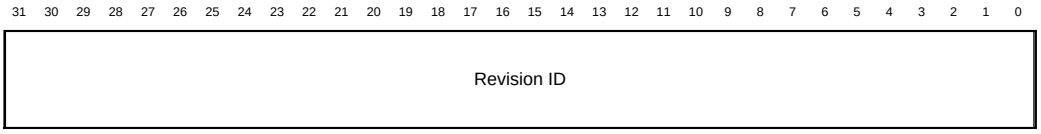
DMA Interrupt Enable register



DMA Configuration register



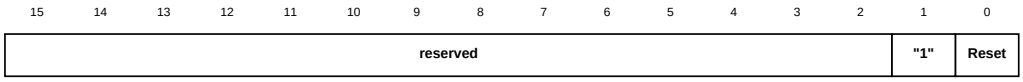
Revision ID register



11 Processing Unit Registers

Reserved bits must be written as zeros and ignored when read. Bits labeled "1" must be written as 1 and ignored when read. Bits labeled "0" must be written as 0 and ignored when read.

11.1 Processing Unit Control Register



This register may be read or written.

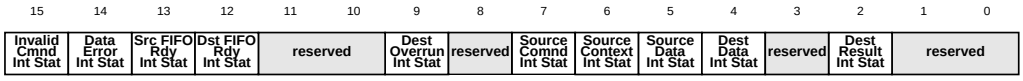
11.1.1 Reset

When this bit is set to one, all processing operations will immediately stop. The Source and Dest FIFOs will be cleared. Any intermediate information still residing in the processing units will be lost. If an operation is prematurely stopped in this manner, the current context will be corrupt and must be cleared next time it is used. This bit will not affect the Processing Unit Configuration register.

Do not access the 7751 (except to read the Processing Unit Control register) after the RESET bit is set to one until after the RESET bit returns a zero when read. Do not write a zero to this bit to clear the RESET bit. It will clear automatically.

After a hardware reset, this bit will return a one until the 7751 is ready for normal operation. After it returns to zero, all Processing Unit registers may be accessed normally.

11.2 Processing Unit Interrupt Status Register



This register may be read or written. It is used to monitor the status of the Interrupt sources. It is also used to clear the interrupts. The default value of all the fields in this register after a reset is zero.

Each bit in this register will be set to one when its corresponding bit in the Processing Unit Status register becomes a one. When the corresponding Processing Unit Status register bit returns to zero, the Interrupt Status bit will remain a one. To clear the Interrupt Status bit, a one must be written to the bit in the Processing Unit Interrupt Status register.

The relationship of the Processing Unit Status register bits, the Processing Unit Interrupt Status register bits and the Processing Unit Interrupt Enable register bits is shown in Figure 26.

Refer to the Processing Unit Status Register section for a description of all the bits in the Processing Unit Interrupt Status Register.

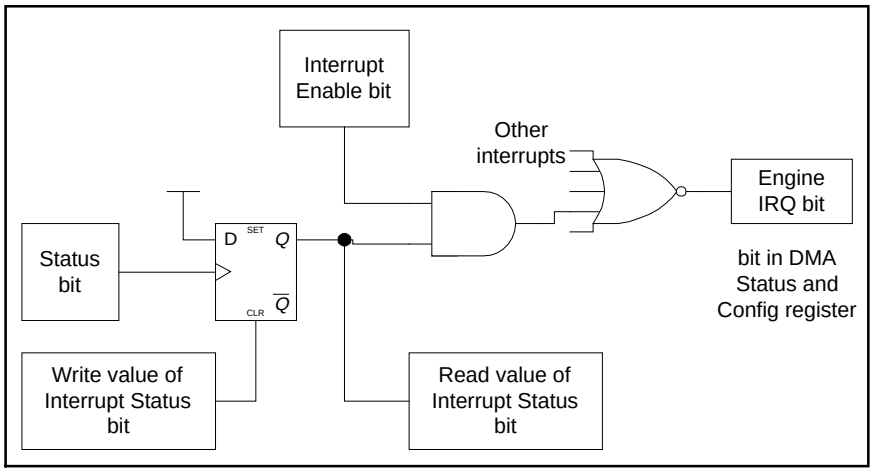


Figure 26. Interrupt Logic

11.3 Processing Unit Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRAM Size			DRAM Refresh Rate		reserved	"1"	"1"	Big Endian	"1"	Chip ID	DRAM	reserved	Comp Config	Encrypt Config	reserved

This register may be read or written and is used to configure chip options. The Processing Unit Configuration register cannot be written to unless the chip is idle; that is, the FIFO's are empty and there is no current operation. When read, it will contain the last value written. The default value of all the fields in this register after a hardware reset is zero. This register is not affected by a software reset (setting the RESET bit in the Processing Unit Control register to one)

11.3.1 DRAM Size

This field is significant only if EDO DRAM is used for Context RAM. It selects the EDO DRAM address configuration.

Value	Size (8-bit words)	Rows	Columns
0 0 0	512K	9	9
0 0 1	1M	10	9
0 1 0	2M	10	10
0 1 1	4M	11	10
1 0 0	8M	11	11
1 0 1	16M	12	11
1 1 0	32M	12	12
1 1 1	Reserved		

Figure 27. EDO DRAM size values

11.3.2 DRAM Refresh

This field is significant only if EDO DRAM is used for Context RAM. It selects the frequency of EDO DRAM refresh cycles. The refresh frequency is a divisor of the ECLK frequency, as set by the bits in this field.

Value	Divisor
0 0	512
0 1	256
1 0	128
1 1	Reserved

Figure 28. Refresh frequency

11.3.3 Big Endian

This bit selects the byte order of data transferred via the DMA interface and via the Data register.

If this bit is set to zero, this chip operates as if it were attached to a Little Endian processor. Bits 7-0 of the 7751 data bus will enter or leave the Processing Unit first.

If this bit is set to one, this chip operates as if it were attached to a Big Endian processor. Bits 31-24 of the 7751 data bus will enter or leave the Processing Unit first.

The big endian bit affects the data which enters the processing units. Commands, context, and internal registers are affected by the endianness of the 7751, as described in section 2.11.

11.3.4 ChipID

This bit allows the ChipID value to be read from the Processing Unit Status register. When this bit is set to one, the ChipID value can be read once from the Processing Unit Status register. After the ChipID value is read from the Processing Unit Status register, this bit returns to zero.

This bit is readable. When this bit is set to one, it returns the value one until after the ChipID value is read from the Processing Unit Status register. After the ChipID value is read from the Processing Unit Status register this bit is read as the value zero.

11.3.5 DRAM

This bit selects the RAM type used for the Context RAM. This bit must be set to zero if SRAM is used, and one if DRAM is used. Note that if DRAM is used, it must be EDO DRAM. Please refer to the Context RAM section for additional details on the Context RAM interface.

11.3.6 Compression Configuration

This bit determines whether single or multiple compression contexts are to be used.

If this bit is set to zero, multiple compression contexts may be used simultaneously. The encryption contexts will be interlaced within the compression context.

If this bit is set to one, only one compression context can be used, and will be stored at the beginning of the allocated memory. All the encryption contexts will be stored above the compression context.

Figure 29 shows local context memory maps for both multiple and single history configurations. When multiple history is used, the 128 or 512 byte MAC and encryption keys or context are embedded in unused memory within the 16K compression histories. The single history size will be 32Kbytes with either the LZS or MPPC algorithm.

The numbers to the right of both the multiple and single history memory maps (i.e. 0, 1, 2, etc) show the Session number associated with each history.

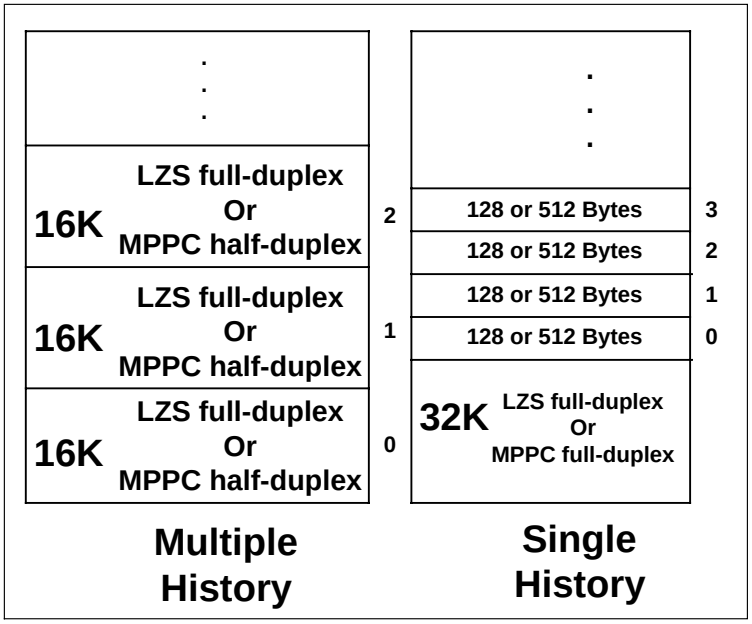


Figure 29. Local Context Memory map

11.3.7 Encryption Configuration

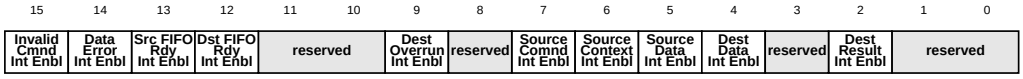
This bit determines how much local RAM is required for each encryption context. This bit is only significant if the COMPRESSION CONFIGURATION bit is set to one (to use only a single compression context).

If the ENCRYPTION CONFIGURATION bit is set to zero, 512 byte encryption contexts will be used. This is compatible with any encryption algorithm.

If this bit is set to one, 128 byte encryption contexts will be used. This configuration cannot be used if the RC4 encryption algorithm is desired.

The MAC algorithms require 40 bytes of context, which is included in the encryption portion of the context RAM.

11.4 Processing Unit Interrupt Enable Register



This register may be read or written. It is used to configure the conditions under which an interrupt will be generated. The default value of all the fields in this register after a reset is zero.

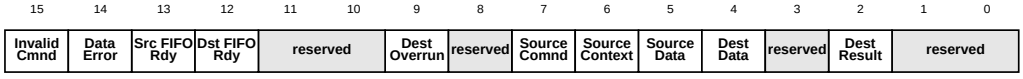
The bits in this register determine if the corresponding bits in the Interrupt Status register affect the IRQ signal. If a Processing Unit Interrupt Enable register bit is set to zero, the corresponding Processing Unit Interrupt Status register bit will not affect the IRQ signal.

If the Interrupt Enable bit is set to one, the corresponding Processing Unit Interrupt Status register bit will affect the IRQ signal. In this case, if the corresponding Processing Unit Interrupt Status register bit is a one, the IRQ signal will be asserted.

Writing to this register will not affect the values in the Processing Unit Interrupt Status register.

The relationship of the Processing Unit Status register bits, the Processing Unit Interrupt Status register bits, and the Processing Unit Interrupt Enable register bits is shown in Figure 26 which may be found in the Processing Unit Interrupt Status register description.

11.5 Processing Unit Status Register



This register may be read only. It is used to monitor the status of the chip. The default value of the bits in this register are listed under each bit description.

Some of the status bits indicate a “Fatal Error”. When a fatal error occurs, operation of the Processing Units will stop. The FIFOs will continue to operate to allow information residing in the Destination FIFO to be read. The processing units will not start again until the 7751 is reset, either by a software reset (MSTRESET#) or a hardware reset.

11.5.1 Invalid Command

This bit is set to one if a command is issued that contains an invalid parameter. It is not guaranteed that all invalid parameters will be detected.

This condition is a Fatal Error. See the Fatal Error description at the beginning of this register for more details concerning Fatal Errors. The default value of this bit is zero.

11.5.2 Data Error

This bit is set to one if the Data register is written when it is not ready, or the Data register is read when it is not ready.

This condition is a Fatal Error. See the Fatal Error description at the beginning of this register for more details concerning Fatal Errors. The default value of this bit is zero.

11.5.3 Source FIFO Ready

This bit is set to one when the available space in the Source FIFO exceeds the threshold programmed in the FIFO Configuration register.

This bit will be set to zero when the available space in the Source FIFO is less than, or equal to the threshold programmed in the FIFO Configuration register.
The default value of this bit is zero.

11.5.4 Dest FIFO Ready

This bit is set to one when the number of bytes in the Destination FIFO exceeds the threshold programmed in the FIFO Configuration register, or when the last byte of data from the result phase has entered the Dest FIFO.

This bit will be set to zero when the number of bytes in the Destination FIFO is less than, or equal to the threshold programmed in the FIFO Configuration register.
The default value of this bit is zero.

11.5.5 Dest Overrun

This bit is set to one when the number of bytes produced by the command exceeds the dest count of the command.

11.5.6 Source Command, Source Context, Source Data

These three bits indicate the current state of the Source FIFO: Command, Context, or Data phase. One and only one of these bits will be set at a time. The Source FIFO state is relative to the input of the FIFO.

11.5.7 Dest Data, Dest Result

These two bits indicate the current state of the Destination FIFO: Data or Result phase. One and only one of these bits will be set at a time. The Destination FIFO state is relative to the output of the FIFO.

11.5.8 Chip ID

When the CHIPID bit (bit 5) in the Processing Unit Configuration register is set, this register returns the Chip ID value. The upper 8 bits are defined as the product ID code and the lower 8 bits are reserved.

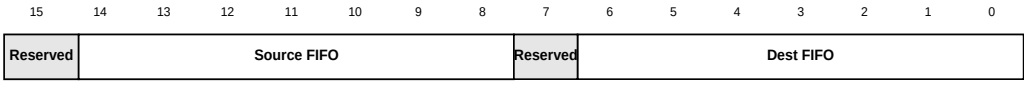
Setting the CHIPID bit allows only one read of this register. If the ChipID value is to be read again the CHIPID bit in the Processing Unit Configuration register must be set to one again.

There are 3 Chip ID values that may be returned, based on the security level that is enabled by the unlock procedure (see section 5.1.2 Enabling Encryption). Figure 30 summarizes these Chip IDs, along with the associated security levels.

7751 Chip ID	Security Level
0x30XX	Security Level 0 (default)
0x10XX	Security Level 1
0x11XX	Security Level 2

Figure 30. 7751 Security Levels

11.6 FIFO Status Register



This is a read-only register. It is used to determine the number of bytes residing in the Source FIFO and Dest FIFO.

During normal operation, the DMA interface should be used to transfer data to the Source FIFO and from the Dest FIFO. Therefore, this register should only be used under special conditions, or for testing or debugging.

The Source FIFO and Dest FIFO are each 64 bytes in size.

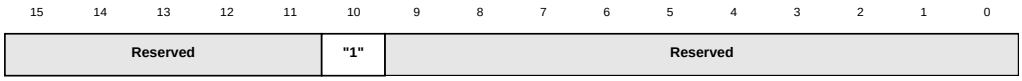
11.6.1 Source FIFO

This field indicates the number of bytes available to be written to the Source FIFO. For example, if the Source FIFO is empty, this field will indicate 64. The default value of this field is 64.

11.6.2 Dest FIFO

This field indicates the number of bytes residing in the Dest FIFO. For example, if the Dest FIFO is empty, this field will indicate zero. The default value of this field is zero.

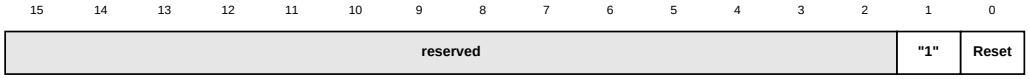
11.7 FIFO Configuration Register



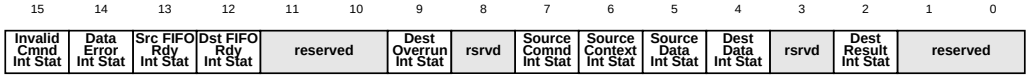
This register may be read or written. The value 0x0400 must be written to this registe prior to 7751 processing any data.

12 Processing Unit Register Summary

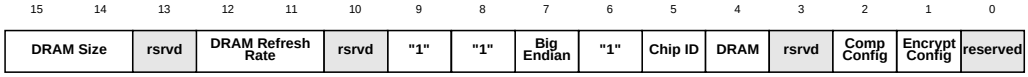
Processing Unit Control register



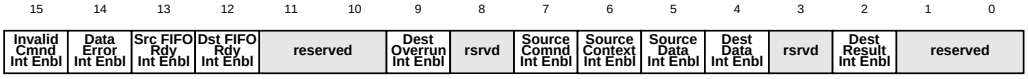
Processing Unit Interrupt Status register



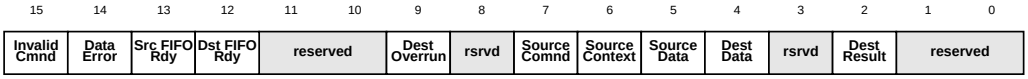
Processing Unit Configuration register



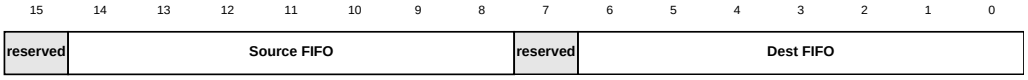
Processing Unit Interrupt Enable register



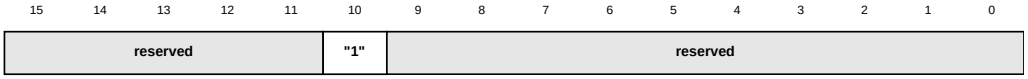
Processing Unit Status register



FIFO Status register



FIFO Configuration register



13 Descriptor Structures

Reserved bits must be written as zeros and ignored when read. Bits labeled "1" must be written as 1 and ignored when read. Bits labeled "0" must be written as 0 and ignored when read.

13.1 Overview

Descriptor structures reside in shared memory and can be accessed by the host system and the 7751. They are composed of descriptors that follow the Command, Source Data, Result, or Destination Data Descriptor formats defined on the following pages. Descriptors contain control information used by the 7751 to manage memory moves between shared memory and the 7751.

Descriptors are read by the 7751 at the beginning of processing and updated by the 7751 with completed status at the end of a data transfer. Descriptors may be linked together to form descriptor rings.

13.2 Command Block

Command blocks control the operation of the 7751 engines. Command blocks identify some information on the source data, and how the 7751 will process this source data. This information is transferred to the 7751 at the start of an operation.

13.3 Source Data Block

Source data blocks contains the actual data which will be sent to the 7751 for processing. A single command may operate on any number of source blocks, which are referred to as buffer fragments.

Context information may be appended to the command block. In this case, the CPU must set the LAST bit in the descriptor that points to the context information. Alternatively, the context information may be prepended to the source data block. In this case the system must set the LAST bit in the descriptor that points to the source data.

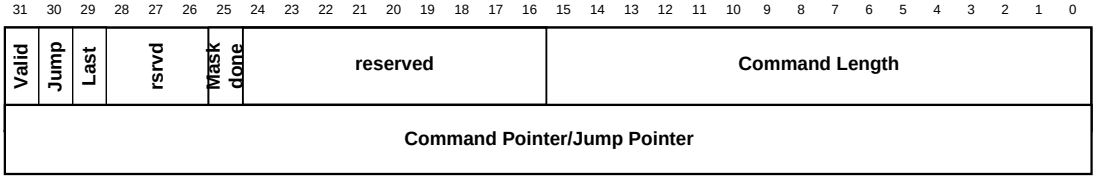
13.4 Destination Data Block

Destination data buffers are where the 7751 will put all processed data. A single command may distribute data to any number of destination buffers, which are referred to as buffer fragments.

13.5 Result Block

Result blocks contain the results for a specific command. These results indicate the success or failure of a 7751 command, as well as other information such as the destination byte count. After the completion of a command, the Result block is automatically transferred by the 7751 to a specified shared memory buffer.

13.6 Command Descriptor Format



The format of the Command Descriptor is shown above. This descriptor is used to point to a block (or blocks) of command data that is stored in system memory. Multiple command descriptors, which also reside in system memory, can be linked to form descriptor rings. This allows the 7751 to transfer non-contiguous blocks of command data from system memory. The structure of the Command Descriptor is described below.

13.6.1 Valid (bit 31)

This bit is set by the host CPU and reset by the 7751.

The 7751 polls the VALID bit of the next descriptor in each descriptor ring to determine when the descriptor is ready to be processed. Therefore, when the host software is

preparing a descriptor, setting this bit to a one must be the last operation performed. The 7751 will clear this bit to a zero as the last operation in processing the descriptor.

If the VALID bit is set, but the length is set to zero (and the jump bit is not set), the 7751 will continue to poll until the length becomes non-zero.

13.6.2 Jump (bit 30)

This bit is read-only by the 7751 and write-only by the host CPU.

If this bit is set to one, the descriptor does not contain command information. Instead, it contains a pointer to the next descriptor in the descriptor ring. This can either link a separate linear section of the ring, or point the end of the ring to the beginning. Software must ensure that these pointers are valid before starting the 7751 DMA engine.

13.6.3 Last (bit 29)

This bit designates whether the current descriptor is the last descriptor of a command buffer. This bit will be set by the host CPU.

13.6.4 Maskdone (bit 25)

This bit is read-only by the 7751 and write-only by the host CPU.

When this bit is set, it will mask the setting of the DONE interrupt (for the Command Ring)..

13.6.5 Command Length (bits 15-0)

This field is read-only by the 7751 and write-only by the host CPU.

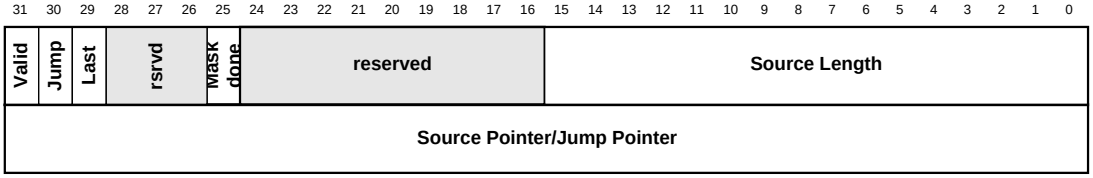
This field specifies the number of bytes in the command buffer. For all valid, non-jump descriptors, this value must be non-zero.

13.6.6 Command Pointer/Jump Pointer (bits 31-0)

This field is read-only by the 7751 and write-only by the host CPU.

This field contains the address in memory of the beginning of a command block.

13.7 Source Data Descriptor Format



The format of the Source Data Descriptor is shown above. This descriptor is used to point to a block (or blocks) of source data stored in system memory. Multiple source descriptors, which also reside in system memory, can be linked to form descriptor rings. This allows the 7751 to transfer non-contiguous blocks of source data from system memory. The structure of the Source Data Descriptor is described below.

13.7.1 Valid (bit 31)

This bit is set by the host CPU and reset by the 7751.

The 7751 polls the VALID bit of the next descriptor in each descriptor ring to determine when the descriptor is ready to be processed. Therefore, when the host software is preparing a descriptor, setting this bit to a one must be the last operation performed. The 7751 will clear this bit to a zero as the last operation in processing the descriptor.

If the VALID bit is set, but the length is set to zero (and the jump bit is not set), the 7751 will continue to poll until the length becomes non-zero.

13.7.2 Jump (bit 30)

This bit is read-only by the 7751 and write-only by the host CPU.

If this bit is set to one, the descriptor does not contain source data. Instead, it contains a pointer to the next descriptor in the descriptor ring. This can either link a separate linear section of the ring, or point the end of the ring to the beginning. Software must ensure that these pointers are valid before starting the 7751 DMA engine.

13.7.3 Last (bit 29)

This bit designates whether the current descriptor is the last descriptor of the Source Data buffer. This bit will be set by the host CPU.

13.7.4 Maskdone (bit 25)

This bit is read-only by the 7751 and write-only by the host CPU.

When this bit is set, it will mask the setting of the DONE interrupt (for the Source Data Ring).

13.7.5 Source Length (bits 15-0)

This field is read-only by the 7751 and write-only by the host CPU.

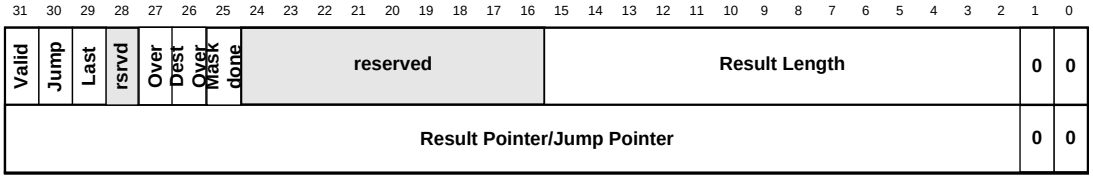
This field specifies the number of bytes in the source buffer. For all valid, non-jump descriptors, this value must be non-zero.

13.7.6 Source Pointer/Jump Pointer (bits 31-0)

This field is read-only by the 7751 and write-only by the host CPU.

This field contains the address in memory of the beginning of a source data block

13.8 Result Descriptor Format



The format of the result descriptor is shown above. This descriptor is used to point to a block (or blocks) of result data that is stored in system memory. Multiple result descriptors, which also reside in system memory, can be linked to form descriptor rings. This allows the 7751 to transfer non-contiguous blocks of result data to system memory. The structure of the result descriptor is described below.

13.8.1 Valid (bit 31)

This bit is set by the host CPU and reset by the 7751.

The 7751 polls the VALID bit of the next descriptor in each descriptor ring to determine when the descriptor is ready to be processed. Therefore, when the host software is preparing a descriptor, setting this bit to a one must be the last operation performed. The 7751 will clear this bit to a zero as the last operation in processing the descriptor.

If the VALID bit is set, but the length is set to zero (and the jump bit is not set), the 7751 will continue to poll until the length becomes non-zero.

13.8.2 Jump (bit 30)

This bit is read-only by the 7751 and write-only by the host CPU.

If this bit is set to one, the descriptor does not contain result information. Instead, it contains a pointer to the next descriptor in the descriptor ring. This can either link a separate linear section of the ring, or point the end of the ring to the beginning. Software must ensure that these pointers are valid before starting the 7751 DMA engine.

13.8.3 Last (bit 29)

This bit designates whether the current descriptor is the last descriptor of the Result buffer. This bit will be set by either the host or the 7751 based on the value of the LAST bit in the DMA Configuration register. Please see the description of the LAST bit in the DMA Configuration register for details on when this bit must or will be set.

13.8.4 Over (bit 27)

This bit signifies that the 7751 produced more results data than was allocated in the results descriptor (when the CPU is in control of the descriptor LAST flag). This bit is set under the same conditions that set the OVER bit of the Results Ring field (bit 17) in the DMA Status and Control register. This bit is only significant if the LAST bit (bit 4) in the DMA Configuration register was set to one. See sections 2.5, 2.6, and 2.7 for functional details on overrun conditions.

13.8.5 Dest Over (bit 26)

This bit signifies that the 7751 produced more destination data than was allocated in the destination descriptor for this result. This bit is set under the same conditions that set the OVER bit of the Destination Data Ring field (bit 25) in the DMA Status and Control register. This bit is only significant if the LAST bit (bit 4) in the DMA Configuration register was set to one. See sections 2.5, 2.6, and 2.7 for functional details on overrun conditions.

13.8.6 Maskdone (bit 25)

This bit is read-only by the 7751 and write-only by the host CPU.

When this bit is set, it will mask the setting of the DONE interrupt (for the Result Ring).

13.8.7 Result Length (bits 15-0)

This field can be read and written by the host CPU and by the 7751.

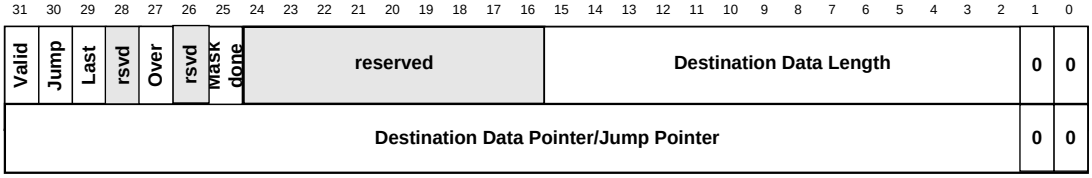
This field specifies the number of bytes in the result buffer. The host initializes this value to the maximum size of the buffer, then the 7751 will overwrite the value with the actual returned byte count.

For all valid, non-jump descriptors, this value must be non-zero. This value must be a multiple of four bytes, so bits 0 and 1 in this field must be zero.

13.8.8 Result Pointer/Jump Pointer (bits 31-0)

This field is read-only by the 7751 and write-only by the host CPU.
This field contains the address in memory of the beginning of a result block. This value must be a multiple of four bytes, so bits 0 and 1 in this field must be zero.

13.9 Destination Data Descriptor Format



The format of the Destination Data Descriptor is shown above. This descriptor is used to point to a block (or blocks) of destination data stored in system memory. Multiple destination data descriptors, which also reside in system memory, can be linked to form descriptor rings. This allows the 7751 to transfer destination data to non-contiguous blocks in system memory. The structure of the Destination Descriptor is described below.

13.9.1 Valid (bit 31)

This bit is set by the host CPU and reset by the 7751.

The 7751 polls the VALID bit of the next descriptor in each descriptor ring to determine when the descriptor is ready to be processed. Therefore, when the host software is preparing a descriptor, setting this bit to a one must be the last operation

performed. The 7751 will clear this bit to a zero as the last operation in processing the descriptor.

13.9.2 Jump (bit 30)

This bit is read-only by the 7751 and write-only by the host CPU.

If this bit is set to one, the descriptor does not contain destination data. Instead, it contains a pointer to the next descriptor in the descriptor ring. This can either link a separate linear section of the ring, or point the end of the ring to the beginning. Software must ensure that these pointers are valid before starting the 7751 DMA engine.

13.9.3 Last (bit 29)

This bit designates whether the current descriptor is the last descriptor of the Dest Data buffer. This bit will be set by either the host or the 7751 based on the value of the LAST bit in the DMA Configuration register. Please see the description of the LAST bit in the DMA Configuration register for details on when this bit must or will be set.

13.9.4 Over (bit 27)

This bit signifies that the 7751 produced more destination data than was allocated in the destination descriptor (when the CPU is in control of the descriptor LAST flag). This bit is set under the same conditions that set the Over bit of the Destination Data Ring field (bit 25) in the DMA Status and Control register. This bit is only significant if the Last bit (bit 4) in the DMA Configuration register was set to one. See sections 2.5, 2.6, and 2.7 for functional details on overrun conditions.

13.9.5 Maskdone (bit 25)

This bit is read-only by the 7751 and write-only by the host CPU.

When this bit is set, it will mask the setting of the DONE interrupt (for the Dest Ring).

13.9.6 Destination Length (bits 15-0)

This field can be read and written by the host CPU and by the 7751.

This field specifies the number of bytes in the destination buffer. The host initializes this value to the maximum size of the buffer, and the 7751 will then overwrite the value with the actual returned byte count.

For all valid, non-jump descriptors, this value must be non-zero. This value must be a multiple of four bytes, so bits 0 and 1 in this field must be zero.

13.9.7 Destination Pointer/Jump Pointer (bits 31-0)

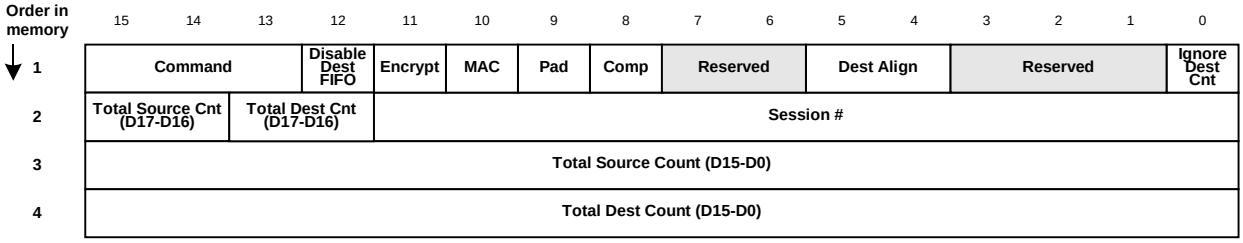
This field is read-only by the 7751 and write-only by the host CPU.

This field contains the address in memory of the beginning of a destination data block. This value must be a multiple of four bytes, so bits 0 and 1 in this field must be zero.

14 Command Structures

Reserved bits must be written as zeros and ignored when read. Bits labeled "1" must be written as 1 and ignored when read. Bits labeled "0" must be written as 0 and ignored when read.

14.1 Base Command Structure



All commands (except the Read RAM and Write RAM commands) begin with this Base structure. Fields within this structure determine whether additional structures will be required.

The Base structure contains general information that relates to the overall command. Most of this information is related to the Source FIFO and the Destination FIFO.

In order to reduce the number of bytes required for a complete Command structure, if the enable bit of a processing unit is not set to a one (as set by the first word of the Base structure), then the structure destined for that processing unit must not be present.

For example, if only the encryption processing unit is enabled, then only two structures must be sent—the Base structure and the Encryption structure.

If the Base Command Structure is the only structure sent (i.e. all four processing engines are disabled), eight additional bytes must be appended to the end of the Base Command Structure. The value of these bytes is reserved (all zeroes).

All commands will produce result information when they are completed. The result information will appear in the Destination FIFO as a result structure during the result phase.

14.1.1 Processing Unit Ordering

The processing units are arranged in-line with the data path. The ordering of the processing units is determined by the type of operation (encoding or decoding) and the value of the MAC POSITION field in the MAC structure (if present). A summary is shown in Figure 31.

Operation	MAC Position	1st	2nd	3rd	4th
Encode	After Comp	Comp	MAC	Pad	Encrypt
	After Pad	Comp	Pad	MAC	Encrypt
	After Encrypt	Comp	Pad	Encrypt	MAC
Decode	Before Decomp	Decrypt	Pad	MAC	Decomp
	Before Pad	Decrypt	MAC	Pad	Decomp
	Before Decrypt	MAC	Decrypt	Pad	Decomp

Figure 31. Processing Unit Ordering

14.1.2 Command Termination Conditions

All commands will terminate normally when the Total Source Counter reaches zero. A "normal termination" means that the command completes its operation, and the result information is properly appended to the output data stream.

A "fatal termination" means that all operations will immediately stop. The processing units will be cleared. No result information will be generated. The only way to recover from a fatal termination is to reset the 7751 with a software or hardware reset.

See the INVALID COMMAND field in the Processing Unit Status register for a description of the conditions that can cause a fatal termination.

14.2 Base Structure Fields

14.2.1 Command

The COMMAND field specifies the command to be executed by the enabled processing units. The valid values are listed in Figure 32. All other values are invalid.

Command	Command field
Encode	0
Decode	1
Read RAM	2
Write RAM	3

Note: All other values for the command field are reserved.

Figure 32. Commands

Encode
This command will encode a block of data using the Context specified in the SESSION # field. This command affects the operation of all the processing units. For example, the encode command will cause the compression processing unit to perform compression instead of decompression.

Decode
This command will decode a block of data using the Context specified in the SESSION # field. This command affects the operation of all the processing units. For example, the decode command will cause the compression processing unit to perform decompression instead of compression.

Read RAM
This command is used to read data directly from the Context RAM. This is normally used for debugging purposes or for memory testing.

See the *Read RAM structure* description for the definition of the special fields for this command.

Write RAM
This command is used to write data directly to the Context RAM. This is normally used for debugging purposes or for memory testing.

See the *Write RAM structure* description for the definition of the special fields for this command.

14.2.2 Disable Dest FIFO

If this bit is set to zero, the destination FIFO is enabled. After data passes through all the processing units, the data will enter the destination FIFO for DMA transfer out of the chip. This is the normal setting.

If this bit is set to one, the destination FIFO will be disabled. After data passes through all the processing units, the data will not be sent to the destination FIFO - it will be dropped. All processing units will operate normally, but no output data will be generated in the data phase. Context phase and result phase will produce data normally.

The Dest Counter will be decremented even when the Dest FIFO is disabled.

14.2.3 Comp

If this bit is set to one, the compression processing unit is enabled.

If this bit is set to zero, the compression processing unit is disabled. Data will pass through the compression processing unit unaltered.

14.2.4 Pad

If this bit is set to one, the pad processing unit is enabled.

If this bit is set to zero, the pad processing unit is disabled. Data will pass through the pad processing unit unaltered.

14.2.5 MAC

If this bit is set to one, the MAC processing unit is enabled.

If this bit is set to zero, the MAC processing unit is disabled. Data will pass through the MAC processing unit unaltered.

14.2.6 Encrypt

If this bit is set to one, the encryption processing unit is enabled.

If this bit is set to zero, the encryption processing unit is disabled. Data will pass through the encryption processing unit unaltered.

14.2.7 Dest Align

If it is desired to force the first byte of destination data to a non-32-bit (or non-16-bit) alignment, the DEST ALIGN field may be used to produce a few bytes of data that will be inserted in front of the destination data. The value of the inserted bytes are undefined.

The number of bytes inserted is set in this field, and values three to zero are valid. The inserted bytes are counted by the Dest Counter.

14.2.8 Ignore Dest Count

If this bit is set to one, an unlimited amount of data may enter the Destination FIFO. The Total Dest Counter will wrap from zero to 0x3FFFF.

If this bit is set to zero, the Destination FIFO will stop accepting data from the processing units when the Total Dest Counter reaches zero. The processing units will continue normally, but no data will be produced.

In either case, the Total Dest Counter will be initialized to the value set in the DEST COUNT field.

14.2.9 Session

For encode and decode commands this field specifies the context to be used.

The maximum number of sessions the 7751 can support is determined by the amount of context RAM attached to the chip, the compression algorithm used, and the compression mode set in the Processing Unit Configuration register.

If the COMPRESSION CONFIGURATION bit in the Processing Unit Configuration register is set to zero (meaning that multiple compression contexts are to be used), then the SESSION # field is restricted to 11 bits. In this case bit 11 is reserved.

If the COMPRESSION CONFIGURATION bit in the Processing Unit Configuration register is set to one (meaning that only one compression context is to be used), then the SESSION # field comprises all 12 bits.

14.2.10 Total Source Count

This is the initial value used for the Total Source Counter. The Total Source Counter is decremented for each data phase source byte processed by the processing units.

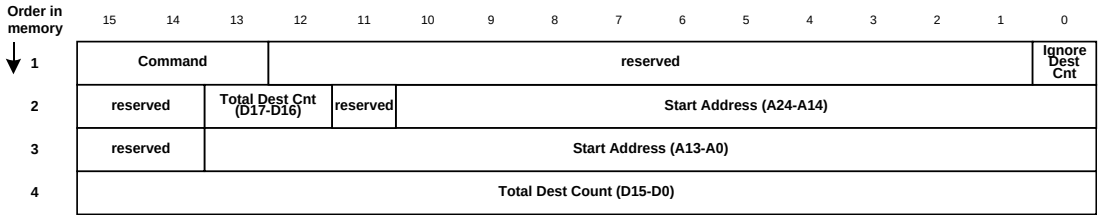
Total Source Count will decrement for every data phase byte processed. It will not decrement for any extra bytes resulting from non-word alignment in the last transfer.

14.2.11 Total Dest Count

This is the initial value used for the Total Dest Counter. The Total Dest Counter is decremented for each data phase byte entering the destination FIFO.

Total Dest Count will decrement for every data phase byte processed, including bytes specified by the DEST ALIGN field. It will not decrement for any extra bytes inserted due to non-word alignment in the last transfer.

14.3 Read RAM Command Structure



If the COMMAND field contains the Read RAM command, then the Base command structure will be formatted as described here.

During a Read RAM command, the Read RAM Command Structure should be sent as the command data block. The source data block must consist of eight bytes of zero's. Note that a source data block must be included in order for the Read RAM command to operate correctly.

The destination data block will consist of the contents in the context RAM, and results will be passed in the result block.

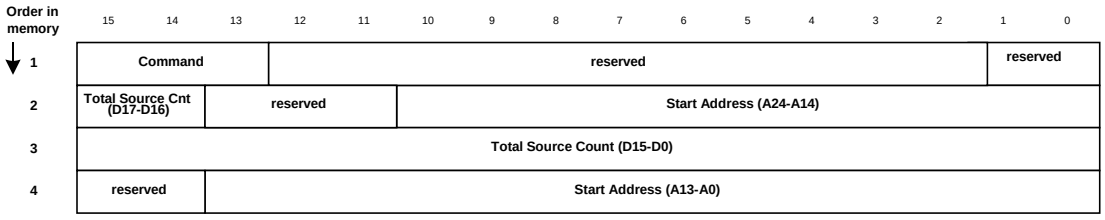
14.3.1 Total Dest Count

The TOTAL DEST COUNT field determines the number of bytes to read. The operation of this command requires that the Context RAM address bits 24-14 remain constant. So, the value of the TOTAL DEST COUNT and START ADDRESS fields must be chosen to prevent the Context RAM address bits 24-14 from incrementing. Bit 0 in the TOTAL DEST COUNT field must be set to zero. The minimum value of the TOTAL DEST COUNT field is four.

14.3.2 Start Address

This field determines the starting Context RAM address that will be read from the Context RAM. Bit 0 of the START ADDRESS field must be zero.

14.4 Write RAM Command Structure



If the COMMAND field contains the Write RAM command, then the Base command structure will be formatted as described here.

During a Write RAM Command, eight additional bytes must be appended to the end of the Write RAM Command Structure. The value of these bytes is reserved.

The processing unit produces 8 bytes of results, of which four bytes are transferred via the destination data phase and four via the results phase. The DMA unit must be programmed for four bytes of destination data and four bytes of results data. There is no destination data produced by processing unit for this command.

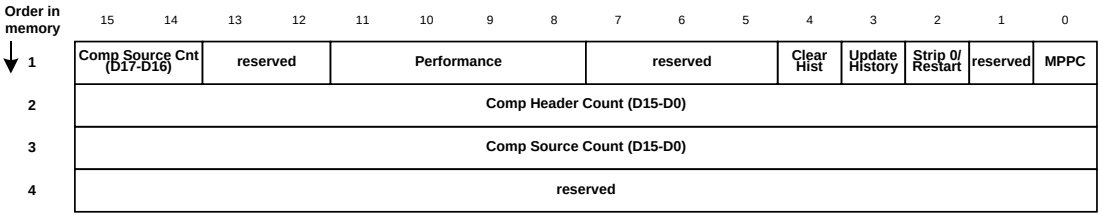
14.4.1 Total Source Count

The TOTAL SOURCE COUNT field determines the number of bytes to write. The operation of this command requires that the Context RAM address bits 24-14 remain constant. So, the value of the TOTAL SOURCE COUNT and START ADDRESS fields must be chosen to prevent the Context RAM address bits 24-14 from incrementing. Bit 0 in the TOTAL SOURCE COUNT field must be set to zero. The minimum value of the TOTAL SOURCE COUNT field is four.

14.4.2 Start Address

This field determines the starting Context RAM address that will be written to the Context RAM. Bit 0 of the START ADDRESS field must be zero.

14.5 Compression Command Structure



If the COMP bit in the Base command structure is set to one, then the Compression structure will follow the Base command structure.

14.5.1 Performance

This field is valid only for an encode operation, and is reserved for a decode operation. This field sets the compression performance. In general, there is a trade-off between compression speed and compression ratio. The fastest performance setting is one, and the performance degrades as the value of this field is increased. The fastest setting (one) produces the lowest compression ratio. The slowest setting (zero) produces the highest compression ratio.

Figure 35 illustrates the effect of the PERFORMANCE field when compressing the text of the Constitution of the United States when the Context RAM is SRAM.

Figure 36 illustrates the effect of the PERFORMANCE field when compressing the text of the Constitution of the United States when the Context RAM is DRAM.

14.5.2 Clear History

If this bit is set to one, the compression (or decompression) history specified by the SESSION # field will be cleared before the operation begins.

If this bit is set to zero, the compression (or decompression) history will not be cleared. The operation will use history information from previous operations which used the same context.

This bit must be set to one for the first compress operation which uses the specified context after a hardware reset.

Due to the nature of the LZS compression format, it is never necessary to clear a Decompression History. Also, it is not necessary to reset the MPPC decompression history unless specified in the MPPC protocol.

14.5.3 Update History

This bit is significant for decompress operations only. If this bit is set to zero, the decompression operation will be processed normally.

If this bit is set to one, the decompress operation will pass data through the compression processing unit unaltered, similar to the pass-through operation that occurs if the compression processing unit is disabled. However, the decompression history within the specified context will be updated with the uncompressed source data.

14.5.4 Strip 0/Restart

This bit has two functions depending on the compression algorithm selected. In LZS mode (the MPPC bit is set to zero), this bit is known as the STRIP 0 bit, and is used to enable the "Strip 0" mode of the LZS compression format. The Strip 0 bit cannot be set if decompressing with the padding processing unit disabled.

In MPPC mode (the MPPC bit is set to one), this bit is known as the RESTART bit, and is used to implement the “restart” function of the MPPC protocol.

Enabling the Strip 0 feature may reduce the size of the compressed data stream by one byte. If this bit is set to one on a Compress operation, the last byte of compressed data is eliminated if the value of this last byte is zero. Based on the LZS format, this last byte is always part of the End Marker and will be zero approximately 88% of the time. If the last byte is eliminated, it will not be counted by the Dest Counter. If padding is enabled, then Strip 0 should not be used.

On a Decompress operation, a zero is inserted in the source compressed data stream just before the check field, or at the end of the compressed data stream if there is no check field. The inserted byte will not be counted by the Source Counter.

If the Strip 0 mode is enabled during a Decompress operation, the 7751 must know the exact number of source bytes so that it can insert the zero at correct location in the data stream. The pad length must contain the exact number of padding bytes.

Many data communication standards define this feature as an option. However, this mode is incompatible with the ANSI X3.241-1994 compression format standard.

If the STRIP 0 bit is set to zero, this feature will be disabled.

In MPPC mode this bit is used to signal the decompression engine to move the data to the front of the compression history, as specified in the MPPC protocol. This bit applies only to the decompression operation when the MPPC bit is also set. If this bit is set to one the packet is moved to the front of the compression history. If this bit is set to zero the packet is not moved to the front of the compression history.

During a compression operation, the processing unit automatically moves the data to the beginning of the history buffer as required, so the RESTART bit must be set to zero.

14.5.5 MPPC

This bit selects the compression algorithm used. If this bit is set to zero, the LZS algorithm will be used. If this bit is set to one, the MPPC algorithm will be used.

Once a compression (or decompression) history within a context is cleared (with the CLEAR HIST bit), the selected compression algorithm must remain constant whenever that context is used again until the compression history is cleared.

This bit also affects the context memory map as described in the Context section.

LZS Mode
When the LZS format is selected (the MPPC bit is set to zero), each full duplex session context requires 16 Kbytes of Session RAM. Although an individual encoding session context is functionally independent of a decoding session context, the associated memory areas are allocated together in a single 16 Kbyte memory block.

Session type	Session # field value	Actual CRAM location
1st encode session context	0	0
1st decode session context	0	0
2nd encode session context	1	16K
2nd decode session context	1	16K

Figure 33. Example LZS Session RAM usage

MPPC Mode
When the MPPC format is selected (the MPPC bit is set to one), each half-duplex session context requires 16 Kbytes of Session RAM. Each encoding session requires 16 Kbytes, and each decoding session requires an additional 16 Kbytes. Both cannot exist in a single 16 Kbyte block of memory.

Session type	Session # field value	Actual CRAM location
1st encoding session context	0	0
1st decoding session context	1	16K
2nd encoding session context	2	32K
2nd decoding session context	3	48K

Figure 34. Example MPPC Session RAM usage

14.5.6 Comp Header Count

This field sets the number of bytes that the compression processing unit will let pass through before it begins processing the incoming data stream.

14.5.7 Comp Source Count

After the header bytes have been passed through (as specified by the HEADER COUNT field), the processing unit will begin processing data in the data stream. The processing unit will stop processing data and return to pass-through mode after the COMP SOURCE COUNTER reaches zero. The COMP SOURCE COUNTER will begin to decrement after the COMP HEADER COUNTER has expired.

During a compression operation, when the COMP SOURCE COUNTER reaches zero, or after the last byte (as defined by the Total Source Count) has been processed, the compression processing unit will flush out its internal data. If in LZS mode, an End Marker will be appended to the compressed data stream.

During a decompression operation, in LZS mode, if the End Marker is found before the COMP SOURCE COUNTER reaches zero, all the source data between the End Marker and the last source byte will be discarded. If the End Marker is present in the last source byte, then no data will be discarded. If there is no End Marker in the source data, the Decompression History may become invalid.

In MPPC mode, the maximum COMP SOURCE COUNTER value is 8192. Also, the last source byte must be the last byte produced by the original compression operation. Otherwise the Decompression History will become invalid.

If the compression processing unit is enabled, it must process at least one byte. The TOTAL SOURCE COUNT field in the Base Command Structure as well as the COMP HEADER COUNT and COMP SOURCE COUNT fields in the Compression Command Structure control the number of bytes processed by the compression processing unit. The minimum value of the COMP SOURCE COUNT is one.

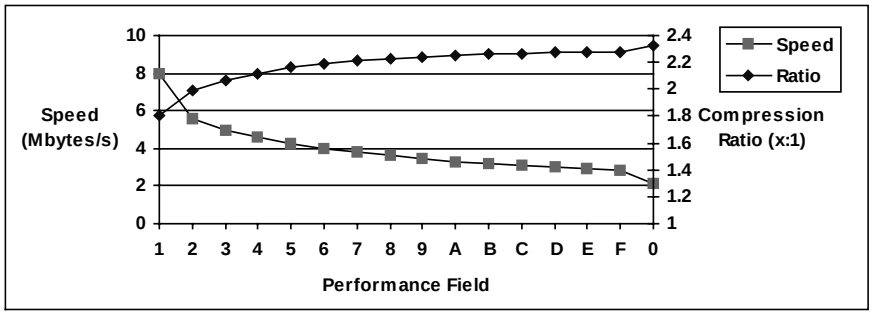


Figure 35. Effect of Performance field with SRAM

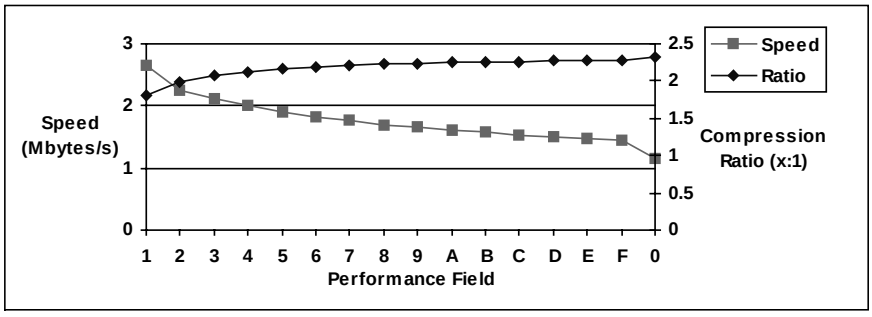
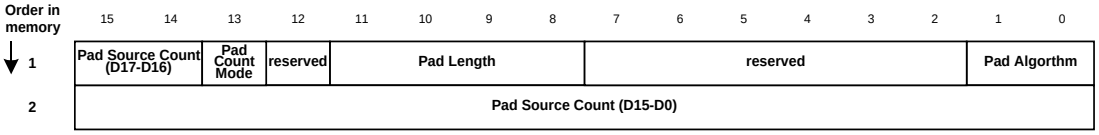


Figure 36. Effect of Performance field with DRAM

14.6 Pad Command Structure



If the PAD field in the Base Command structure is set to one, then the Pad structure will follow the Compression structure.

If the padding engine is the only engine enabled, four additional bytes of zero must be appended to the end of the Pad Command Structure.

When the Pad Processing Unit completes an encode operation, the specified pad will be inserted into the data stream. When the Pad Processing Unit completes a decode operation, the pad will be removed from the data stream.

14.6.1 Pad Count Mode

This bit is valid only for a encode operation. If this bit is set to zero, the pad Source Counter will begin to decrement after the last header byte has been passed through (as set by the Pad Header Count field). Note: The Pad Header Count field starting value is taken from the ENCRYPT HEADER COUNT field in the Encryption Command Structure.

If this bit is set to one, the Pad Source Counter will begin to decrement after the last byte processed by the previous processing unit (compression or MAC) as determined by the source counter of the previous processing unit.

14.6.2 Pad Length

This field operates differently for encoding than for decoding.

While encoding, this field is used to specify a count that the padding processing unit should add to the number of bytes processed by the padding unit to determine the correct number of padding bytes to insert. Generally this encompasses bytes that follow the padding field in the packet format. Only values zero through seven (0-7) are valid while encoding.

While decoding, this field is the number of bytes that the padding processing unit should strip off. The padding processing unit cannot automatically determine this value during a decode operation. Only values zero through eight (0-8) are valid while decoding.

During a decode operation, if LZS compression is used and padding appears immediately after the compressed data, the padding processing unit should be disabled, and the compression source count field set to strip off any extra padding.

If MPPC compression is used, or no compression is used, then the PAD LENGTH field should be set to the number of padding bytes that must be stripped.

14.6.3 Pad Algorithm

This field sets the algorithm used by the padding processing unit. All algorithms will pad out to modulo 8 bytes.

This field is only significant for an encode operation. During a decode operation, this field will be ignored.

Mode 0
The Padding Processing Unit will insert one to eight bytes, with at least one byte being inserted. The value of all bytes will be the number of bytes inserted less one.

Example: If three bytes are required to pad out to modulo 8, three bytes will be inserted, each with a value of two.

Mode 1
The Padding Processing Unit will insert one to eight bytes. The byte values will be an incrementing byte value, starting with the value one.

Mode 2
The Padding Processing Unit will insert zero to seven bytes. The bytes will all be the same value. The value will be the number of bytes inserted. Zero bytes may be inserted.

Mode 3
The Padding Processing Unit will insert two fields. The first field will be zero to seven bytes. The byte values will be an incrementing byte value, starting with the value one.

The second field will be exactly one byte. The value will be the number of bytes inserted in the first field.

Value	Algorithm
0 0	Mode 0
0 1	Mode 1
1 0	Mode 2
1 1	Mode 3

Figure 37. Pad Algorithms

14.6.4 Pad Header Count

This field does not exist in the Pad Command Structure. The value of the Pad Header Count will be the same as the value set in the Encrypt Header Count in the Encryption Command Structure. If the Encryption Processing Unit is disabled, then the value of Pad Header Count will be zero.

This value sets the number of bytes that the padding processing unit will let pass through before it begins processing the incoming data stream.

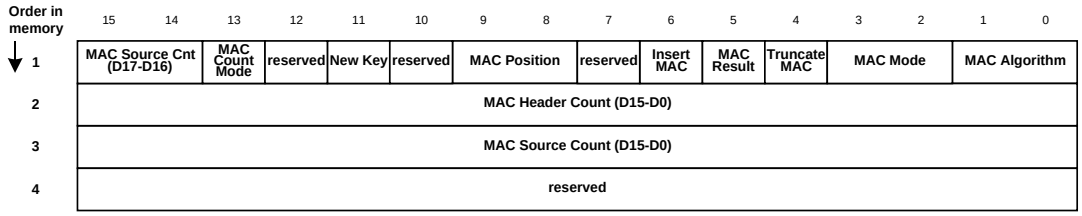
14.6.5 Pad Source Count

The Padding processing unit will process data after the PAD HEADER COUNT reaches zero, until the PAD SOURCE COUNT reaches zero. When the PAD SOURCE COUNTER reaches zero, or after the last byte (as defined by the Total Source Count) has been processed, the padding processing unit will stop.

The PAD SOURCE COUNTER will begin to decrement after the PAD HEADER COUNTER has expired, or after the last byte processed by the previous processing unit, as set by the PAD COUNT MODE bit.

After the PAD SOURCE COUNTER expires, all data that follows (as defined by the Total Source Count) will be passed through.

14.7 MAC Command Structure



If the MAC field in the Base command structure is set to one, then the MAC structure will follow the Pad structure.

When the MAC Processing Unit completes an encode operation, the MAC will be calculated and will be ready to be inserted into the data stream (as determined by the INSERT MAC and MAC RESULT bits). When the MAC Processing Unit completes a decode operation, the MAC will be calculated and will be ready to be compared to the MAC in the data stream.

14.7.1 MAC Count Mode

This bit is valid only for a encode operation. If this bit is set to zero, the MAC Source Counter will begin to decrement after the last header byte has been passed through (as set by the MAC Header Count field).

If this bit is set to one, the MAC Source Counter will begin to decrement after the last byte processed by the previous processing unit (compression, padding, or encryption) as determined by the source counter of the previous processing unit.

14.7.2 New Key

This bit selects if a new MAC key is to be supplied. If a new key is supplied, it must appear in the Source FIFO context phase. The key is a representation of the binary key as defined in *Source Context* section.

This bit must be zero if the MAC MODE field is set to hash only (10).

14.7.3 MAC Position

This field selects the logical position of the MAC processing unit in relationship to the other processing units. The valid values are shown in Figure 38.

MAC Position value	MAC relative location
0 0	between compression and padding
0 1	between padding and encryption
1 0	after encryption (or before decryption for decode)
1 1	Reserved

Figure 38. MAC Position

14.7.4 Insert MAC

This bit selects if the MAC is inserted into the data stream. If this bit is set to zero, the MAC will not be inserted into the data stream after it is calculated.

If this bit is set to one on an encode operation, the MAC will be inserted into the data stream after it is calculated. If set to a one on a decode operation, the MAC will be stripped from the data stream (after it is calculated), and the stripped MAC will be compared to the calculated MAC. A miscompare will be indicated by the MAC MISCOMPARE bit in the Result Phase.

See Figure 41 for a summary of the relationship between this bit, and other bits in the MAC Command Structure.

14.7.5 MAC Result

This bit selects if the MAC is inserted into the MAC Result Structure. If this bit is set to zero, the MAC will not be inserted into the MAC Result Structure after it is calculated. If this bit is set to one, the calculated MAC will be inserted into the MAC Result Structure after it is calculated.

See Figure 41 for a summary of the relationship between this bit, and other bits in the MAC Command Structure.

14.7.6 Truncate MAC

This bit selects if the MAC is truncated to 12 bytes. If this bit is set to zero, the MAC will be composed of the full, calculated MAC. The MAC will be 20 bytes for SHA, or 16 bytes for MD5.

If this bit is set to one, the MAC will be truncated to 12 bytes. The most significant bytes (leading bytes) will be truncated.

14.7.7 MAC Mode

This field selects the MAC mode as defined in Figure 39.

Value	Mode
0 0	HMAC
0 1	SSL MAC*
1 0	hash only
1 1	reserved

Note: The SSL MAC Mode is valid only if the MD5 MAC algorithm is selected. The SHA algorithm may not be used in SSL MAC Mode.

Figure 39. MAC Modes

14.7.8 MAC Algorithm

This field selects the MAC hash algorithm as defined in Figure 40.

Value	Algorithm
0 0	SHA
0 1	MD5
1 0	Reserved
1 1	Reserved

Figure 40. MAC Hash Algorithms

14.7.9 MAC Header Count

This field sets the number of bytes that the MAC processing unit will let pass through before it begins processing the incoming data stream.

14.7.10 MAC Source Count

The MAC processing unit will process data after the MAC HEADER COUNT reaches zero. When the MAC SOURCE COUNTER reaches zero, or after the last byte (as defined by the Total Source Count) has been processed, the MAC processing unit will stop.

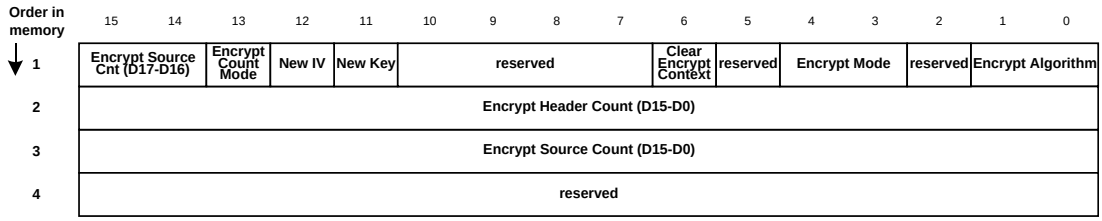
The MAC SOURCE COUNTER will begin to decrement after the MAC HEADER COUNTER has expired, or after the last byte processed by the previous processing unit, as set by the MAC COUNT MODE bit.

After the MAC SOURCE COUNTER expires, all data that follows (as defined by the Total Source Count) will be passed through.

Encode/ Decode	Insert MAC	MAC Result	Behavior
Encode	0	0	MAC is calculated, but not used.
Encode	0	1	MAC is calculated and inserted into MAC Result Structure.
Encode	1	0	MAC is calculated and inserted into data stream.
Encode	1	1	MAC is calculated and inserted into both the data stream and the MAC Result Structure.
Decode	0	0	MAC is calculated, but not compared to MAC in data stream.
Decode	0	1	MAC is calculated, but not compared to MAC in data stream. Calculated MAC is inserted into MAC Result Structure.
Decode	1	0	MAC is calculated and compared with MAC in data stream.
Decode	1	1	MAC is calculated and compared with MAC in data stream. Calculated MAC is also inserted into MAC Result Structure.

Figure 41. MAC Behavior

14.8 Encryption Command Structure



If the ENCRYPT field in the Base structure is set to one, then the Encryption structure will follow the MAC structure.

14.8.1 Encrypt Count Mode

This bit is valid only for a encode operation. If this bit is set to zero, the Encrypt Source Counter will begin to decrement after the last header byte has been passed through (as set by the Encrypt Header Count field).

If this bit is set to one, the Encrypt Source Counter will begin to decrement after the last byte processed by the previous processing unit (compression, padding, or MAC) as determined by the source counter of the previous processing unit.

14.8.2 New IV

This bit is only significant for the DES and Triple-DES algorithms, and only if the CBC, CFB, or OFB modes are enabled. This bit selects if a new Encryption Initialization Vector (IV) is to be supplied. Many parameters affect whether an IV needs to be supplied. See the Source Context Structure section for details. If a new IV is supplied, it must appear in the Source FIFO context phase.

This bit must be zero if the ALGORITHM field is set to RC4 (10) or if the ENCRYPT MODE field is set to ECB (00). If this bit is set to zero, the last 8 bytes of cipher text from the previous block in the same session utilizing the same command will be used as the IV.

14.8.3 New Key

This bit selects if a new Encryption key is to be supplied. If a new key is supplied, it must appear in the Source FIFO context phase. The key is a representation of the binary key as defined in *Source Context* section.

14.8.4 Clear Encrypt Context

If this bit is set to zero, the encryption context will be saved normally.

If this bit is set to one, the encryption context will not be saved to local RAM. This may improve speed performance if it is known in advance that the context will not be used in the future.

When this bit is set, the encryption context residing in the Context RAM will remain valid. That is, if this bit is set for a command, not only will the encryption context for the current command not be written to Context RAM, but the encryption context from the previous command (with the same session number) will remain valid.

14.8.5 Encrypt Mode

This field is only significant if the DES or Triple-DES algorithms are selected in the ALGORITHM field. This field selects the mode of the algorithm as defined in Figure 42.

If DES or Triple-DES is not selected, this field must be set to zero.
The 7751 CFB mode is CFB-64.

Value	Mode
0 0	ECB
0 1	CBC
1 0	CFB
1 1	OFB

Figure 42. Encrypt Mode

14.8.6 Encrypt Algorithm

This field selects the encryption algorithm as defined in Figure 43.

If the DES or Triple-DES encryption algorithms are used, the source data for the encryption processing unit must be a modulo 8 length. This is usually accomplished by enabling the padding processing unit.

Value	Algorithm
0 0	DES
0 1	Triple-DES
1 0	RC4
1 1	reserved

Figure 43. Encrypt Algorithm

14.8.7 Encryption Header Count

This field sets the number of bytes that the encryption processing unit will let pass through before it begins processing the incoming data stream.

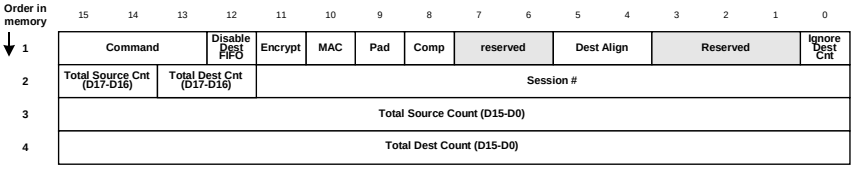
14.8.8 Encryption Source Count

The Encryption processing unit will process data after the ENCRYPTION HEADER COUNT reaches zero, until the ENCRYPTION SOURCE COUNT reaches zero. When the ENCRYPTION SOURCE COUNTER reaches zero, or after the last byte (as defined by the Total Source Count) has been processed, the Encryption processing unit will stop.

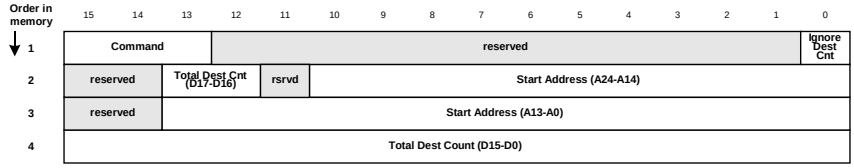
The ENCRYPTION SOURCE COUNTER will begin to decrement after the ENCRYPTION HEADER COUNTER has expired, or after the last byte processed by the previous processing unit, as set by the ENCRYPTION COUNT MODE bit.
After the ENCRYPTION SOURCE COUNTER expires, all data that follows (as defined by the Total Source Count) will be passed through.

15 Command Structure Summary

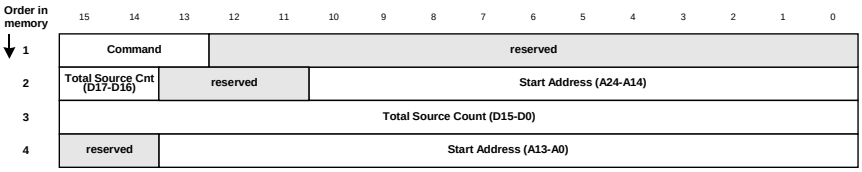
Base



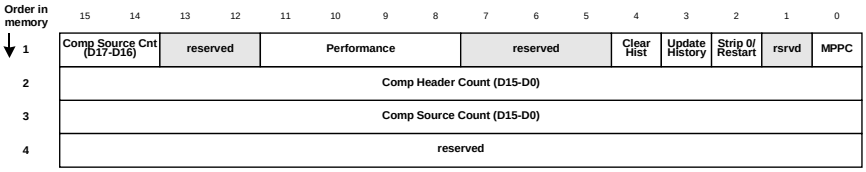
Read RAM



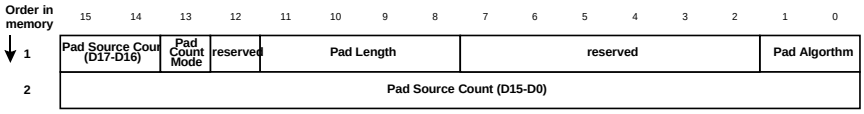
Write RAM



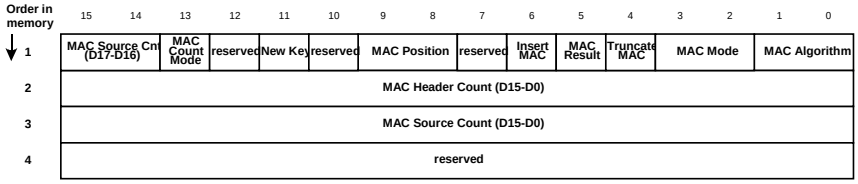
Compression



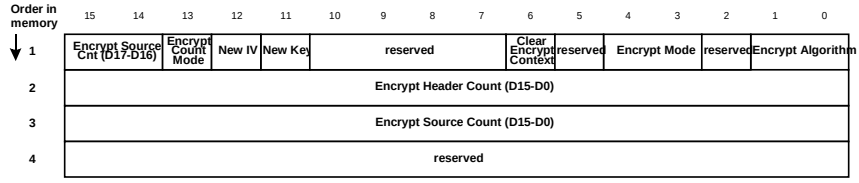
Pad



MAC



Encrypt



16 Source Context Structures

MAC and encryption information will be transferred to the 7751 in the Source Context phase. This information (if required by the device) will consist of a MAC key, and encryption key, or an encryption IV.

Source context is transferred to the 7751 in the following order: MAC key, encryption key, and then encryption IV.

16.1 MAC Source Context Structure

MAC Source Context, which consists of the MAC key, must be provided if it is expected by the 7751. The chip will expect a MAC key if the NEW KEY bit in the MAC Command Structure is set.

New key information must be provided as a string of 64 data bytes, formatted as follows:

$K_0, K_1, \ldots, K_{n-1}, 0_0, 0_1, \ldots, 0_{64-(n-1)},$

where K is the key, and K_0 is the least significant byte of the key, and n is the length of the key in bytes. 0_n represent the zero value used as padding in the key.

Note that when the MAC key is sent to the device, the key undergoes a hashing which takes about 300 clocks. Storing the MAC key information in Context RAM (as compared to sending the key with every command) will remove the need for this key hashing and may improve performance.

16.2 Encryption Source Context Structure

Encryption Source Context, which consists of an encryption key and/or IV, must be provided if it is expected by the 7751. The device will expect an encryption key if the new KEY BIT in the Encryption Command Structure is set, and it will expect a new IV is the NEW IV bit is set.

16.2.1 Keys

New key information must be provided as a formatted string. Each encryption algorithm requires a unique format as described below.

DES
The DES key is a string of 64 bits that are transferred to the 7751 in network byte order, as shown in Figure 44..

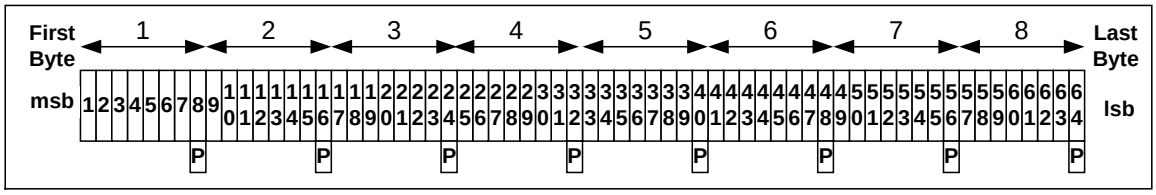


Figure 44. DES key format

Triple-DES
The Triple-DES key is a string of 168 bits that are transferred to the 7751 in network bytes order.

RC4
The key is a string of 260 bytes. The last four bytes must be zero. If the desired RC4 key is less than 256 bytes, the key must be repeated as required to complete 256 bytes, then appended with four zeros. The least significant byte is first. This may be represented as follows:

$K_0, K_1, \ldots, K_{n-1}, K_0, K_1, \ldots, K_{n-1}, \ldots, 0, 0, 0, 0$

Note: The last copy of the repeated key may be incomplete in order to satisfy exactly 256 bytes.

16.2.2 IV

The Initialization Vector (IV) is a string of 8 bytes, which are transferred to the 7751 in network byte order. Only DES and Triple-DES support this 8 byte IV. RC4 does not use an IV, nor does DES-ECB or Triple-DES-ECB.

17 Source Data Structures

Data entering the Source FIFO is simply a string of data bytes. No special formatting is required.

The Source Data Structure size must be a multiple of 32-bits. Unused bytes will be automatically discarded. The number of bytes used is set by the TOTAL SOURCE COUNT field in the Base Command Structure.

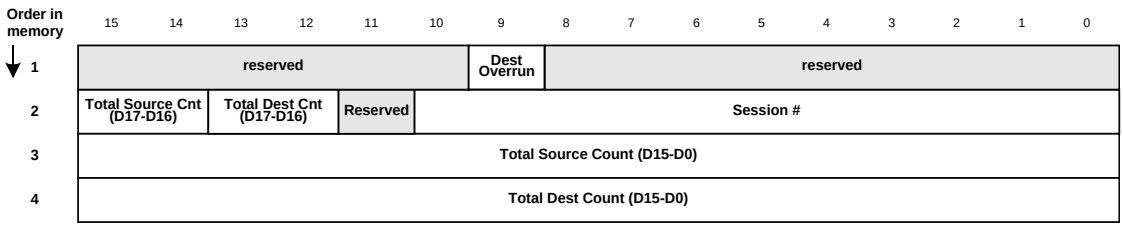
18 Destination Data Structures

Data produced by the Destination FIFO is simply a string of data bytes. No special formatting is performed.

The Destination Data Structure size will be a multiple of 32-bits. Unused bytes should be automatically discarded. The number of valid bytes may be determined by analyzing the TOTAL DEST COUNT field in the Base Result Structure.

19 Result Structures

19.1 Base Result Structure



The Result phase always begins with this structure for all commands. The options of the command executed determine whether additional structures will be produced. The smallest result contains at least the four words of the Base Result structure.

The Base Result structure contains general information that relates to the overall command. Most of this information is related to the Source FIFO and the Destination FIFO.

In order to reduce the number of bytes required for a complete Result structure, if the enable bit of a processing unit was not set to a one (as set by the first word of the Base Command structure), then a structure from that processing unit will not be presented.

For example, if only the Encryption processing unit was enabled, then only two structures will be produced—the Base Result structure and the Encryption Result structure.

19.1.1 Dest Overrun

This bit is set when the command produced more data than specified in the TOTAL DEST COUNT field of the Base Command Structure, and the IGNORE DEST COUNT bit was set to zero.

19.1.2 Session

This field contains the original value of the SESSION # field of the corresponding Base Command structure. This value may be used to help verify synchronization between commands and results. The Session # is undefined for a read RAM or write RAM command.

Note: bit 11 of the SESSION# field of the corresponding Base Command structure is not returned (i.e. it is Reserved) in the Base Result structure.

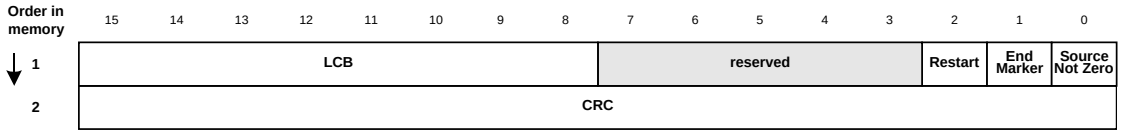
19.1.3 Total Source Count

This is the final value of the Total Source Counter at command termination. The Total Source Counter is decremented for each source byte taken from the Source FIFO, and should be zero if the command terminated cleanly. The total Source Count is undefined for a read RAM.

19.1.4 Total Dest Count

This is the final value of the Total Dest counter at command termination. The Total Dest counter is decremented for each byte that enters the Destination FIFO. The Total Dest Count is undefined for a write RAM.

19.2 Compression Result Structure



If the Compression processing unit is enabled, this data structure will be appended to the Base Result Structure.

19.2.1 LCB

This field is valid only if the compression processing unit was the only processing unit engaged. If any of the other engines were used in the command, then this field is reserved.

This field represents the 8-bit LCB (longitudinal Check Byte) for the command. The LCB is a byte-wise exclusive-OR sum of the uncompressed data. The LCB is initialized to FF_{16} before each compress or decompress operation.

19.2.2 Restart

This bit is significant for the MPPC algorithm only. This bit is used to indicate that the compression engine moved the data to the front of the compression context, as specified in the MPPC protocol. If this bit is set to one the data was moved to the front of the compression context. If this bit is set to zero the data was not moved to the front of the compression context.

19.2.3 End Marker

This bit is significant for the Decompress command while in LZS mode. If this bit is set to one, the Processing Unit detected the LZS End Marker in the source data stream. This condition may be one reason why the decompress operation terminated.

If this bit is set to zero, the End Marker was not detected before the operation terminated.

This bit is undefined and should be ignored in MPPC mode. This bit is also undefined if the UPDATE HISTORY bit in the Compression Command Structure is set.

19.2.4 Source Not Zero

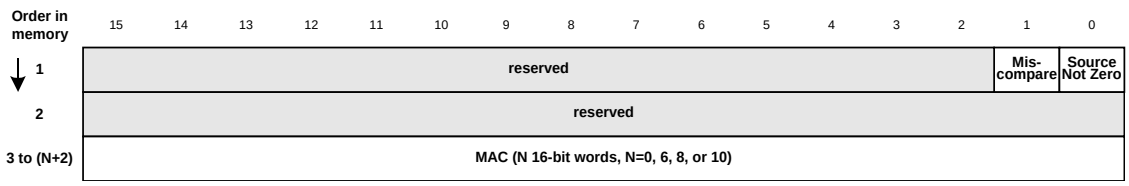
If this bit is set to one, the compression processing unit stopped processing a block of data before the COMP SOURCE COUNT reached zero. This may be caused by improper setting of the COMP SOURCE COUNT relative to the TOTAL SOURCE COUNT.

19.2.5 CRC

This field is valid only if the compression processing unit was the only processing unit engaged. If any of the other engines were used in the command, then this field is reserved.

This field represents the CRC value for the command. The CRC is initialized to "FFFF16" before each compress or decompress operation, and is performed on the uncompressed data. The CRC polynomial is $x^{16}+x^{12}+x^5+1$.

19.3 MAC Result Structure



If the MAC processing unit is enabled, this data structure will be appended to the previous Result Structure (Base or Compression).

19.3.1 Miscompare

During a decode operation, if the MAC processing unit detects a MAC miscompare, this bit will be set to one.

If there is no miscompare, this bit will be set to zero

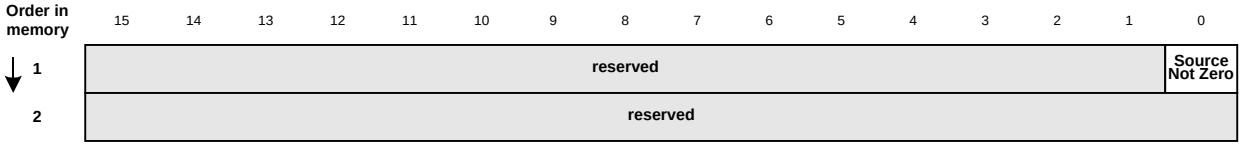
19.3.2 Source Not Zero

If this bit is set to one, the MAC processing unit stopped processing a block of data before the MAC SOURCE COUNT reached zero. This may be caused by improper setting of the MAC SOURCE COUNT relative to the TOTAL SOURCE COUNT.

19.3.3 MAC

This is the result of the MAC processing unit. The number of bytes produced is determined by the MAC algorithm selected and the value of the TRUNCATE MAC bit in the MAC Command Structure. No MAC will be presented unless the MAC RESULT bit in the MAC Command Structure is set to one.

19.4 Encrypt Result Structure



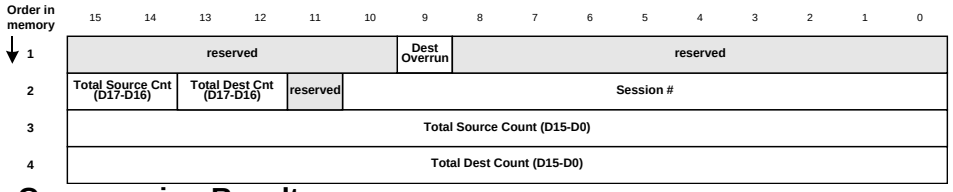
If the Encryption processing unit is enabled, this data structure will be appended to the end of the previous Result Structure (Base, Compression, or MAC).

19.4.1 Source Not Zero

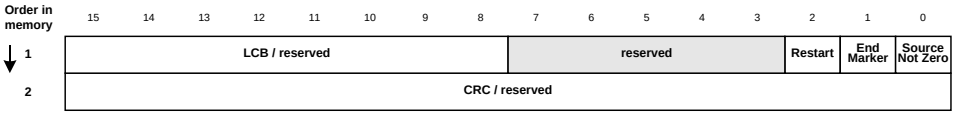
If this bit is set to one, the Encryption processing unit stopped processing a block of data before the ENCRYPTION SOURCE COUNT reached zero. This may be caused by improper setting of the ENCRYPTION SOURCE COUNT relative to the TOTAL SOURCE COUNT.

20 Result Structure Summary

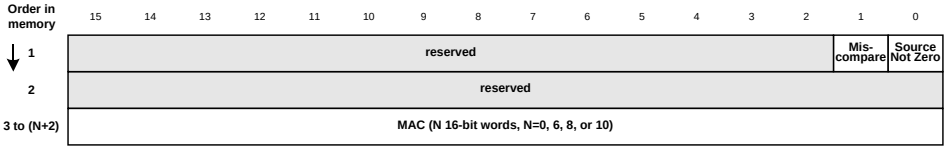
Base Result



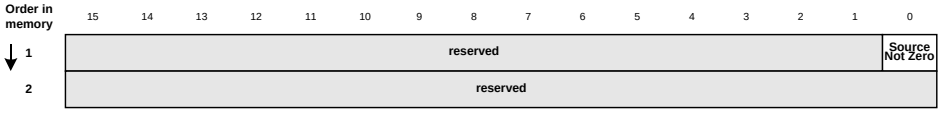
Compression Result



MAC Result



Encryption Result



21 Electrical Specifications

DC Supply Voltage (V_{DD})	-0.3V to +7.0V
DC Input Voltage	-0.3V to +7.0V
DC Input Current	±10mA
Storage Temperature	-40°C to +125°C

Caution: Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

Figure 45. Absolute maximum ratings

DC Supply Voltage	+3.0V to +3.6V
Operating Temperature	0°C to +70°C

Note: All input and I/O pins are 5 V tolerant. The supply voltage Vcc must not lag behind the voltage at an input pin by more than 0.5V as the supply voltage is ramped up to 2.7V. Once Vcc reaches 2.7V, the 5V tolerant inputs function normally.

Figure 46. Recommended operating conditions

Symbol	Parameter		Conditions	Min	Typ	Max	Units
V_{IL}	Low level input voltage (I, SI)					0.8	V
		Clock Input (CI)				0.8	V
V_{IH}	High level input voltage (I, SI)			2.0			V
		Clock Input (CI)		2.4			V
V_H	Schmitt hysteresis (SI)				0.8		V
I_{IL}	Low level input current (I, SI)		$V_{IN} = V_{SS}$ $V_{DD} = 3.6V$	-10			µA
		With pullup (PI)		-40		-5	µA
I_{IH}	High level input current (I, SI)		$V_{IN} = V_{DD}$ $V_{DD} = 3.6V$			10	µA
V_{OL}	Low level output voltage		$V_{DD} = 3.0V$				
		(O2)	$I_{OL} = 2mA$			0.4	V
		(O4)	$I_{OL} = 4mA$			0.4	V
		(O6)	$I_{OL} = 6mA$			0.4	V
		(O8)	$I_{OL} = 8mA$			0.4	V
V_{OH}	High level output voltage		$V_{DD} = 3.0V$				
		(O2)	$I_{OH} = -2mA$	2.4			V
		(O4)	$I_{OH} = -4mA$	2.4			V
		(O6)	$I_{OH} = -6mA$	2.4			V
		(O8)	$I_{OH} = -8mA$	2.4			V
I_{OZ}	High impedance output leakage current		$V_O = V_{SS}$ or V_{DD} $V_{DD} = 3.6V$	-10			µA
I_{DD}	Quiescent supply current					300	µA
C_{IN}	Input capacitance		$V_{DD} = 3.3V$		2.4		pF
C_{OUT}	Output capacitance		$V_{DD} = 3.3V$		5.6		pF
$C_{I/O}$	I/O capacitance		$V_{DD} = 3.3V$		6.6		pF
P_A	Power dissipation		$V_{DD} = 3.6V$		0.5	1.0	W

I=input, SI=schmitt input; O=output; I/O=bidirectional; OD=open drain output

Figure 47. DC electrical characteristics

22 Timing Specifications

Note: The 7751 is PCI 2.1 compliant and meets all timing parameters. Please refer to the PCI 2.1 specification for the complete timing parameter specification.

Symbol	Parameter	Conditions*
C_{L1}	Output load on Context RAM Interface – DRAM	5pF min., 40 pF max.
C_{L2}	Output load on Context RAM Interface - SRAM	5pF min., 40 pF max.
C_{L3}	Output load on all other pins	50 pF
V_{DD}	Supply voltage	3.3V ± 5%
V_{SS}	Ground potential	0V
T_A	Ambient operating temperature	0°C to +70°C

*See derating information below for other load conditions.

Figure 48. AC specification definition

Signal	Pins	Derating (ns per 10pF)*
Context RAM data bus (high to low)	CDATA15-0	0.6ns per 10pF
Context RAM data bus (low to high)	CDATA15-0	0.35ns per 10pF
Context RAM address bus	CADDR19-0	0.28ns per 10pF

* These derating values represent typical case. For worst case derating, multiply these values by 1.75. For best case derating, multiply these values by 0.5.

Figure 49. AC specification load derating

Number	Description	Min	Max	Units
	Oscillator frequency		66.67	MHz
2	Clock width high	6		ns
3	Clock width low	6		ns
4	Clock rise time from V_{IL} to V_{IH}		5	ns
5	Clock fall time from V_{IH} to V_{IL}		5	ns

Figure 50. External PCLK clock with PLL disabled

Number	Description	Min	Max	Units
	Oscillator frequency		33.33	MHz
2	Clock width high	12		ns
3	Clock width low	12		ns
4	Clock rise time from V_{IL} to V_{IH}		5	ns
5	Clock fall time from V_{IH} to V_{IL}		5	ns

Figure 51. External PCLK clock with PLL enabled

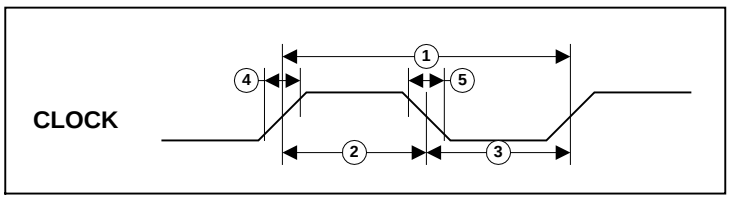


Figure 52. External clock (PCLK)

Number	Description	Min	Max	Units
1	Data valid after ADDR/LB/UB valid (access time)		t_{ECLK} – 18	ns
2	Data valid after OE# active		t_{ECLK}-18	ns
3	Address valid width (cycle time)	t_{ECLK} - 8		ns
4	Data hold after OE inactive	0	5	ns
5	LB#/UB# Access Time		t_{ECLK} - 18	ns
6	Output Enable time from LB#/UB#	3		ns
7	Output Disable time from LB#/UB#		6	ns
8	Output Enable time from OE#	0		ns
9	Output Data Hold time from ADDR change	0		ns

Figure 53. Compression SRAM Read Timing

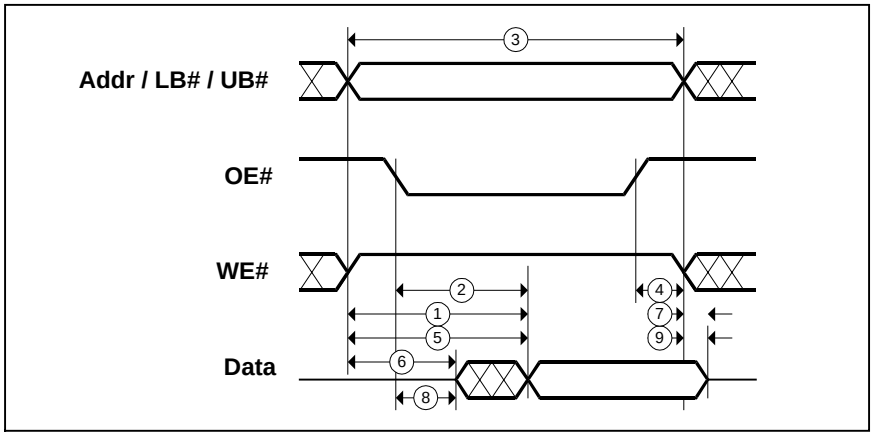


Figure 54. Compression SRAM Read Timing

Number	Description	Min	Max	Units
1	ADDR/LB/UB setup to WE# valid	0.5 t_{ECLK} – 9		ns
2	WE# active width	0.5 t_{ECLK} – 1		ns
3	ADDR hold after WE# inactive	2		ns
4	Data setup before WE# inactive	0.5 t_{ECLK} – 8		ns
5	Data hold after WE# inactive	1		ns
6	ADDR valid to WE# inactive	t_{ECLK} – 9		ns
7	Write Cycle time	t_{ECLK} – 8		ns
8	LB#/UB# to End of Write	t_{ECLK} - 8		ns

Figure 55. Compression SRAM Write Timing

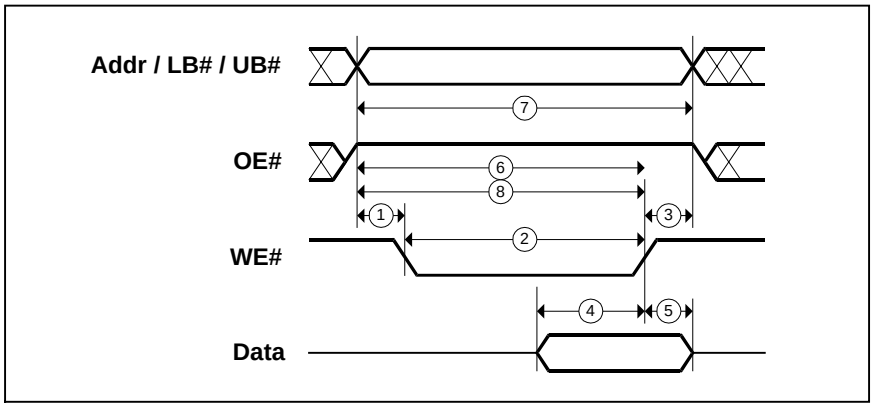


Figure 56. Compression SRAM Write Timing

Number	Description	Min	Max	Units
1	RAS# read / write cycle time	4 t_{ECLK}		ns
2	RAS# pulse width	2.5 t_{ECLK} - 2		ns
3	RAS# precharge time	1.5 t_{ECLK} - 5		ns
4	CAS# read / write cycle time	t_{ECLK}		ns
5	RAS# hold time from CAS# precharge	1.5 t_{ECLK}		ns
6	CAS# hold time	1.5 t_{ECLK} – 3		ns
7	RAS# to CAS# delay	t_{ECLK} – 3	1.5 t_{ECLK} + 1	ns
8	CAS# pulse width	0.5 t_{ECLK} – 3		ns
9	CAS# hpage precharge time	0.5 t_{ECLK}		ns
10	CAS# to RAS# precharge	1.5 t_{ECLK}		ns
11	CAS# precharge time	2.5 t_{ECLK} - 10		ns
12	CAS# to col. Address delay time	0.5 t_{ECLK} - 2	0.5 t_{ECLK} + 6	ns
13	Row address setup time	0.5 t_{ECLK} – 10		ns
14	Row address hold time	0.5 t_{ECLK} - 1		ns
15	Column address setup time	0.5 t_{ECLK} – 8		ns
16	Column address hold time	0.5 t_{ECLK} – 2		ns
17	Column address to RAS#	1.5 t_{ECLK} – 4		ns
18	Read command setup (read)	2 t_{ECLK} - 8		ns
19	Read command hold from RAS# (read)	1.5t_{ECLK}		ns
20	Read command hold from CAS# (read)	t_{ECLK} - 10		ns
21	Access time from CAS# precharge (read)		1.5t_{ECLK} – 10	ns
22	Read data access time from OE#		2t_{ECLK} - 30	ns
23	Read data access time from RAS#		2.5 t_{ECLK} - 15	ns
24	Read data access time from column address		1.5t_{ECLK} - 15	ns
25	Read data access time from CAS#		t_{ECLK} - 13	ns
26	Read data output hold time	0		ns
27	RAM buffer turn-off from OE# (read)	0	15	ns
28	WE# setup to CAS# (write)	0.5 t_{ECLK} – 6		ns
29	WE# hold from CAS# (write)	0.5 t_{ECLK} - 5		ns
30	Write data setup to CAS#	0.5 t_{ECLK} – 13		ns
31	Write data hold from CAS#	0.5 t_{ECLK} - 5		ns
32	Output buffer turn-off from WE# active (write)		15	ns
33	WE# Pulse Width	t_{ECLK} - 10		ns
34	RAS# hold after CAS#	0.5 t_{ECLK}		ns
35	RAM buffer turn off from RAS- (read)	0	15	ns

Figure 57. Compression DRAM Read and Write Timing

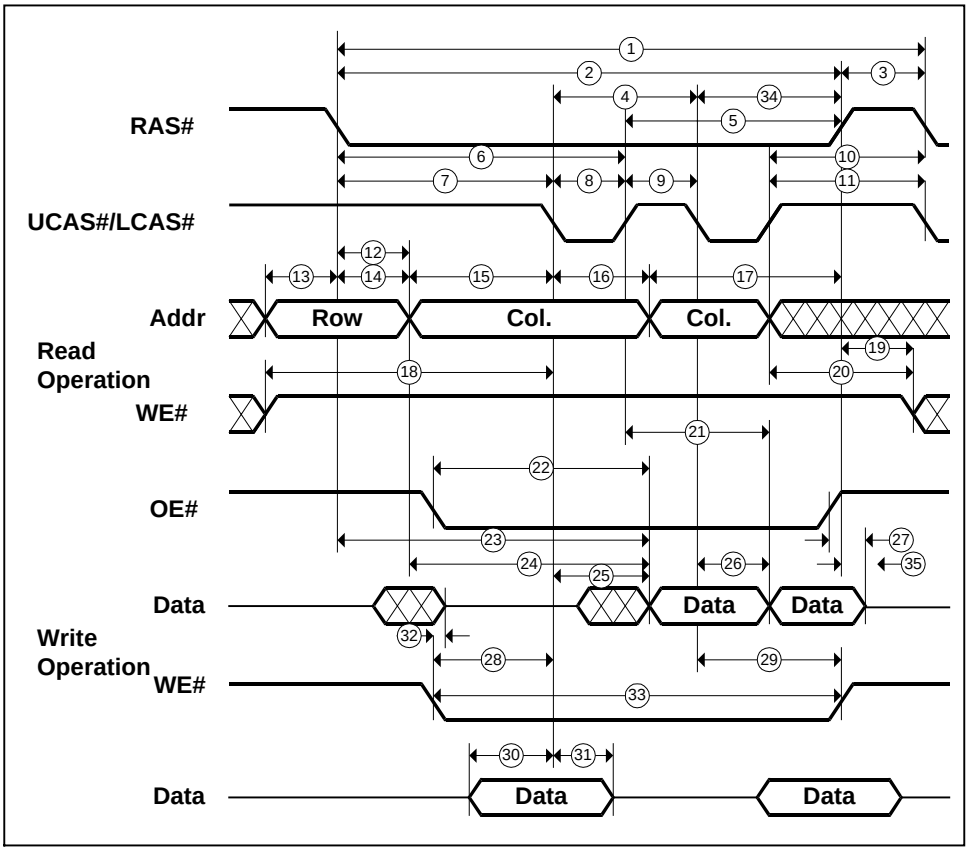


Figure 58. Compression DRAM Read and Write Timing

Number	Description	Min	Max	Units
1	CAS- setup to RAS-	0.5 t_{ECLK} - 10		ns
2	CAS- hold from RAS-	1.5 t_{ECLK} - 3		ns
3	RAS- presharge to CAS- hold	t_{ECLK} - 1		ns

* Note: All other relevant timing is referenced above.

Figure 59. Compression DRAM Refresh Timing

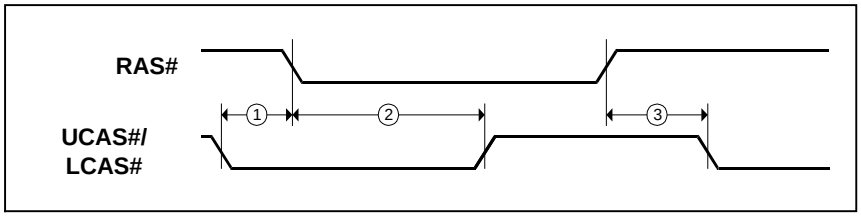


Figure 60. Compression DRAM Refresh Timing

Number	Description	Min	Max	Units
1	Fsk		64 t_{PCICLK}	MHz
2	Tskh	32 t_{PCICLK} – 5		ns
3	Tskl	32 t_{PCICLK} – 5		ns
4	Tsks	32 t_{PCICLK} – 5		ns
5	Tcs	64 t_{PCICLK} – 5		ns
6	Tcss	32 t_{PCICLK} – 10		ns
7	Tdh	0		ns
8	Tdis	32 t_{PCICLK} – 10		ns
9	Tcsh	0		ns
10	Tdih	31 t_{PCICLK} – 10		ns
11	Tpd0		31 t_{PCICLK} – 10	ns
12	Tpd1		31 t_{PCICLK} – 10	ns

Figure 61. EEPROM Timing

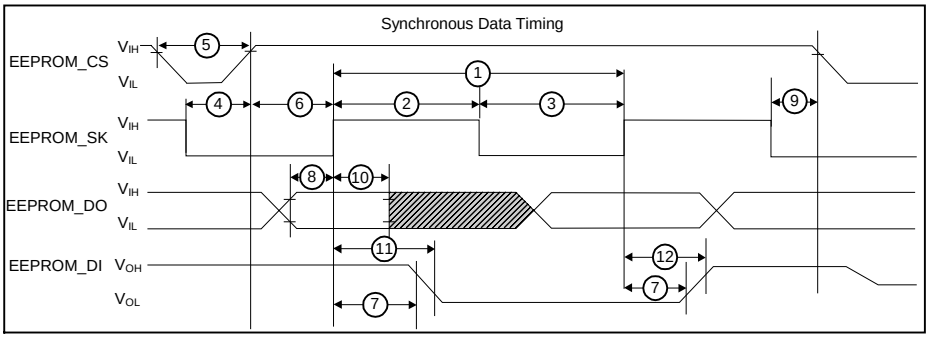


Figure 62. EEPROM Timing

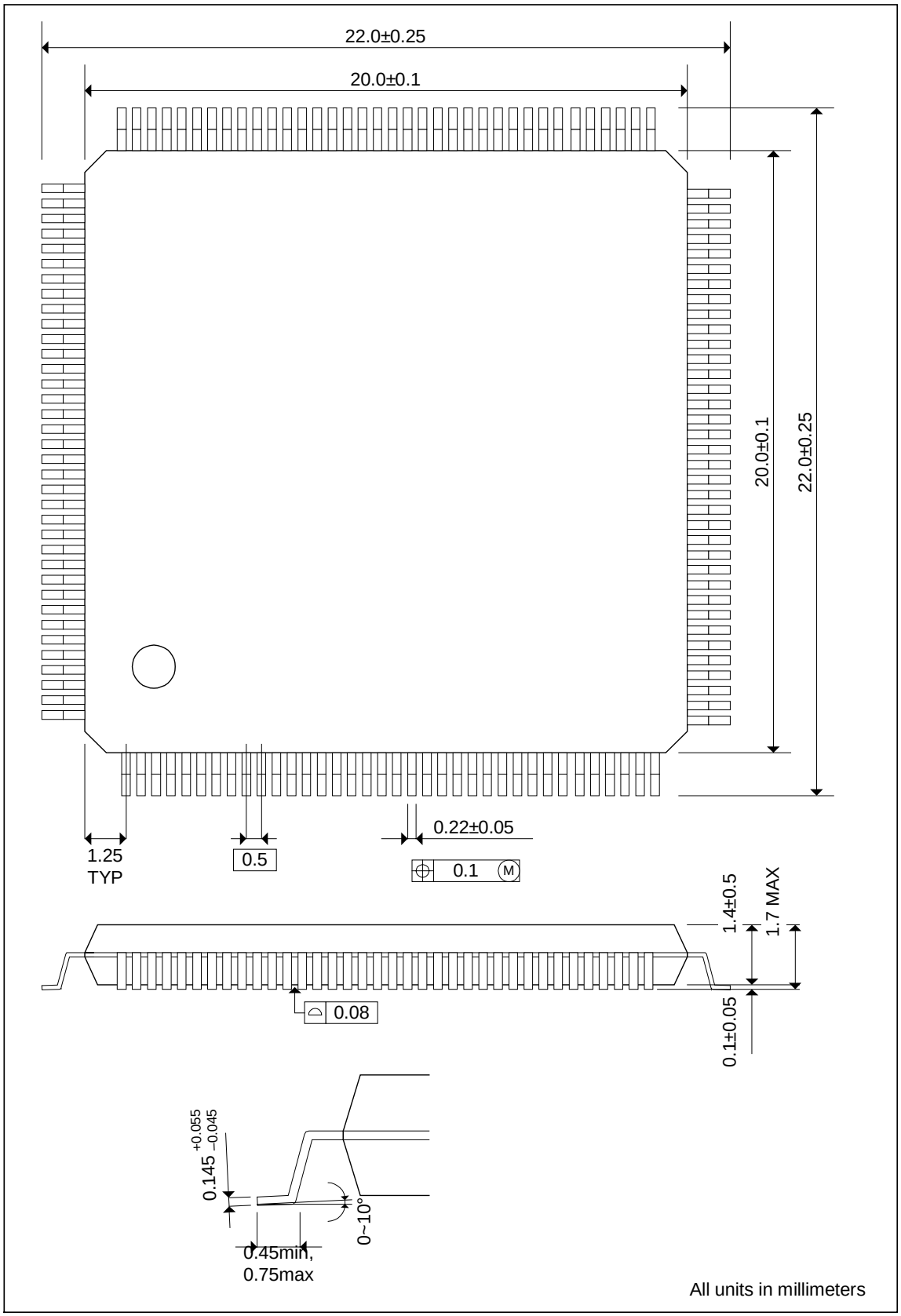


Figure 63. 144-pin TQFP package