DATA SHEET 7711 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® 7711 is a high-performance data encryption processor that may be used in a variety of data communications 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 7711 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 data compression ratio. Each individual DS0 line may be channelized with a separate encryption and compression context.

CPU intervention is not required between operations, eliminating any latency issues. Command parameters are embedded in the data stream, allowing for an unlimited amount of command queuing.

Features

- Supports DES, Triple-DES, RC4, LZS, MPPC, SHA-1, and MD5 algorithms
- No CPU intervention required between operations (no latency)
- Simple operation
- Supports Multiple sessions
- Single bus, dual-DMA slave interface
- 3.3V operation with 5V tolerant I/O's

Applications

- Data communication products
- Routers and Bridges
- Remote Access
- Mass storage products

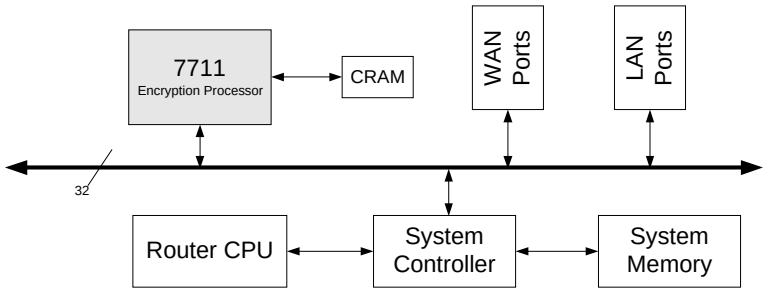


Figure 1. Example System Concept

2 7711 Engine Performance

Figure 2 summarizes the analysis of the individual 7711 engine performance. Performance of 7711 when multiple engines are used (e.g. the compression, MAC, and encryption engines are all engaged) can be approximated by considering the 7711 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 synthesizing 1500 byte packets through the individual engines.

	Speed (Mbps)	
Engine	**Encode**	**Decode**
LZS	64	117
MPPC	52	98
SHA HMAC	84	84
MD5 HMAC	100	100
DES	245	245
3-DES	82	82
RC4	129	129

Figure 2. Individual Engine Performance

Figure 3 shows the results of 7711 packet processing performance analysis for IPSec and PPTP.

		Speed (Mbps)	
Protocol	**Engine**	**Encode**	**Decode**
IPSec	LZS/SHA HMAC/DES	62	112
	LZS/SHA HMAC/3DES	62	112
PPTP	MPPC/RC4	50	96

Figure 3. Packet processing speed

3 Product Overview

3.1 Command and Data Pipeline

Commands, context (keys), and data are all transferred to the chip through one DMA channel. As a command progresses, data and result information are transferred from the chip with a second DMA channel. Optionally, the same information may be transferred to or from the chip via programmable I/O.

Commands, context, results, and data are all formatted into structures that are passed to the chip or read from the chip via the DMA interface. The formats of these structures are defined later in this document.

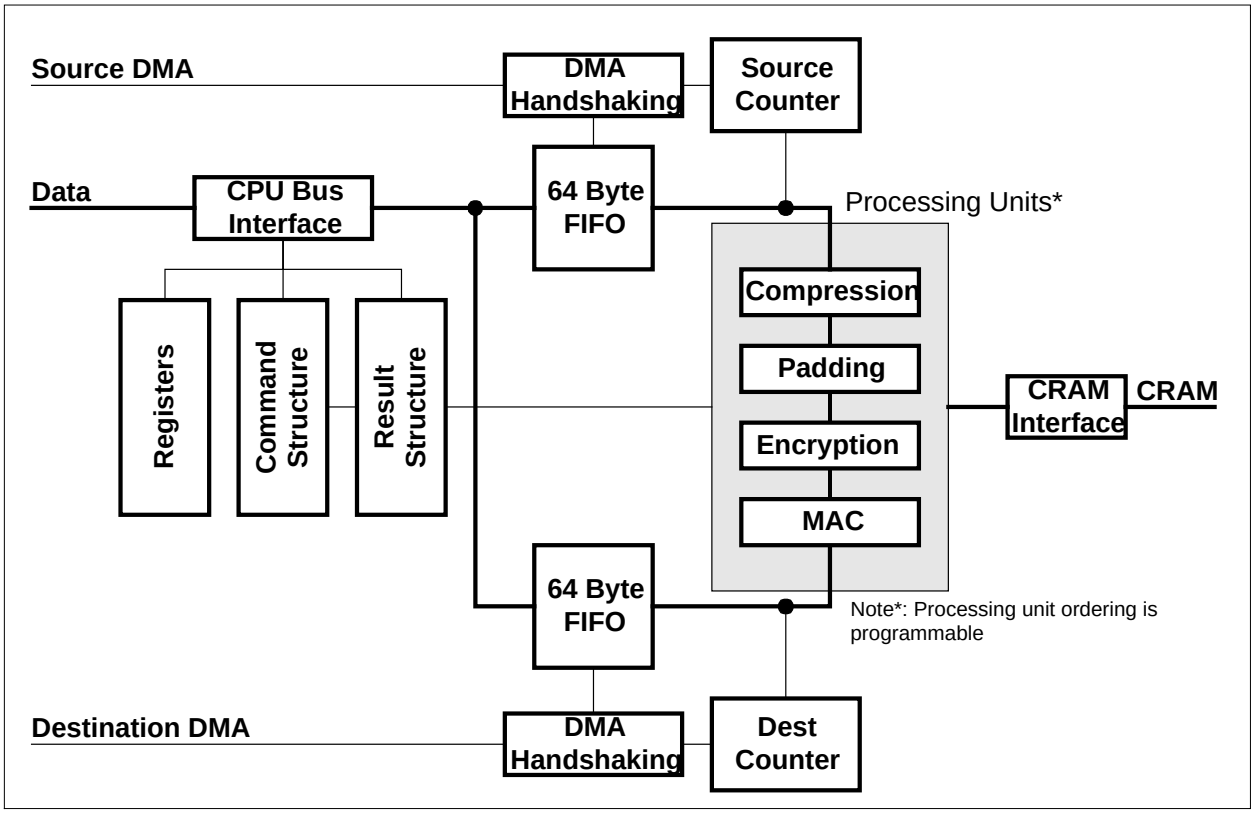


Figure 4. Internal Block Diagram

Although details of the structures are provided later in this document, it is important to understand the context structure and what a context is. Context represents the current state of all four processing units: compression, padding, MAC, and encryption. Each unique communication link normally requires a separate encryption key for security purposes, and therefore a separate state (or context).

All context information is stored in local RAM, called the Context RAM. This RAM is attached directly to the 7711. A context structure sent to the 7711 will contain key or IV information, such as a DES encryption key, or a new

Initialization Vector (IV). New key (and possibly IV) information will need to be sent to the device at the beginning of each session number.

After a hardware or software reset, the chip enters its disabled state. The Enable DMA control mode must be set in the Control register to enable the DMA interface. If programmed I/O is to be used instead of DMA, then setting the Enable DMA control mode is not necessary. Note that programmed I/O and DMA operations may not be used to service the same FIFO. A reset must be issued between servicing either the Source or Dest FIFO with programmed I/O and DMA transfers.

3.1.1 Source FIFO

Initially, the Source FIFO will be in the *command phase*. The Source FIFO will accept a command structure.

After a command structure has been written, the Source FIFO enters the *context phase*. The Source FIFO will accept a context structure which contains key or IV information. After the context information has been written, the Source FIFO enters the *data phase*.

In some cases, the context phase will be skipped. This will occur if no new key or IV is required.. If the context phase is skipped, the Source FIFO will move immediately to the data phase from the command phase.

While in the data phase, the Source FIFO will accept a data structure. After the data phase, the Source FIFO will again enter the command phase, and the whole process repeats itself indefinitely.

In some rare cases, the data phase will be skipped. Generally, this will occur during a Write RAM command. If the data phase is skipped, the Source FIFO will move immediately to the command phase from the previous phase (command or context).

3.1.2 Destination FIFO

Initially, the Destination FIFO will be in the *data phase*. The Destination FIFO will produce a data structure.

After the data phase, the FIFO enters the *result phase*. While in the result phase, the Destination FIFO will produce a result structure. After the result structure has been transferred from the Destination FIFO, the FIFO will again enter the data phase, and the whole process repeats itself indefinitely.

3.1.3 Processing Structure

Commands, keys , and data reside in system memory and are transferred to and from the 7711. There are many possible methods for organizing the commands and data in system memory. Two examples are shown in Figure 5.

Note: Although keys are not shown in the two examples, this information may be stored in system memory in similar ways.

Example 1 demonstrates commands and data presented sequentially in memory. In this example, a single DMA transfer could be used to transfer both the command

structure and data structure to the chip as a single operation. In fact, multiple commands may be issued in a single DMA operation.

Example 2 demonstrates the segregation of command structures from the data structures. The external DMA controller will switch between the two based either on its own DMA counters, or based on the Terminal Count signals residing on the 7711.

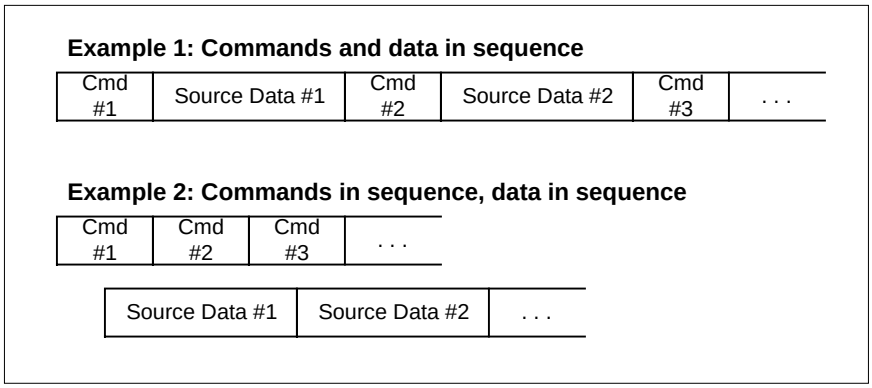


Figure 5. Commands and Data in System Memory

The organization of commands and data may be arbitrarily complex based solely on the capabilities of the external DMA controller.

Commands, context, or data may also be transferred through programmed I/O to the Data register.

The 7711 consists of four processing units: Compression, Padding, MAC, and Encryption. The order of processing is determined by the operation (encoding or decoding) as well as other programmable options.

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) before it begins processing the data. This counter is useful to skip headers 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 reaches zero, the command will complete, and the next command will begin to be processed.

3.2 Data Blocks

The 7711 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 Cnt and Source Cnt values in each processing unit, relatively complex patterns of compression, encryption, and authentication may be performed.

3.3 Example Data Block

The following simple example will help demonstrate how the 7711 processes a block of data. For a more detailed example, please see the "7711 Design Guide."

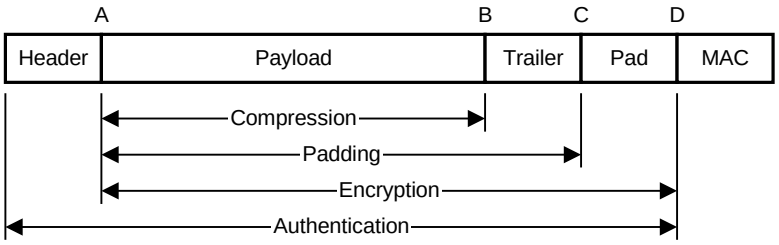


Figure 6. Example Data Block

Consider an encode operation being performed on the block of data shown in Figure 6. 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 7711 (the padding and MAC are added to the data by the 7711). As such, the total source count value is "C", as shown in Figure 7.

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 without compressing. Once the Header Cnt is passed through, the compression processing unit will compress Source Cnt bytes, "B-A". Note that "B", as a numerical value, will change after the compression operation.

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.

Figure 7 summarizes the count values for the different processing units in this example.

Command Structure Fields	Command
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 7. Command Length

3.4 Source FIFO Data Flow

The Source FIFO requests data transfers by asserting the SDREQ# signal and by setting the SOURCE FIFO READY bit in the Status register to one.

Although the Source FIFO is only 64 bytes in size, more than 64 bytes could be transferred into the Source FIFO in one burst because the Processing Unit is reading data from the Source FIFO at the same time, making room for additional data.

Once the FIFO becomes full, the Source FIFO will stop requesting data. The Source FIFO will request data again once the number of bytes of empty space in the Source FIFO is greater than the SOURCE FIFO THRESHOLD value set in the FIFO Configuration register.

The STC# signal may be programmed to be asserted during the last data transfer of the data phase, or after the last data transfer of each phase (command, context, and data). Figure 8 shows how STC# is asserted for encode and decode operations for each setting of TC ALL PHASES bit in the Configuration register. For a data flow description during Read RAM or Write RAM operations, refer to sections 7.2 and 7.3.

At the end of each phase (when TOTAL SOURCE COUNT reaches zero), any trailing bytes that are not 32-bit (or 16-bit) aligned will be discarded. The first byte of data for the next phase will come from the next 32-bit (or 16-bit) aligned value.

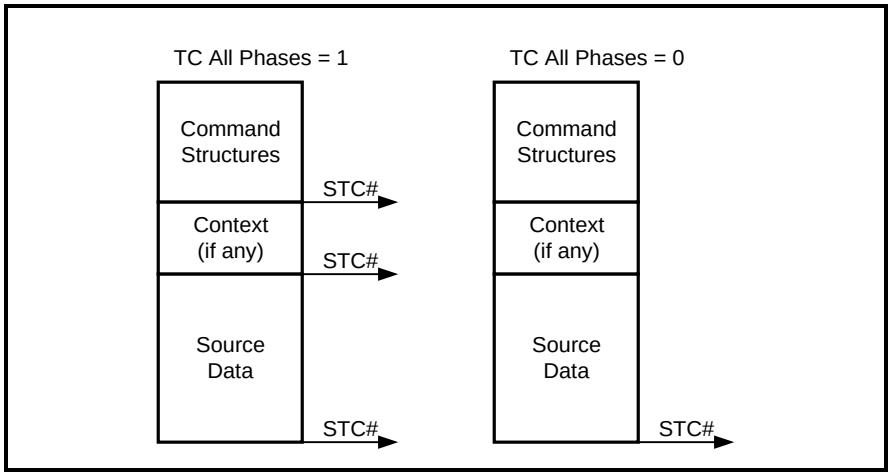


Figure 8. STC# assertion

3.5 Dest FIFO Data Flow

The Dest FIFO will not request any data transfer unless there is data available in the Dest FIFO. After data enters the Dest FIFO it will begin requesting data transfers once the number of bytes residing in the Dest FIFO is greater than the DEST FIFO THRESHOLD value set in the FIFO Configuration register. The request will remain active until the Dest FIFO becomes empty.

The Dest FIFO requests data transfers by asserting the DDREQ# signal and by setting the DEST FIFO READY bit in the Status register to one.

Although the Dest FIFO is only 64 bytes in size, more than 64 bytes could be transferred out of the Dest FIFO in one burst because the Processing Unit is writing data to the Dest FIFO at the same time, adding additional data.

Once the Dest FIFO becomes empty, the Dest FIFO will stop requesting data transfers. It will request data again once the number of bytes residing in the Dest FIFO is greater than the value set in the FIFO Configuration register.

After the last byte of data from the result phase has entered the Dest FIFO, the Dest FIFO will request data transfers, independent of the FIFO threshold set by the FIFO Configuration register, until this byte is transferred out of the Dest FIFO.

The DTC# may be programmed to be asserted during the last data transfer of the result phase, or after the last data transfer of each phase (data and result). When the last byte of data enters the Dest FIFO from any phase, the data is padded with data to fill out an entire 32-bit (or 16-bit) value. The value of these bytes is undefined. The first byte of data from the next phase will be 32-bit (or 16-bit) aligned. Figure 9 shows how DTC# is asserted for encode and decode operations for each setting of TC ALL PHASES bit in the Configuration register. For a data flow description during Read RAM or Write RAM operations, refer to sections 7.2 and 7.3.

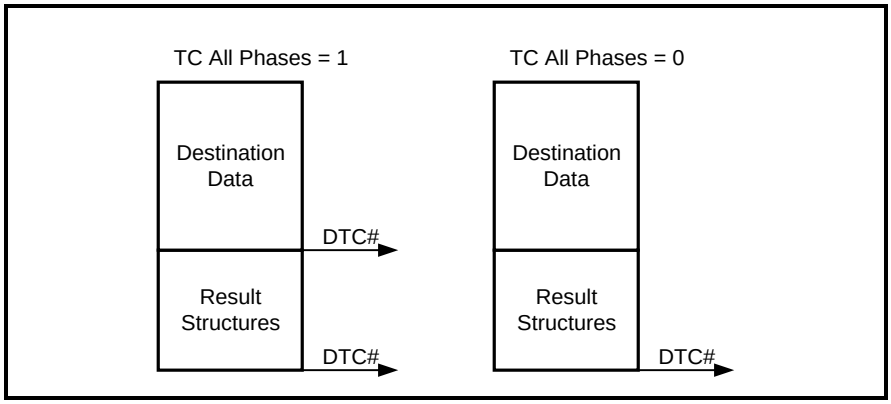


Figure 9. DTC# assertion

3.6 Destination Overrun

A 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 Status register will be set. Additionally, the DEST OVERRUN bit in the Base Result Structure will be set when the results are produced. Furthermore, the DEST OVERRUN INTERRUPT ENABLE bit in the Interrupt Enable register may be used to generate an interrupt when a destination overrun occurs. And, the DEST OVERRUN INTERRUPT STATUS in the Interrupt Status register may be used to detect a destination overrun condition.

Note that the DEST OVERRUN bit will only be set momentarily in the Status Register, and may not be detectable. However, the DEST OVERRUN bit in the Interrupt Status Register will remain set until cleared. Clearing a bit in the Interrupt Status Register is described in the Interrupt Status Register description below.

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.

3.6.1 Destination Overrun affect 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 7711 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 7711 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.

3.7 Context RAM

The 7711 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 7711 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 7711 supports a maximum of 32 Mbytes of EDO DRAM.

The 7711 encoding and decoding performance is affected by the setting of the DRAM bit due to the memory speed differences. Figure 10 shows the difference in 7711 performance with SRAM and DRAM.

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

Figure 10. 7711 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.

3.8 Bus Mode

The bus mode is determined by the BUS MODE (1-0) signals. These pins select the timing mode of the CPU interface. See Figure 11 for a summary of the bus modes. Refer to the AC Timing section for timing details. These pins must be tied high or low throughout the entire operation of the 7711.

BMODE1-0	Bus type	R/W polarity	If DS# not used
0 0	Asynchronous	R/W#	See AC Timing section
0 1	Asynchronous (BCLK=PCLK / 2)	R/W#	See AC Timing section
1 0	Synchronous	R#/W	Tie high
1 1	Motorola 68360	R/W#	Tie high or low

Figure 11. BUS Modes and DS# Usage

3.9 Clock Interface

There are three clock domains in the 7711: PCLK, BCLK, and ECLK. The PCLK and BCLK signals are external inputs, and ECLK is an internal signal generated from PCLK. The relationship between these three clocks is shown in Figure 12.

PCLK is an input which is used to drive the internal ECLK signal. In addition, the PCLK signal drives BCLK when the 7711 is in asynchronous (BCLK=PCLK/2) mode. The maximum input frequency for the PCLK signal is 66MHz if the PLL is disabled, and 33MHz if the PLL is enabled. See Figure 13 for a summary of how the PLL, PCLK, and ECLK are related.

BCLK is an input which is used to drive the bus interface in all bus modes except asynchronous (BCLK=PCLK/2). The maximum input frequency for the BCLK signal is 33MHz. The BCLK signal operates independently of PLCK, and is described in the BCLK pin description.

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 PCLK, and has a maximum frequency of 33MHz. See Figure 12 and Figure 13 for a summary of how PCLK and ECLK are related.

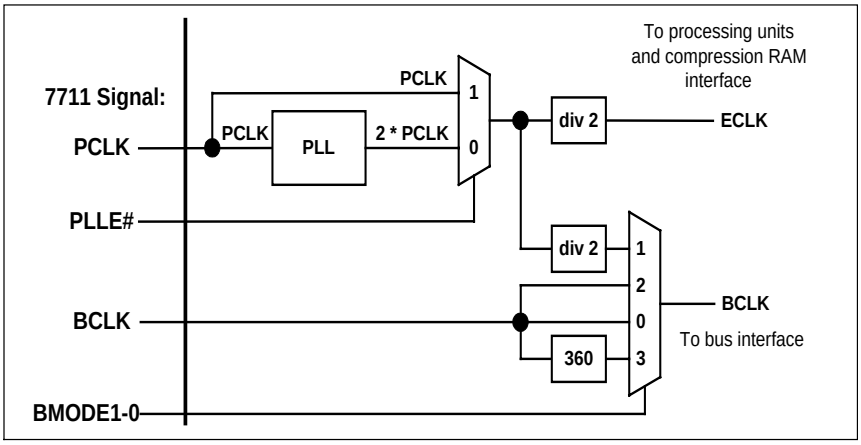


Figure 12. Clock relationships

PLL	PCLK	ECLK
Disabled	0-66MHz	PCLK / 2 (0-33MHz)
Enabled	25-33MHz	PCLK (25-33)

Figure 13. PLL/PCLK/ECLK relationship

3.10 PLL

The PLL is used to double the input value of the PCLK signal. This is useful if a 33MHz clock signal is readily available, and peak 7711 performance is desired. Peak 7711 performance will occur when the PCLK input is 66MHz with the PLL disabled, or 33MHz with the PLL enabled.

3.10.1 Disabled

When the PLL is disabled, the ECLK value will be one-half of the PCLK value. Disabling the PLL is done by connecting the PLL pins as shown in Figure 14. If the PLL is disabled, the 7711 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 7711 PLL components.

Figure 14. PLL disable

3.10.2 Enabled

If the 7711 PLL is used, several external components are required. The schematic of these external components is shown in Figure 16, with the appropriate component values shown in Figure 15.

Discrete	Function	Value	Symbol
R1	Fin@25MHz	160	KΩ
R1	Fin@33MHz	36	KΩ
C1	PLL filter	1500	pF
C2	Decoupling	0.1	µF
C3	Decoupling	10	µF

Figure 15. PLL external components

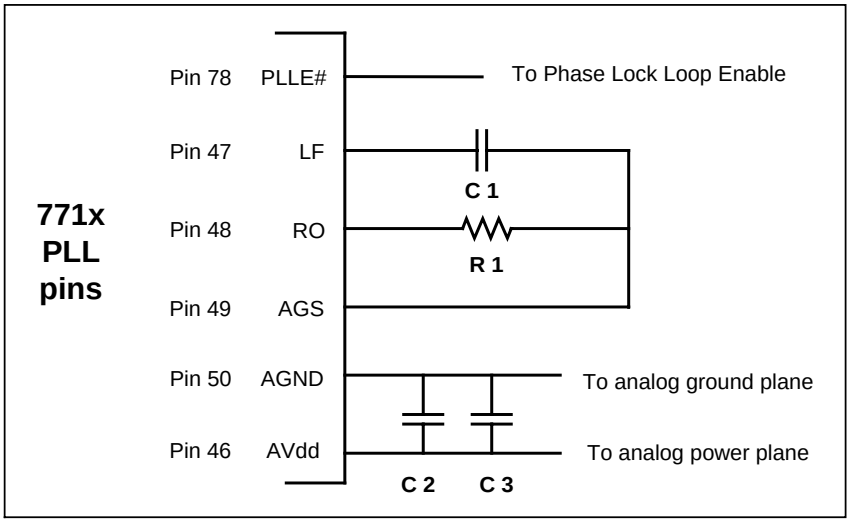


Figure 16. PLL pin connection

If the PLL is enabled, the input range for the PCLK signal must be between 25MHz and 33MHz. Since PCLK drives ECLK (which drives the processing units), 7711 performance will scale with the value of PCLK.

For more information on using the internal phase lock loop, please see the Hi/fn 7711 Design Guide.

3.11 Endianness

The BIG ENDIAN bit in the 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.

3.11.1 Command and Result Structures

The 7711 BIG ENDIAN bit operates on the Command and Result Structures in a 16-bit (word) level. If the device is configured for 16-bit operation, the BIG ENDIAN bit will not affect the Command and Result Structures.

While configured for 32-bit operation, the BIG ENDIAN bit affects the Command and Result Structures in the following way:

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 18 represents the first word of the Command/Result Structure, and word "B" represents the second word of the Structure.

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 18 represents the first word of the Command /Result Structure, and word "A" represents the second word of the Structure.

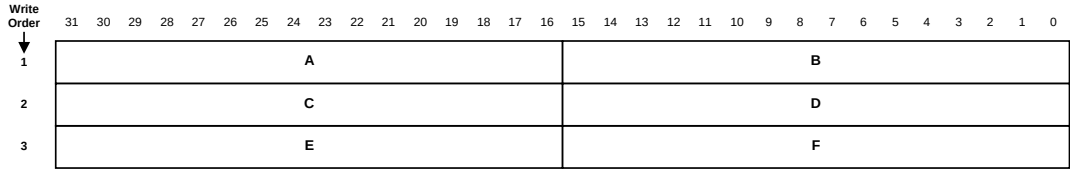


Figure 17. 32-bit Mode Command/Result Structure Words

3.11.2 Source Context, Source Data, and Dest Data

The 7711 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 18) will enter the engine as the first byte, and the byte residing on bits 7-0 (byte D in Figure 18) 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 18) will enter the engine first, and the byte residing on bits 7-0 (byte D in Figure 18) will enter the engine second.

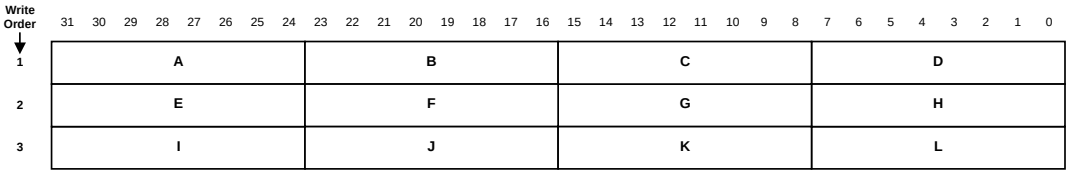


Figure 18. 32-bit Mode Source Context / Data Structure bytes

Little Endian

When configured for 32-bit little endian Mode, the byte residing on bits 7-0 (byte D in Figure 18) will enter the engine as the first byte, and the byte residing on bits 31-24 (byte A in Figure 18) 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 18) will enter the engine as the first byte, and the byte residing on bits 15-8 (byte C in Figure 18) will enter the engine as the second byte..

4 Pin Descriptions

Pin Number	Signal	Type	Description
CPU Interface			
21-20, 17-14, 12-7, 5, 143-139, 137, 134-128, 126, 124-120	DATA31-0	I/O8	**32-bit data bus . (16-bit option)** Bi-directional data bus for the CPU Interface. This bus may be configured for either 16-bits or 32-bits with the BUS WIDTH bit in the Configuration Register. After a hardware reset, the bus will be configured for 16-bit mode. While configured for 16-bit mode, the upper 16 bits (D31-D16) will remain tri-stated. It is not necessary to tie the unused data bus signals high or low as they are terminated internally with high impedance pull-ups.
3-1	ADDR2-0	I	**Register select** Address input signals for the CPU Interface. These inputs are significant only for Register accesses, not for DMA accesses.
30	R/W	I	**Read/Write** This input signal determines the direction of the data bus during a register transfer. The polarity of this signal is selected by the state of the BMODE (Bus Mode) signals.
31	CS#	I	**Command strobe** Active low input. While this signal is active, a register access will take place. The data direction is determined by the R/W signal.
37	IRQ#	OD2	**Interrupt output** Active low output. This signal will become active when any event occurs that is enabled in the Configuration register. See the Interrupt Enable register description for further information about when this signal is asserted and deasserted. This signal is an open drain output requiring an external pull-up resistor to V_{DD}
32	DS#	I	**Data strobe** This input pin is used as additional control information for bus timing. The operation of this signal is selected by the setting of the BMODE (Bus Mode) signals. While in asynchronous modes (BMODE 00 or 01), the DS# signal must be asserted while the CS# signal is asserted for CPU accesses, and it must be asserted while the SDACK# and DDACK# signals are asserted for DMA accesses. See the AC Timing section for timing details. When using a Motorola 68302, the DS# signal may be tied to the DS# signal of the 68302. While in synchronous mode (BMODE 10), the DS# signal may be asserted as shown in the AC Timing section. Otherwise, the DS signal must be tied high. While in 68360 mode (BMODE 11), the DS# signal must be tied high or low.
DMA Interface			
39	SDREQ#	O4	**Source FIFO DMA request** Active low output. This signal will become active when the Source FIFO is ready to accept a DMA transfer. See the *Source FIFO Data Flow* section for more information. This signal will become inactive when the Source FIFO becomes full.
41	SDACK#	I	**Source FIFO DMA acknowledge** Active low input. While this signal is active, a DMA write operation will take place to the Source FIFO.

Pin Number	Signal	Type	Description
38	STC#	OD / O4	**Source FIFO terminal count** Active low output. This signal can be either an open drain or a totem pole output, depending on the setting of the STC / DTC OUTPUT DRIVE bit in the Configuration register. The pin defaults to open drain after a reset. This signal will become active just before and during the last DMA transfer in a phase to the Source FIFO. This may occur only after the data phase, or after all phases (command, context, and data) depending on the setting of the TC ALL PHASES bit in the Configuration register. Further Source DMA activity will be for the next phase.
54	DDREQ#	O4	**Dest FIFO DMA request** Active low output. This signal will become active when the Dest FIFO is ready to accept a DMA transfer. See the *Dest FIFO Data Flow* section for more information. This signal will become inactive when the Dest FIFO becomes empty.
53	DDACK#	I	**Dest FIFO DMA acknowledge** Active low input. While this signal is active, a DMA read operation will take place from the Dest FIFO.
55	DTC#	OD / O4	**Dest FIFO terminal count** Active low output. This signal can be either an open drain or a totem pole output, depending on the setting of the STC/DTC OUTPUT DRIVE bit in the Configuration register. The pin defaults to open drain after a reset. This signal will become active just before and during the last DMA transfer in a phase from the Dest FIFO. This may occur only after the result phase, or after all phases (data, context, and result) depending on the setting of the TC ALL PHASES bit in the Configuration register. Any further Dest DMA activity will be from the next phase.
Context RAM			
43, 22, 107-102, 100, 98, 96-93, 91, 89-85	CADDR19-0	O8	**Context RAM address** These output signals select the address of the Context RAM to be accessed. Either SRAM or DRAM may be used as set in the Configuration register. 7711 performance will be better when using SRAM than it will be with DRAM. Figure 31 and Figure 32 show the performance difference between SRAM and DRAM. If SRAM is used, these 20 address signals may be tied directly to the address bus of the Context RAM. For maximum 7711 performance, at least 10ns SRAM must be used. If DRAM is used, CADDR11-CADDR0 are attached to the multiplexed address bus of the Context RAM. CADDR12 becomes the RAS# signal. CADDR13 becomes the LCAS# signal. CADDR14 becomes the UCAS# signal. CADDR15 becomes the WE# signal. CADDR16 becomes the OE# signal. At least 60ns EDO DRAM must be used.
76, 74-65, 63-59	CDATA15-0	I / O8	**Context RAM data** 16-bit bi-directional data bus for the Context RAM. 16-bit Context RAM must be used.
109	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. The CADDR15 signal is used for the DRAM WE#.

Pin Number	Signal	Type	Description
110	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. The CADDR16 signal is used for the DRAM OE#.
113	LB#	O4	**Lower Byte enable** Active low output. This signal will become active during each access to the lower byte of the Context RAM if SRAM is used. This signal is not used for DRAM. It must be left unconnected.
111	UB#	O4	**Upper Byte enable** Active low output. This signal will become active during each access to the upper byte of the Context RAM if SRAM is used. This signal is not used for DRAM. It must be left unconnected.
PLL			
47	LF	A	**Loop filter** Analog pin that connects to the external PLL Filter. See the *PLL Section* for more information concerning the PLL and the external components required.
48	RO	A	**Resistor output** Analog pin that sets the VCO center frequency. See the *PLL Section* for more information concerning the PLL and the external components required.
49	AGS	A	**Analog ground sense** Analog pin that connects to the external PLL Filter. See the *PLL Section* for more information concerning the PLL and the external components required.
78	PLLE#	I	**PLL enable** Active low input. If this signal is high (inactive), the PLL will be disabled. The PCLK will drive the internal logic directly. If this signal is low (active), the PLL will be enabled. The PCLK frequency will be doubled before driving the internal engine logic.
46	AVdd		**Analog +3.3 volts**
50	AGND		**Analog ground**
Miscellaneous Signals			
26	RESET#	SI	**Reset** Active low input. While this signal is active, this chip will immediately stop any current activity and will go into a known state. After a hardware reset, each context should be cleared on its first use. Do not access the chip (except to read the Control register) after a RESET# is deasserted until after the RESET DONE bit in the Control register is set to zero. RESET# does not disable the output drivers of the data bus when the chip is enabled or selected via active assertion of CS# or DDACK#. As a result, if either of these two signals is asserted during reset, the data bus will be driven. Therefore, CS# and DDACK# should be pulled up or driven to an inactive logical high state during reset.
57-56	BMODE1-0	I	**Bus mode** These pins select the timing mode of the CPU interface. See the *Bus Mode* section for additional information on the bus modes.
28	BCLK	I	**Bus clock** This input drives the clock of the bus, which determines the speed of the bus. There are 4 modes of BCLK operation. See the *Bus Mode* and *Clock Interface* sections for additional information on the bus clock.

Pin Number	Signal	Type	Description
24	PCLK	I	**Processor clock** This input drives the ECLK signal. The ECLK signal drives the processing units and Context RAM interface clocks. The speed of the processing units is directly linear with the speed of ECLK. See the *Clock Interface* section for details on PCLK.
33-36, 80-82, 115, 117-118	NC		**No connection(1)**
6, 18, 19, 25, 29, 42, 51, 52, 77, 83, 90, 99, 114, 125, 138	Vdd		**+3.3 volts**
4, 13, 23, 27, 40, 44, 45, 58, 64, 75, 79, 84, 92, 97, 101, 108, 112, 116, 119, 127, 135, 136, 144	GND		**Ground**

I=input, SI=schmitt input; O=output; I/O=bidirectional; OD=open drain output, 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, O6 implies a 6mA output drive capability, etc.
Note 1. Pins labeled as "No connection" must not have anything connected to them.

Figure 19 . Pin Connections

5 Register Descriptions

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.

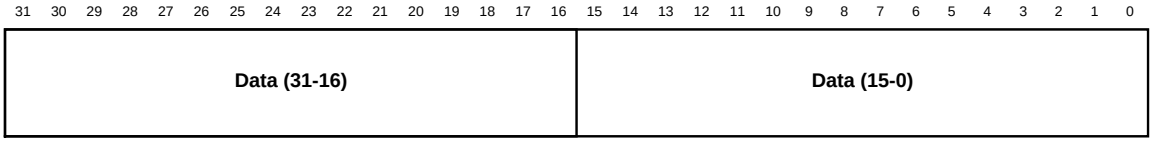
5.1 Overview

The 7711 uses 8 address locations for configuration and use. All 7711 registers may be accessed as either 16-bit or 32-bit registers, as defined by the BUS WIDTH bit in the Configuration register. Reserved bits must be written as zeros and ignored when read.

Name	Address
Data	0
Control	1
Interrupt Status	2
Configuration	3
Interrupt Enable	4
Status	5
FIFO Status	6
FIFO Configuration	7

Figure 20. Register list

5.2 Data (0)



This register can be read or written. It is used to transfer data to the Source FIFO or from the Dest FIFO. See the BUS WIDTH bit in the Configuration register for a description of the data bus width. If configured for a 32-bit bus, four bytes (32 bits) will be transferred at each access. If configured for a 16-bit bus, two bytes (16 bits) will be transferred at each access. See the BIG ENDIAN bit in the Configuration register for a description of the byte ordering.

A CPU bus write operation will transfer four (or two) bytes to the Source FIFO, depending on the data bus width. A CPU bus read operation will transfer four (or two) bytes from the Dest FIFO.

Normally, the DMA interface is used to transfer data to and from the 7711. However, the Data register may be used if desired.

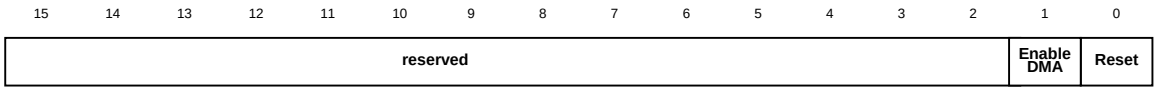
The Data register may be used to transfer commands, data, results, or context. Data for different phases may be transferred via either DMA or the Data register. As an example, the data phase may be transferred via DMA, and the command, result, and context phases may be transferred via the Data register.

Transfers to or from the Data register may only take place under the same conditions that a DMA transfer may take place. This is described in the *Source FIFO Data Flow* and *Dest FIFO Data Flow* sections. Use the SOURCE FIFO READY and DEST FIFO READY bits in the Status register (instead of the SDREQ or DDREQ pins) to determine when it is valid to read or write to the Data register.

The Data register may be used even while the Disable DMA mode is set (in the Control register).

If too much data is written (Source FIFO overflow), or too much data is read (Destination FIFO underflow), the command will terminate, and the DATA ERROR bit in the Status register will be set to one.

5.3 Control (1)



This register may be read or written. It is used to control the overall operation of the chip.

5.3.1 Enable DMA

Setting this bit will cause the DMA to become enabled. DMA operation will commence. The DMA mode may only be changed while there are no active DMA transfers occurring. The DMA mode will not affect the Data register.

The value of this bit will always be the last value written to this bit. After a hardware or software reset, this bit will be zero.

5.3.2 Reset

Setting this bit to one is identical to asserting and deasserting the hardware reset pin, except the Configuration register will not be affected. Also, as a result, if the Context RAM is locked, it will remain locked.

When this bit is set to one, all chip operations will immediately stop. The Source and Dest FIFOs will be cleared. Any intermediate information still residing in the chip 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.

Do not access the 7711 (except to read the 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 7711 is ready for normal operation. After it returns to zero, all registers may be accessed normally.

5.4 Interrupt Status (2)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Invalid Cmnd Int Stat	Data Error Int Stat	Src FIFO Rdy Int Stat	Dst FIFO Rdy Int Stat	reserved				Source Comnd Int Stat	Source Context Int Stat	Source Data Int Stat	Dest Data Int Stat	reserved	Dest Result Int Stat	reserved	

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 Status register becomes a one. When the corresponding 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 Interrupt Status register.
The relationship of the Status register bits, the Interrupt Status register bits and the Interrupt Enable register bits is shown in Figure 21.

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

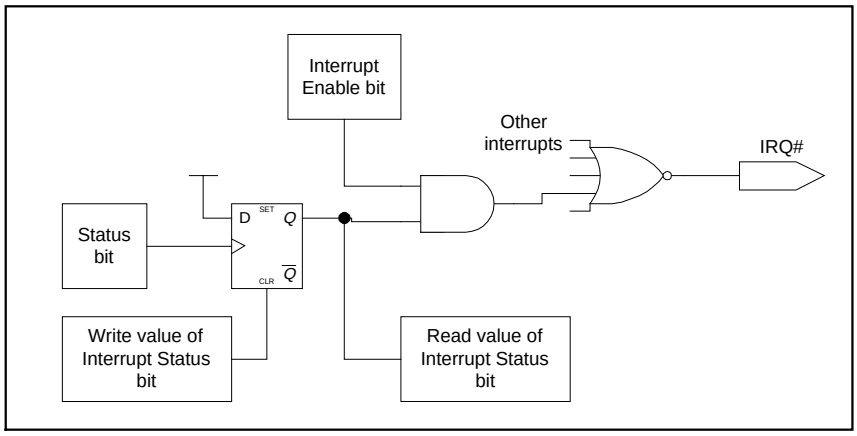


Figure 21. Interrupt Logic

5.5 Configuration (3)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRAM Size			DRAM Refresh Rate		DREQ Mode	TC All Phases	TC Drive	Big Endian	Bus Width	Chip ID	DRAM	reserved	Comp Config	Encrypt Config	reserved

This register may be read or written and is used to configure chip options. The 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 Control register to one)

5.5.1 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 22. EDO DRAM size values

5.5.2 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 23. Refresh frequency

5.5.3 DREQ Mode

This bit selects an alternate mode of bus operation for the SDREQ#, DDREQ#, STC#, and DTC# signals, which is described in "7711 DREQ Mode Timing Application Note" (AN-0008-00). This bus timing is based on the Synchronous bus timing. The DREQ MODE bit will not affect the other bus signals.

When the DREQ MODE bit is set to zero, the SDREQ#, DDREQ#, STC#, and DTC# signals will operate in Synchronous mode as defined in the *Source FIFO Data Flow* and *Dest FIFO Data Flow* sections.

When the DREQ MODE bit is set to one, the BMODE1-0 signals must be set to [1,0], that is, the 7711 bus must be set to Synchronous mode timing. The SDREQ# and DDREQ# signals are asserted as described in the *Source FIFO Data Flow* and *Dest FIFO Data Flow* sections. However, they will be deasserted as described below.

The SDREQ# signal will deassert when the number of bytes of empty space in the Source FIFO is less than or equal to the threshold programmed in the FIFO Configuration register.

The DDREQ# signal will deassert when the number of bytes in the Dest FIFO is less than or equal to the threshold programmed in the FIFO Configuration register. The DDREQ# signal is also asserted after the last byte of data from the result phase enters the Dest FIFO, independent of the FIFO threshold. In this case it will deassert only when this byte leaves the Dest FIFO, and there is not another last byte of a command in the Dest FIFO.

This mode of operation is useful for fixed-length burst DMA operations. See Figure 19 for a summary of SDREQ# and DDREQ# behavior.

DREQ MODE = 0		
	Asserted	**Deasserted**
DDREQ#	# bytes > threshold	FIFO empty
	Last result byte in FIFO	FIFO empty
SDREQ#	Space > threshold	FIFO full
DREQ MODE = 1		
	Asserted	**Deasserted**
DDREQ#	# bytes > threshold	# bytes <= threshold
	Last result byte in FIFO	No last result byte in FIFO
SDREQ#	Space > threshold	Space <= threshold

Figure 24. xDREQ behavior

When the DREQ MODE bit is set to zero, the STC# and DTC# signals behave as described in this data sheet.

When the DREQ MODE bit is set to one, the STC# and DTC# signals behave as described in "7711 DREQ Mode Timing Application Note" (AN-0008-00).

5.5.4 TC All Phases

This bit determines when the Terminal Count (TC) signals will be asserted. If this bit is set to zero, the TC signal will be asserted to signal the end of the data phase for the Source FIFO, and the end of the result phase for the Destination FIFO.

If this bit is set to one, the TC signal will be asserted to signal the end of all phases (command, result, data, and context) for both Source and Destination FIFOs.

5.5.5 TC Drive

This bit determines whether the STC# and DTC# signals are driven with an open drain or a totem-pole drivers.

If this bit is 0, then the output drivers are open drain. If this bit is 1 then the output drivers are totem-pole.

5.5.6 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 7711 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 7711 data bus will enter or leave the Processing Unit first.

The big endian bit affects the data which enters the processing unit. Commands, context, and registers are affected by the endianness of the 7711, as described in section 3.11.

5.5.7 Bus Width

This bit selects the data bus width when performing DMA or accessing the Data register. If this bit is set to zero, a 16-bit bus is assumed. If this bit is set to one, a 32-bit bus is assumed.

The only register affected by the setting of this bit is the Data register.

5.5.8 ChipID

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

5.5.9 DRAM

This bit selects the RAM type used for the Context RAM. This bit must be set to zero if SRAM is used. This bit must be set to one if EDO DRAM is used. The 7711 encoding and decoding performance is affected by the setting of the DRAM bit due to the memory speed differences. Figure 20 shows the difference in 7711 performance with SRAM and DRAM.

Context RAM	7711	
	encode (Mbps)	decode (Mbps)
10ns SRAM	64	120
60ns EDO DRAM	20	40

Figure 25. 7711 Context RAM performance

5.5.10 Compression Configuration

This bit determines whether a 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 26 shows local Context RAM 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.

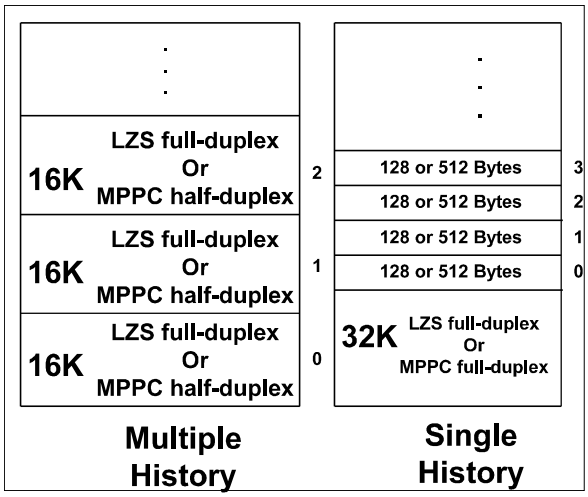


Figure 26. Local Context RAM map

5.5.11 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.

5.6 Interrupt Enable (4)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nva id Cmnd nt Enb	Data Error nt Enb	Src F FO Rdy nt Enb	Dst F FO Rdy nt Enb	reserved				Source Cmnd nt Enb	Source Context nt Enb	Source Data nt Enb	Dest Data nt Enb	reserved	Dest Resu t nt Enb	reserved	

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 Interrupts Status register affect the IRQ signal. If an Interrupt Enable register bit is set to zero, the corresponding Interrupt Status register bit will not affect the IRQ signal.

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

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

The relationship of the Status register bits, the Interrupt Status register bits, and the Interrupt Enable register bits is shown in Figure 21 which may be found in the Interrupt Status register description.

5.7 Status (5)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Invalid Cmnd	Data Error	Src FIFO Ready	Dst FIFO Ready	reserved				Source Comnd	Source Context	Source Data	Dest Data	reserved	Dest Result	reserved	

This register may only be read. 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 7711 is reset (either by a software reset or a hardware reset).

5.7.1 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.

5.7.2 Data Error

This bit is set to one if there is an attempt to write to the source FIFO when it is full, or if there is an attempt to read from the dest FIFO when it is empty. Either programmed I/O or DMA transfers may cause this bit to be set.

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.

5.7.3 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.

5.7.4 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.

5.7.5 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.

5.7.6 Dest Data, Dest Result

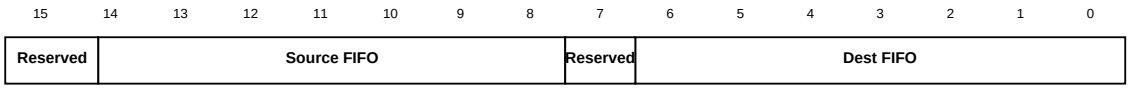
These two bits indicate the current state of the Destination FIFO: Data or Result phase. One and only one of these bits will be set at a time. The Destination FIFO state is relative to the output of the FIFO.

5.7.7 Chip ID

When the ChipID bit (bit 5) in the 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. The 7711 Chip ID value is $11XX_{16}$.

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 Configuration register must be set to one again.

5.8 FIFO Status (6)



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

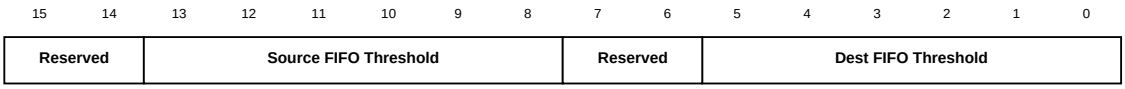
5.8.1 Source FIFO

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.

5.8.2 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.

5.9 FIFO Configuration (7)



This register may be read or written. It is used to configure the FIFO thresholds at which a FIFO will begin to request a data transfer. This threshold is used to determine when the FIFO ready status bits in the Status register will be set to 4one. This may optionally assert an interrupt if configured in the Configuration register.

The threshold is also used to determine when the DMA Interface logic will begin to request for a block of DMA transfers. See the *Product Overview* section for more information.

5.9.1 Source FIFO Threshold

This field sets the Source FIFO Threshold. When the number of bytes available to be written to the Source FIFO is greater than or equal to the value set in this field, the SOURCE FIFO READY bit in the Status register will be set to one. When the number of bytes available to be written to the Source FIFO is greater than the value set in this field, the SDREQ signal will become active. See the *Source FIFO Data Flow* section for more information regarding this field. The default value of this field is zero.

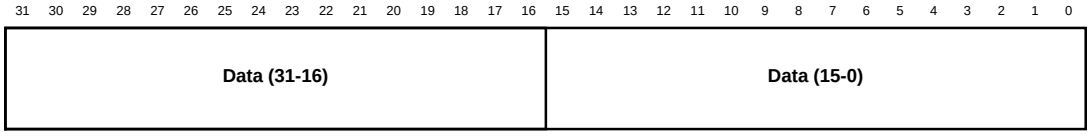
For processor I/O, this field should be set to support at least one bus transfer into the Source FIFO. For example, if the 7711 is configured for 32-bit operation, this field should be set to 4. If configured for 16-bit operation, this field should be set to 2. This setting only affects the SOURCE FIFO READY bit used for processor I/O, and not the SDREQ# pin for DMA operations. The 7711 automatically waits at least one bus transfer before setting the SDREQ# pin.

5.9.2 Dest FIFO Threshold

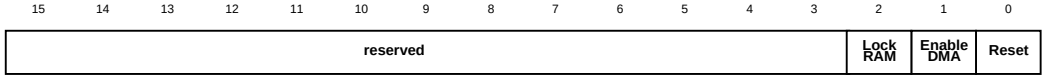
This field sets the Dest FIFO Threshold. When the number of bytes residing in the Destination FIFO is greater than the value set in this field, the DEST FIFO READY bit in the Status register will be set to one and the DDREQ signal will become active. See the *Dest FIFO Data Flow* section for more information regarding this field. The default value of this field is zero.

5.10 Register Summary

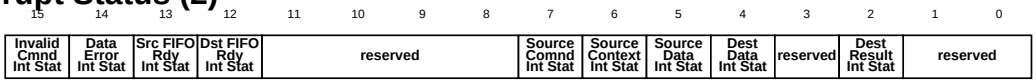
Data (0)



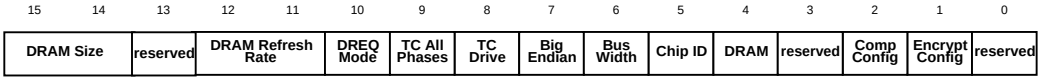
Control (1)



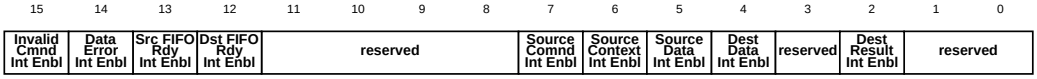
Interrupt Status (2)



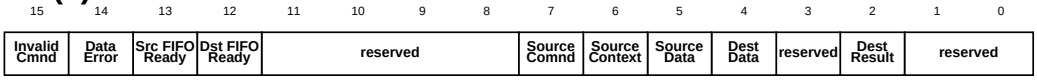
Configuration (3)



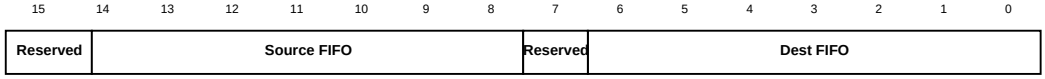
Interrupt Enable (4)



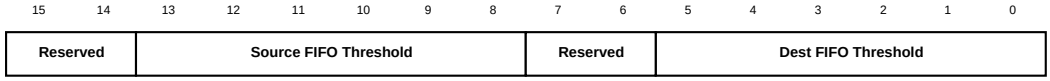
Status (5)



FIFO Status (6)



FIFO Configuration (7)



6 Data Flow

Command Structures, followed by optional Context Structures, followed by Source Data Structures enter the Source FIFO.

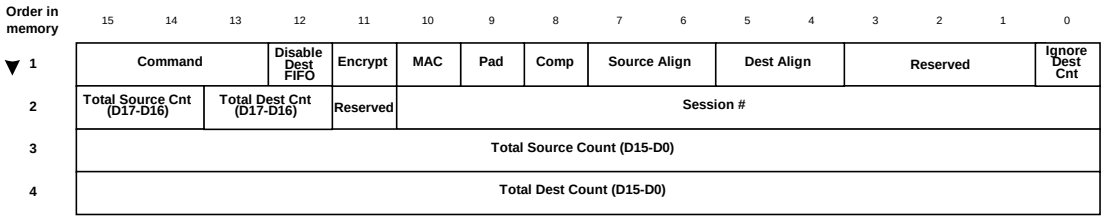
Destination Data Structures, followed by Result Structures, exit the Dest FIFO.

This section will describe these Structures in detail. Reserved bits must be ignored when read and written as zero.
See the Endian Section for details on how the endian bit affects byte alignment for these structures.

7 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.

7.1 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.

7.1.1 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 27.

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 27. Processing Unit Ordering

7.1.2 Command Termination Conditions

A command 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 7711 with a software or hardware reset.

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

7.1.3 Base Structure Fields

Command

The COMMAND field specifies the command to be executed by the enabled processing units. The valid values are listed in Figure 28. 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 28. 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.

If this command is issued while the chip is in the lock RAM mode, then only zeros will be read from the Context RAM.

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.

7.1.4 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. There will be a single data transfer for the data phase, and the value of the data in this transfer is undefined. Context phase and result phase will produce data normally.

The Dest Counter will not be decremented if the DestFIFO is disabled.

7.1.5 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.

7.1.6 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.

7.1.7 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.

7.1.8 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.

7.1.9 Source Align

If the first byte of source data is not 32-bit (or 16-bit) aligned, the SOURCE ALIGN field may be used to ignore the first few bytes of source data. The number of bytes ignored is set in this field.

The ignored bytes are counted by the Total Source Counter.

If the bus width is configured for 16 bits, then the SOURCE ALIGN field must be one or zero. Otherwise the values three to zero are valid.

7.1.10 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.

The inserted bytes are counted by the Dest Counter.

If the bus width is configured for 16 bits, then the DEST ALIGN field must be one or zero. Otherwise the values three to zero are valid.

7.1.11 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.

7.1.12 Session

For encode and decode commands this field specifies the context to be used. The maximum number of sessions the 7711 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.

7.1.13 Total Source Count

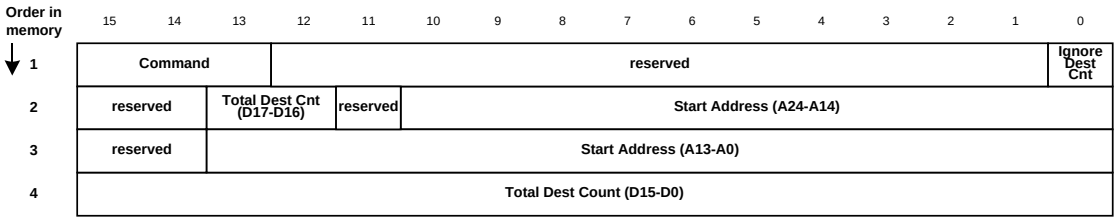
This is the initial value used for the Total Source Counter. The Total Source Counter is decremented for each data phase source byte processed by the processing units.
Total Source Count will decrement for every data phase byte processed, including bytes specified by the SOURCE ALIGN field. It will not decrement for any extra bytes resulting from non-word alignment in the last transfer.

7.1.14 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.

7.2 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 Read RAM Command structure must be appended with eight bytes of zero's. Note that there is no source data block for the Read RAM command.

The destination data block will consist of the contents in the context RAM, and results will be passed in the result structure. Figure 29 shows how the STC# and DTC# signals interact with the Read RAM command, with both settings of TC ALL PHASES.

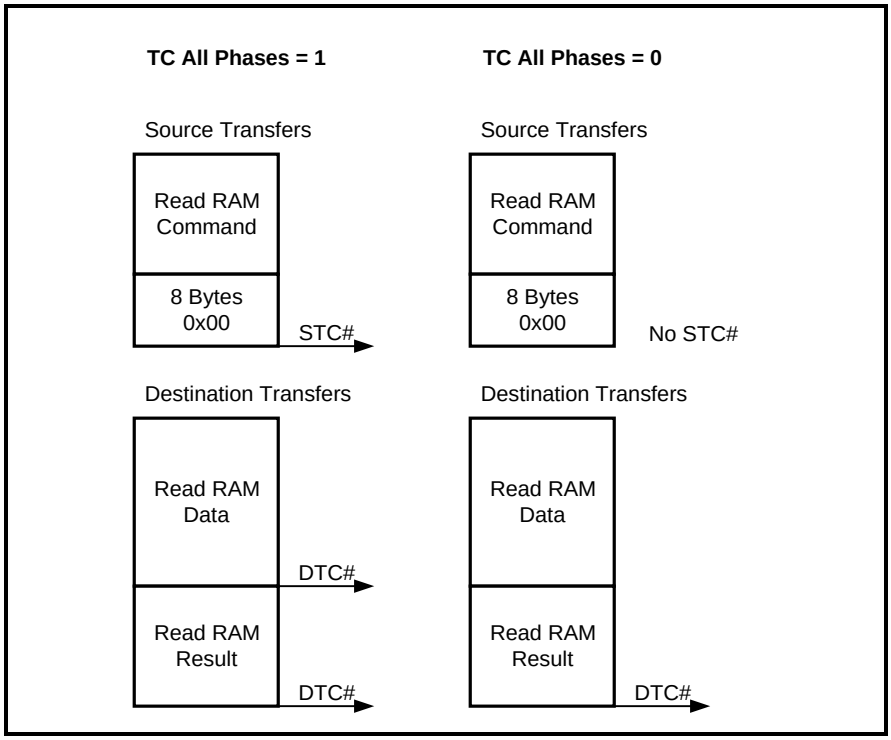


Figure 29. Read RAM data flow

7.2.1 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 value of the TOTAL DEST COUNT field must not be set to zero.

7.2.2 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.

7.3 Write RAM Command Structure

Order in memory	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Command			reserved											Ignore Source Cnt	reserved
2	Total Source Cnt (D17-D16)		reserved			Start Address (A24-A14)										
3	Total Source Count (D15-D0)															
4	reserved		Start Address (A13-A0)													

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 7711 produces 8 bytes of results, of which four bytes are transferred via the destination data phase and four via the results phase. There is no destination data produced by the 7711 for this command. Figure 30 shows how the STC# and DTC# signals interact with the Write RAM command, with both settings of TC ALL PHASES.

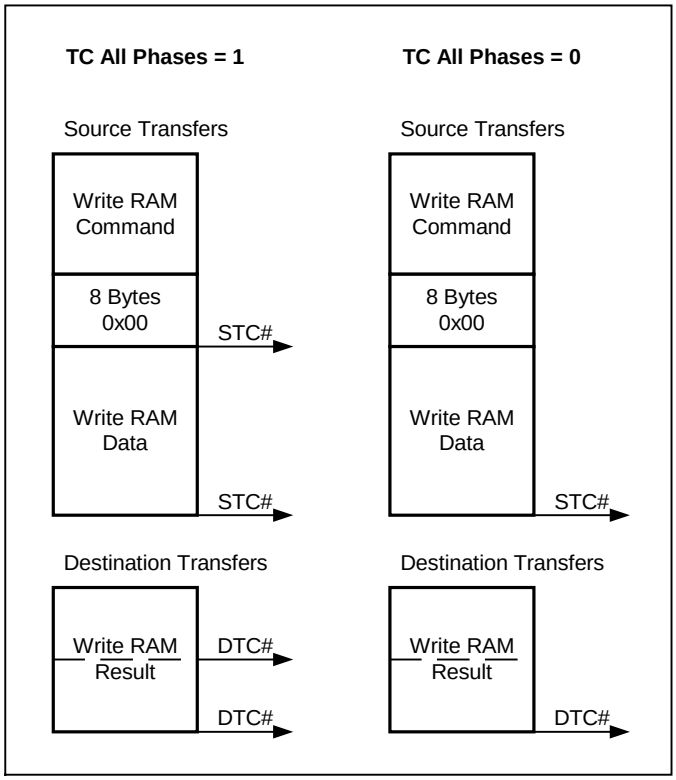


Figure 30. Write RAM data flow

7.3.1 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 value of the TOTAL SOURCE COUNT field must not be set to zero.

7.3.2 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.

7.4 Compression Command Structure

Order in memory	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Comp Source Cnt (D17-D16)		reserved		Performance				reserved			Clear Hist	Update History	Strip 0/ Restart	reserved	MPPC
2	Comp Header Count (D15-D0)															
3	Comp Source Count (D15-D0)															
4	reserved															

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

7.4.1 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 setting produces the lowest compression ratio. The slowest setting produces the highest compression ratio. Figure 31 illustrates the effect of the PERFORMANCE field when compressing a text file containing the Constitution of the United States when the compression RAM is SRAM. Figure 32 illustrates the affect of the PERFORMANCE field when compressing a text file containing the Constitution of the United States when the compression RAM is DRAM.

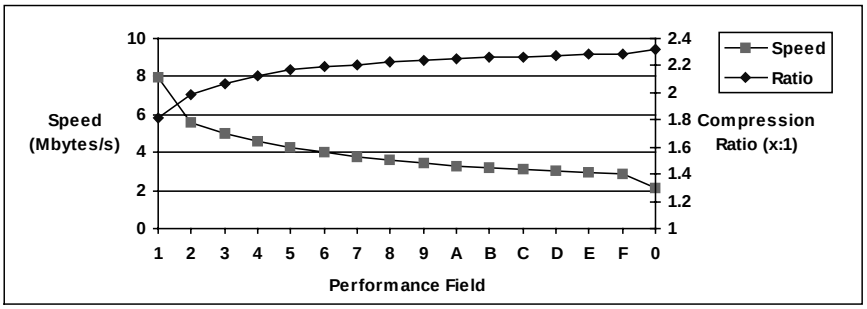


Figure 31. Affect of Performance field with SRAM

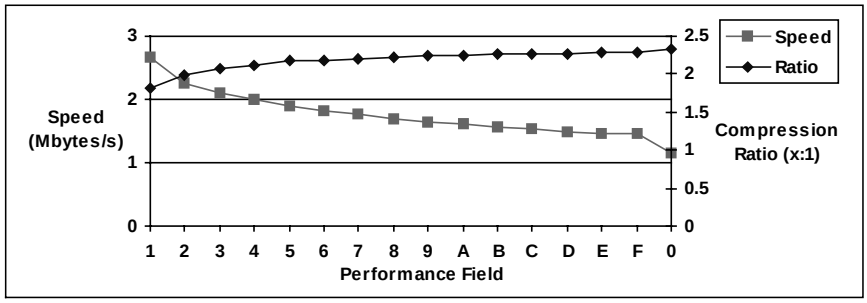


Figure 32. Affect of Performance field with DRAM

7.4.2 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.

7.4.3 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.

7.4.4 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 7711 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.

7.4.5 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 RAM 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 Context RAM 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 Context RAM 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

7.4.6 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.

7.4.7 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.

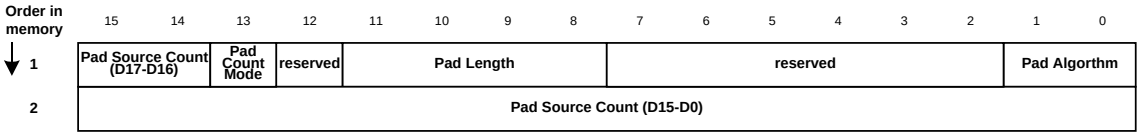
During a compression operation, when the COMP SOURCE COUNTER reaches zero, or after the last byte (as defined by the Total Source Count) has been processed, the compression processing unit will flush out its internal data. If in LZS mode, an End Marker will be appended to the compressed data stream.

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.

7.5 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 must be appended to the end of the Pad Command Structure. The value of these bytes is reserved.

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.

7.5.1 Pad Count Mode

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.

7.5.2 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.

7.5.3 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 35. Pad Algorithms

7.5.4 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.

7.5.5 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.

7.6 MAC Command Structure

Order in memory	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MAC Source Cnt (D17-D16)		MAC Count Mode	reserved	New Key	reserved	MAC Position		reserved	Insert MAC	MAC Result	Truncate MAC	MAC Mode		MAC Algorithm	
2	MAC Header Count (D15-D0)															
3	MAC Source Count (D15-D0)															
4	reserved															

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.

7.6.1 MAC Count Mode

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.

7.6.2 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).

7.6.3 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 36.

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 36. MAC Position

7.6.4 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, and in the Status register.

See Figure 39 for a summary of the relationship between this bit, and other bits in the MAC Command Structure.

7.6.5 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 39 for a summary of the relationship between this bit, and other bits in the MAC Command Structure.

7.6.6 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.

7.6.7 MAC Mode

This field selects the MAC mode as defined in Figure 37.

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

Figure 37. MAC Modes

7.6.8 MAC Algorithm

This field selects the MAC hash algorithm as defined in Figure 38.

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

Figure 38. MAC Hash Algorithms

7.6.9 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.

7.6.10 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.

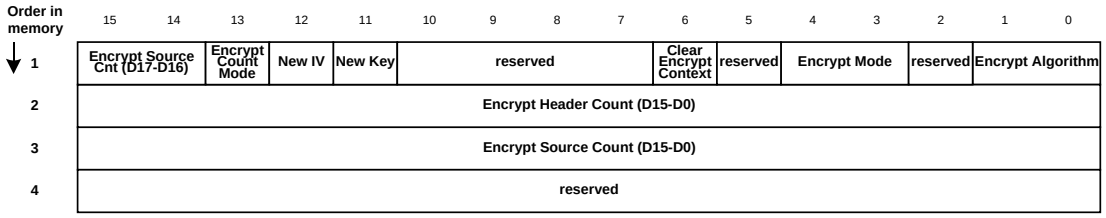
The MAC SOURCE COUNTER will begin to decrement after the MAC HEADER COUNTER has expired, or after the last byte processed by the previous processing unit, as set by the MAC COUNT MODE bit.

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 39. MAC Behavior

7.7 Encryption Command Structure



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

The encryption context will be flushed after the Pad Source Count reaches zero. The internal context information will be cleared before the next operation.

7.7.1 Encrypt Count Mode

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.

7.7.2 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.

7.7.3 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.

7.7.4 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 current context (from the previous command) in the 7711 is preserved. That is, not only is the encryption context not written out, but also the previous encryption context in the Context RAM remains valid.

7.7.5 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 40.

If DES or Triple-DES is not selected, this field must be set to zero.

The 7711 CFB mode is CFB-64.

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

Figure 40. Encrypt Mode

7.7.6 Encrypt Algorithm

This field selects the encryption algorithm as defined in Figure 41.

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 41. Encrypt Algorithm

7.7.7 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.

7.7.8 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.

7.8 Command Structure Summary

Base

Order in memory	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Command			Disable Dest FIFO	Encrypt	MAC	Pad	Comp	Source Align		Dest Align		Reserved			Ignore Dest Cnt
2	Total Source Cnt (D17-D16)			Total Dest Cnt (D17-D16)	Session #											
3	Total Source Count (D15-D0)															
4	Total Dest Count (D15-D0)															

Read RAM

Order in memory	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Command			Reserved												
2	Total Source Cnt (D17-D16)			Reserved												
3	Total Source Count (D15-D0)															
4	Reserved			Start Address (A13-A0)												

Write RAM

Order in memory	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Command			Reserved												Ignore Dest Cnt
2	Reserved			Total Dest Cnt (D17-D16)	Reserved	Start Address (A24-A14)										
3	Reserved			Start Address (A13-A0)												
4				Total Dest Count (D15-D0)												

Compression

Order in memory	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Comp Source Cnt (D17-D16)			Reserved		Performance			Reserved			Clear Hist	Update Hist	Strip 0/ Restart	Reserved	MPPC
2	Comp Header Count (D15-D0)															
3	Comp Source Count (D15-D0)															
4	Reserved															

Pad

Order in memory	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Pad Source Cnt (D17-D16)		Pad Cnt Mode	Reserved	Pad Length				Reserved							
2	Pad Source Count (D15-D0)															

MAC

Order in memory	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MAC Source Cnt (D17-D16)		MAC Cnt Mode	Reserved	New Key	Reserved	MAC Position		Reserved	Insert MAC	MAC Result	Truncate MACt	MAC Mode		MAC Algorithm	
2	MAC Header Count (D15-D0)															
3	MAC Source Count (D15-D0)															
4	Reserved															

Encrypt

Order in memory	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Encrypt Source Cnt (D17-D16)		Encrypt Cnt Mode	New IV	New Key	Reserved				Clr Encrypt Context	Reserved	Encrypt Mode		Reserved	Encrypt Algorithm	
2	Encrypt Header Count (D15-D0)															
3	Encrypt Source Count (D15-D0)															
4	Reserved															

8 Source Context Structures

Encryption and authentication information will be transferred to the 7711 in the Source Context phase. This information (if required by the device) will consist of an encryption key, an encryption IV, and/or an authentication (MAC) key.

Source context is transferred to the 7711 in the following order: MAC key, encryption key, and then encryption IV.

8.1 MAC Source Context Structure

MAC Source Context, which consists of the MAC key, must be provided if it is expected by the 7711. The chip will expect a MAC Key if the NEW KEY bit in the MAC command structure is set to one.

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.

8.2 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 7711. 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.

8.2.1 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 7711 in network byte order, as shown in Figure 42..

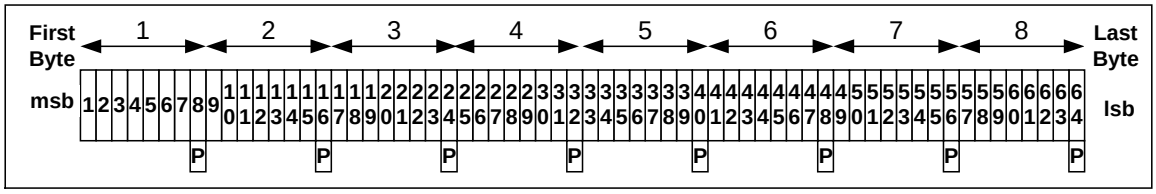


Figure 42. DES key format

Triple-DES
The Triple-DES key is a string of 168 bits that are transferred to the 7711 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.

8.2.2 IV

The Initialization Vector (IV) is a string of 8 bytes, which are transferred to the 7711 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.

9 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 the bus width (16 or 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. Reserved bits must be ignored when read.

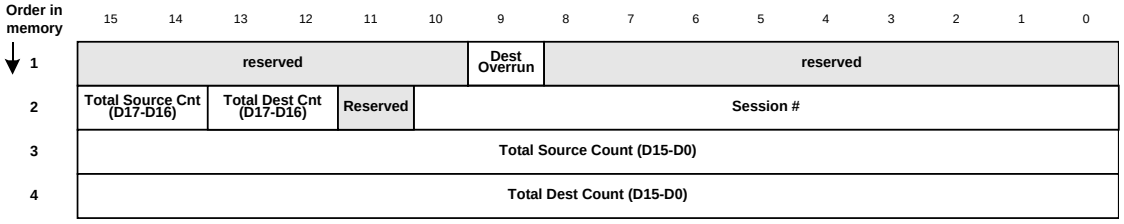
10 Destination Data Structures

Data exiting the Dest FIFO is simply a string of data bytes.

The Dest Data Structure size will be a multiple of the bus width (16 or 32 bits). Unused bytes should be automatically discarded. The number of bytes used can be determined by the TOTAL DEST COUNT field in the Base Result Structure.

11 Result Structures

11.1 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.

11.1.1 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.

11.1.2 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. Furthermore, 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 cannot be used.

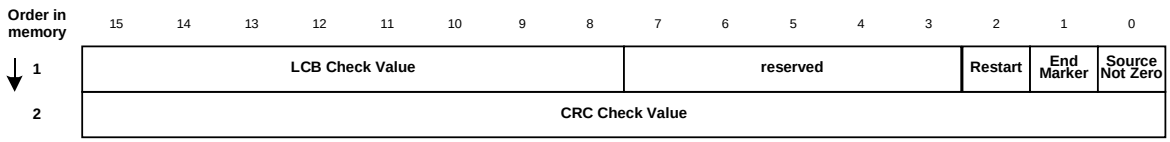
11.1.3 Total Source Count

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. The total Source Count is undefined for a read RAM. The value of the Total Source Count should be zero if the command terminated cleanly.

11.1.4 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.

11.2 Compression Result Structure



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

11.2.1 LCB Check Value

This field is always computed on the raw data side of the compression engine. For an encode operation, the check value will be calculated on the data as it enters the compression engine. For a decode operation, the check value will be computed on the decompressed data as it exits the decompression engine.

The LCB check value will only be calculated during the data phase of the command. The check value will not be calculated during command, context, or result phases.

The LCB check value will be calculated when only the compression engine is used. If any of the other engines are engaged, the LCB value is not valid, and in this case the LCB field in the Compression Result Structure is reserved.

If the LCB check value is not desired, the LCB field may be ignored.

11.2.2 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.

11.2.3 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.

11.2.4 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.

11.2.5 CRC Check Value

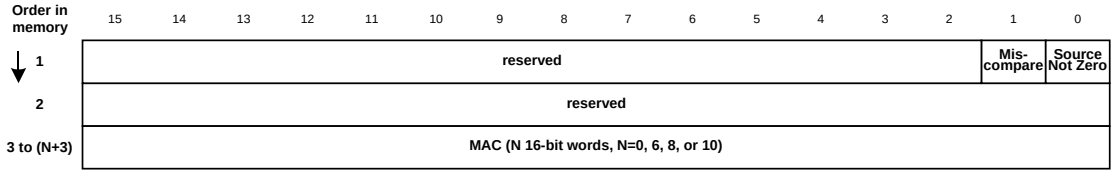
This field is always computed on the raw data side of the compression engine. For an encode operation, the check value will be calculated on the data as it enters the compression engine. For a decode operation, the check value will be computed on the decompressed data as it exits the decompression engine.

The CRC check value will only be calculated during the data phase of the command. The check value will not be calculated during command, context, or result phases.

The CRC check value will be calculated when only the compression engine is used. If any of the other engines are engaged, the CRC value is not valid, and in this case the CRC field in the Compression Result Structure is reserved.

If the CRC check value is not desired, the CRC field may be ignored.

11.3 MAC Result Structure



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

11.3.1 Miscompare

During a decode operation, if the MAC processing unit detects a MAC miscompare, this bit will be set to one. The MAC MISCOMPARE bit in the Status register will also be set to one.

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

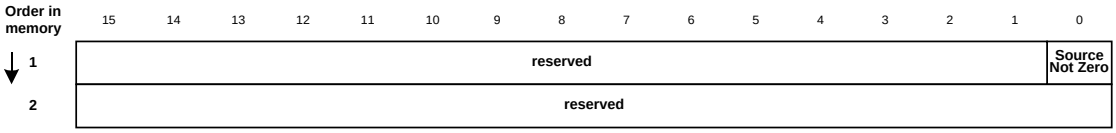
11.3.2 Source Not Zero

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.

11.3.3 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.

11.4 Encrypt Result Structure



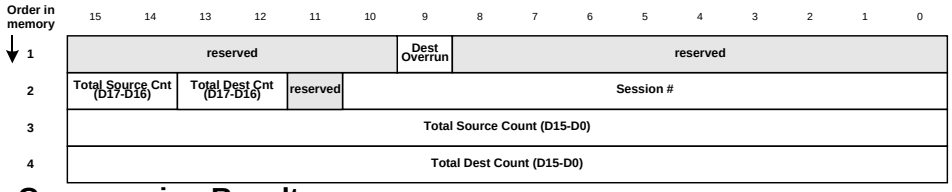
If the Encryption processing unit is enabled, this data structure will be appended to the end of the previous Result Structure (Base, Compression, or MAC).

11.4.1 Source Not Zero

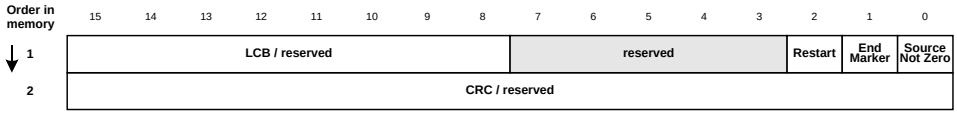
If this bit is set to one, the Encryption processing unit stopped processing a block of data before the ENCRYPTION SOURCE COUNT reached zero. This may be caused by improper setting of the ENCRYPTION SOURCE COUNT relative to the TOTAL SOURCE COUNT.

11.5 Result Structure Summary

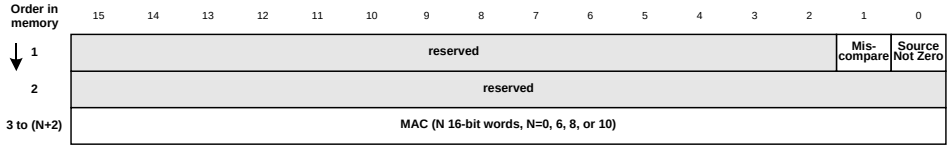
Base Result



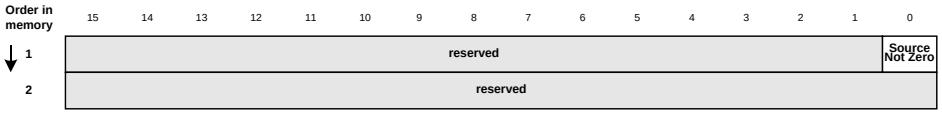
Compression Result



MAC Result



Encryption Result



12 Timing Descriptions

12.1 CPU and DMA Bus Interface

The CPU and DMA bus may be configured for Asynchronous, Synchronous, or 68360 timing modes. This is configured by the use of the BMODE pins. In the following discussions (including the timing section), the signals DREQ# and DACK# are used in place of the signals SDREQ#, DDREQ#, SDACK#, and DDACK#. Furthermore the signal TC# is used in place of the signals STC# and DTC#.

12.1.1 Asynchronous Modes on CPU Interface

If configured for Asynchronous operation, the bus interface is driven either externally by the BCLK input pin or internally by half the frequency of the PCLK input pin, depending on the BMODE pin settings.
In a typical CPU access the R/W and ADDR signals must be asserted prior to the assertion of the CS# and DS# signals. The R/W and ADDR signals must remain stable after the trailing edge of CS# and DS#.

On a read cycle, the 7711 will drive the data bus with valid data after the later of the leading edge of the CS# signal or the leading edge of the DS# signal. The data bus will become tri-state after the earlier of the trailing edge of CS# or the trailing edge of DS#.

On a write cycle, the data bus must be valid before the earlier of the trailing edge of the CS# or the trailing edge of the DS# signals. The data bus must remain valid after the earlier of the trailing edge CS# or the trailing edge of DS#.

The DS# signal may be used to help shape the CS# signal. Internally, these two signals are logically ORed together. That means that the command strobe is active (low) only when both of these inputs are active (low). This can be useful when using certain CPUs, such as the Motorola 68302.

12.1.2 Asynchronous Modes on DMA Interface

If configured for Asynchronous operation, the bus interface is driven either externally by the BCLK input pin or internally by half the frequency of the PCLK input pin, depending on the BMODE pin settings.

A typical DMA transfer consists of DREQ# being asserted by the 7711, and then one or more assertions of the DACK# signal by the system bus to initiate one or more DMA transfers. The minimum DMA transfer cycle time is approximately four BCLK cycle.

The start of a DMA cycle is controlled by the DMA Controller with the assertion of the DACK# signal. The DACK# signal must not be asserted unless the corresponding DREQ# signal is asserted.

On a read cycle, the 7711 will drive the data bus with valid data after the later of the leading edge of the DACK# signal or the leading edge of the DS# signal. The data bus will become tri-state after the earlier of the trailing edge of DACK# or the trailing edge of DS#.

On a write cycle, the data bus must be valid before the earlier of the trailing edge of the DACK# or the trailing edge of the DS# signals. The data bus must remain valid after the earlier of the trailing edge DACK# or the trailing edge of DS#.

The 7711 will deassert the DREQ# signal after the leading edge of the DACK# signal on the last DMA transfer of a burst.

The TC# signal indicates the last DMA transfer to either the Source FIFO (STC#) or from the Dest (DTC#) FIFO for the current command. The 7711 will assert the TC# signal after the trailing edge of DACK# on the second to last DMA transfer. The 7711 will deassert the TC# signal after the trailing edge of the DACK# on the last DMA transfer.

The DS# signal may be used to help shape the DACK# signal. Internally, these two signals are logically ORed together. That means that the command strobe is active (low) only when both of these inputs are active (low). This can be useful when using certain CPUs, such as the Motorola 68302.

12.1.3 Synchronous Mode on CPU Interface

If configured for Synchronous operation, the bus interface is driven by the BCLK signal. All bus signals are relative to the rising edge of the BCLK.

A typical CPU access consists of two clock cycles (T1 and T2), with any number of wait states (Tw) between the T1 and T2 cycles. A bus cycle with no CPU

activity is identified as Ti. Synchronous mode supports a minimum bus cycle time of two clock cycles.

A T1 cycle is identified by the assertion of the CS# signal by the end of a clock. By the end of T1, the R/W and Addr signals must also be valid. The following clock cycle will be either T2 or Tw based on the value of the CS# signal at the end of the following clock cycle. If CS# is active, then this cycle will be Tw. If CS# is inactive, then this cycle will be T2. There can be any number of Tw cycles (including zero).

During a Tw cycle, the data bus will be active. The R/W and Addr signals no longer need to be valid. The following clock cycle will be either T2 or Tw based on the value of the CS# signal at the end of the following clock cycle. If CS# is active, then this cycle will be Tw. If CS# is inactive, then this cycle will be T2.

During T2, the data transfer remains active. The following clock cycle will be either T1 or Ti based on the value of the CS# signal at the end of the following cycle. If CS# is active then this cycle will be T1. If CS# is inactive then this cycle will be Ti.

There can be any number of Ti cycles (including zero). During a Ti cycle, CS# and the data bus will be inactive.

12.1.4 Synchronous Mode on DMA Interface

If configured for Synchronous operation, the bus interface is driven by the BCLK signal. All bus signals are relative to the rising edge of the BCLK.

A typical DMA transfer consists of two clock cycles (T1 and T2), with any number of wait states (Tw) between the T1 and T2 cycles. A bus cycle with no DMA activity is identified as Ti. Synchronous mode will support a minimum bus cycle time of two clock cycles. The start of a DMA cycle is controlled by the DMA Controller with the assertion of the DACK# signal. The DREQ# signal must be valid on a previous clock (or on the same clock) that the DACK# signal is asserted. There can be any number of clock cycles (including zero) between the assertion of the DREQ# signal and the assertion of the DACK# signal. These will be Ti clock cycles.

A T1 cycle is identified by the assertion of the DACK# signal (and DS# being inactive) by the end of a clock. The following clock cycle will be either T2 or Tw based on the values of the DACK# and DS# signals at the end of the following cycle. If DACK# is active and DS# inactive, then this cycle will be Tw. If both DACK# and DS# are active, then this cycle will be T2. There can be any number of Tw cycles (including zero).

During a Tw cycle, the data bus will be active. The following clock cycle will be either T2 or Tw based on the value of the DACK# and DS# signals at the end of the following cycle. If DACK# is active and DS# inactive, then this cycle will be Tw. If both DACK# and DS# are active, then this cycle will be T2.

During T2, the data transfer remains active. The following clock cycle will be either T1 or Ti based on the value of the DACK# and DS# signals at the end of the following cycle. If DACK# is active and DS# inactive, then this cycle will be T1. If DACK# is inactive (and DS# active or inactive) then this cycle will be Ti.

There can be any number of Ti cycles (including zero). During a Ti cycle, the data bus will be inactive.

The DREQ# signal will deassert during the clock following T1 of the last DMA transfer of a burst.

The TC# signal indicates the last DMA transfer to the Source FIFO (STC#) or from the Dest (DTC#) FIFO for the current command. The $\overline{TC}$ signal will be asserted during the clock following the T1 cycle of the last DMA transfer. It will remain asserted for three clocks, and then be deasserted.

The DS# signal may be used to help shape the DACK# signal. Internally, these two signals are combined as follows: COMMAND STROBE # =(DACK# + !DS#). That is internally the COMMAND STROBE# is active when externally DACK# is active (low) and DS# is inactive (high). Data is accessed the clock after the COMMAND STROBE# deactivates. That is, data is accessed the rising edge of the clock when one of these signals change state. From the bus point of view, the data transfer occurs on the rising edge of the clock when DACK# is inactive, or DS# is active while DACK# is active (and has been active for at least one clock).

This can be useful when using certain CPUs, such as the Intel i960 family. For this CPU, you must connect the i960's BLAST# signal to the DS# signal. If DS# is not used, it must be tied high.

12.1.5 68360 Mode on CPU Interface

If configured for 68360 bus mode operation, the bus interface is driven by the BCLK signal. Most of the bus signals are relative to the falling edge of the BCLK.

A typical CPU access consists of six half-clock phases (three clock cycles total) S0 through S5, with any number of wait states (Tw) between the S3 and S4 half clocks (second and third clock). 68360 mode supports a minimum bus cycle time of three clock cycles.

In a typical bus cycle the 68360 asserts the address and R/W signals after the rising edge of S0, and CS# after the falling edge of S1. In a read cycle 7711 drives valid data after the rising edge of the S4 phase of the BCLK, and will hold the data until after CS# deasserts. The 68360 latches the read data on the falling edge of S4. In a write cycle 68360 drives data after the rising edge of S2 and will hold data until after CS# deasserts. 7711 latches write data on the trailing edge of CS#.

Wait states may be inserted by programming internal cycles in the memory controller of greater than 3 clocks. During a wait state, the data bus will be active. The bus cycle ends when CS# is deasserted in the half-clock S5. The next cycle begins when address is asserted on the rising edge and CS# is asserted on the falling edge of S0. This is usually the next CPU clock (BCLK) unless there are idle states.

Idle states are defined when CS# is inactive. There can be any number of idle clock cycles (including zero). During a idle cycle, the data bus will be inactive.

12.1.6 68360 Mode on DMA Interface

If configured for 68360 bus mode operation, the bus interface is driven by the BCLK signal. All bus signals are relative to the falling edge of the BCLK.

68360 mode will support a minimum bus cycle time of three clock cycles.

A typical 68360 IDMA transfer consists of the same 6 half-clock (3 clock cycles) S0 through S5 as the CPU I/O bus cycles. There also may be any number of wait states in IDMA cycles.

The start of a IDMA cycle is controlled by the 68360's IDMA Controller with the assertion of the DACK# signal, after having processed the DREQ# signal it had received from 7711.

The 68360 asserts the DACK# signal on the falling edge of S0. The 68360 will hold DACK# active for the programmed number of external wait states.

If this is a read cycle, then 7711 will drive valid data after the rising edge of the S4 phase of the BCLK, and will hold data until after DACK# deasserts. If this is a write cycle, then the system memory will provide data after the rising edge of S1, and hold data until after CS# or CAS# deasserts. The 7711 will latch data on the trailing edge of DACK#.

The 7711 will deassert the DREQ# signal on the BCLK falling clock edge when DACK# is sampled inactive just before the last DMA transfer of a burst.
The TC# signal indicates the last DMA transfer either to the Source FIFO (STC#) or from the Dest FIFO (DTC#) for the current command. The 7711 will assert the TC# signal on the BCLK falling clock edge when DACK# is sampled inactive just prior to the last DMA transfer of a burst. TC# will remain asserted until the falling edge of BCLK that DACK# is sampled inactive.

If the 7711 requires a single IDMA transfer, then the cycle begins with both DREQ# and TC# being asserted. The IDMA transfer completes normally with DACK# being asserted. DREQ# and TC# are deasserted on the falling edge of BCLK that DACK# is sampled inactive.

The DS# signal is not used in 68360 mode and must be tied low or high.

13 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

Figure 43. Absolute maximum ratings

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.

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 44. 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 45. DC electrical characteristics

Symbol	Parameter	Conditions
C_{L1}	Output load on CPU Data bus	47 pF
C_{L2}	Output load on Context RAM Interface – SRAM	5pF min, 40 pF max
C_{L2}	Output load on Context RAM Interface - DRAM	5pF min, 40 pF max
C_{L3}	Output load on IRQ#	27 pF
C_{L4}	Output load on all DMA pins	20 pF
C_{L5}	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 bus load conditions.

Figure 46. AC specification definition

Signal	Pins	Derating (ns per 10pF)*
Data bus (high to low)	DATA31-0	0.6ns per 10pF
Data bus (low to high)	DATA31-0	0.35ns 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
All DMA signals	All DMA pins	0.51ns per 10pF
Interrupt request	IRQ#	1.37ns 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 47. AC specification derating

14 Timing Specifications

Number	Description	Min	Max	Units
1	Reset width	t_{BCLK} + 7		ns
2	First 7711 access after Reset	12 t_{BCLK} + 30 t_{PCLK}		ns

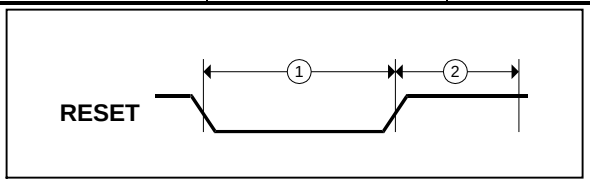


Figure 48. Reset timing

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

Note: Delays are measured to/from 1.4V.

Figure 49. 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 50. External PCLK clock with PLL enabled

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 BCLK clock

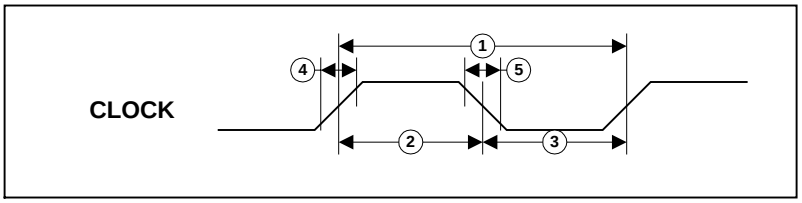


Figure 52. External clock

Number	Description	Min	Max	Units
1	Addr setup before CS#, DS# active	7		ns
2	Addr hold after CS#, DS# inactive	5		ns
3	R/W setup before CS#, DS# active	7		ns
4	CS#, DS# active width	t_{BCLK} +7		ns
5	CS#, DS# inactive width	3 t_{BCLK} +7		ns
6	R/W hold after CS#, DS# inactive	5		ns
7	Read data valid after CS#, DS# active		27	ns
8	Read data hold after CS#, DS# inactive	2	16	ns
9	Write data setup before CS#, DS# inactive	7		ns
10	Write data hold after CS#, DS# inactive	5		ns

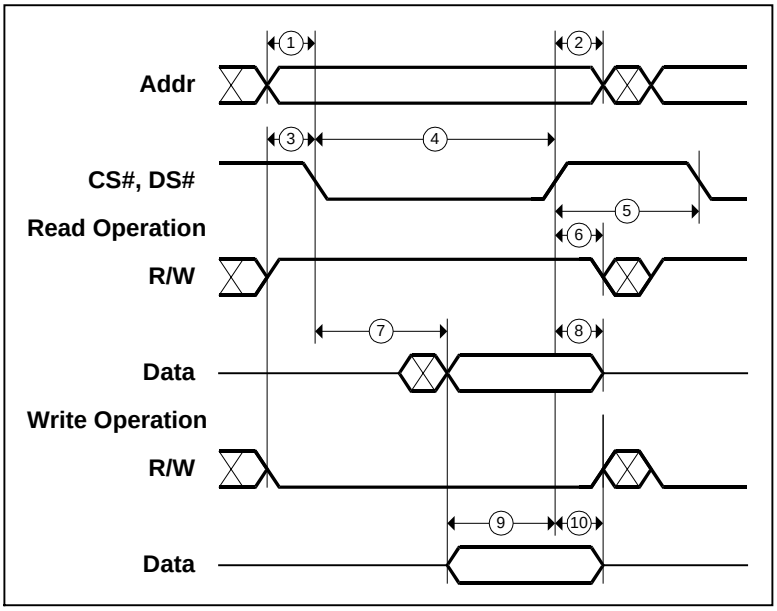


Figure 53. Asynchronous CPU Timing

Number	Description	Min	Max	Units
1	DACK# , DS# active width	t_{BCLK} +7		ns
2	DACK#, DS# inactive width	3 t_{BCLK} +7		ns
3	DREQ# inactive after DACK#, DS# active		15	ns
4	TC# inactive after DACK#, DS# inactive		2 t_{BCLK} +7	ns
5	TC# active after DACK#, DS# inactive		2 t_{BCLK} +7	ns
6	Read data valid after DACK#, DS# active		27	ns
7	Read data hold after DACK#, DS# inactive	2	16	ns
8	Write data setup before DACK#, DS# inactive	7		ns
9	Write data hold after DACK#, DS# inactive	5		ns
10	DACK# asserted after DREQ# asserted	t_{BCLK} +7		ns

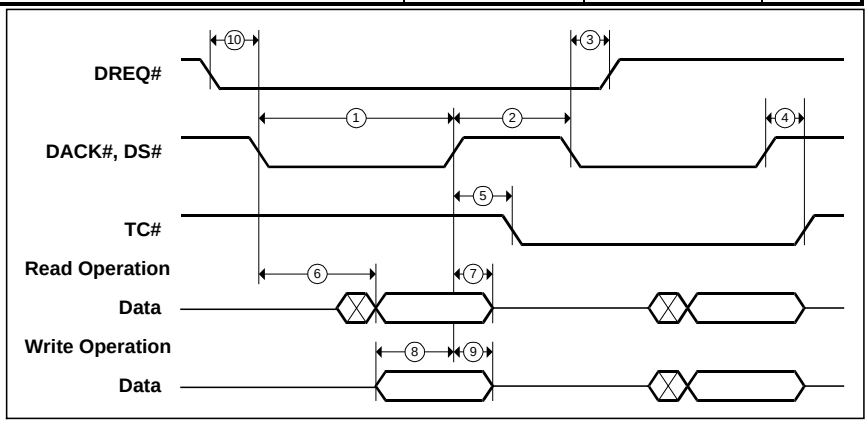


Figure 54. Asynchronous DMA Timing

Number	Description	Min	Max	Units
1	Addr setup	5		ns
2	Addr hold	3		ns
3	CS# setup	5		ns
4	CS# hold	3		ns
5	R/W setup	5		ns
6	R/W hold	3		ns
7	Read data output valid delay		24	ns
8	Read data hold	5		ns
9	Write data setup	11		ns
10	Write data hold	3		ns

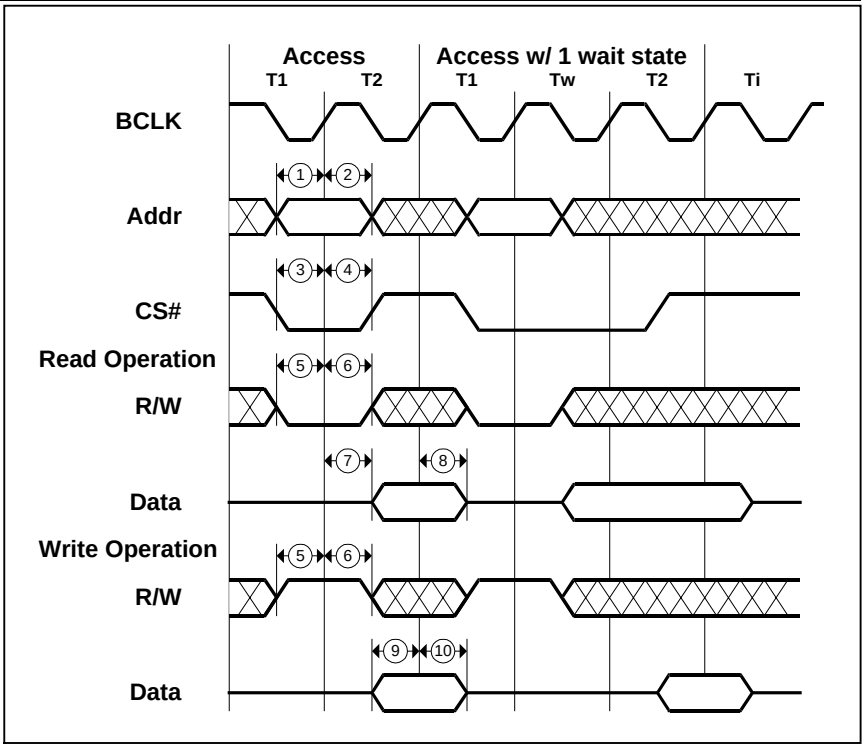


Figure 55. Synchronous CPU Timing

Number	Description	Min	Max	Units
1	DREQ# output valid delay		14	ns
2	DREQ# hold	3		ns
3	DACK# setup	5		ns
4	DACK# hold	4		ns
5	DS# setup	5		ns
6	DS# hold	4		ns
7a	TC# output valid delay - totem pole mode		14	ns
7b	TC# output valid delay - tri-state mode		16	ns
8	TC# hold	3		ns
9	Read data output valid delay		24	ns
10	Read data hold	5		ns
11	Write data setup	11		ns
12	Write data hold	3		ns

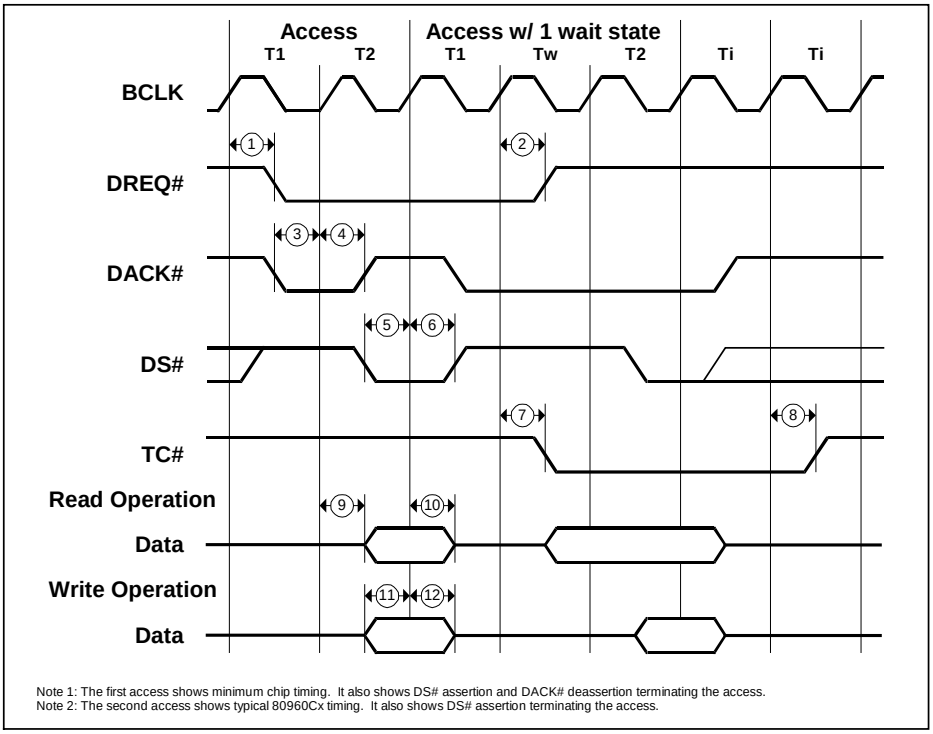


Figure 56. Synchronous DMA Timing

Number	Description	Min	Max	Units
1	Addr setup to CS# active	7		ns
2	Addr hold from CS# inactive	7.5		ns
3	CS# setup to falling edge of BCLK	16		ns
4	CS# hold from falling edge of BCLK	4		ns
5	R/W setup to CS# active	6.5		ns
6	R/W hold from CS# inactive	7.5		ns
7	Read data output valid before S4 falling edge	6		ns
8	Read data hold after CS# inactive	1		ns
9	Write data setup to CS# inactive	20		ns
10	Write data hold after CS# inactive	7.5		ns

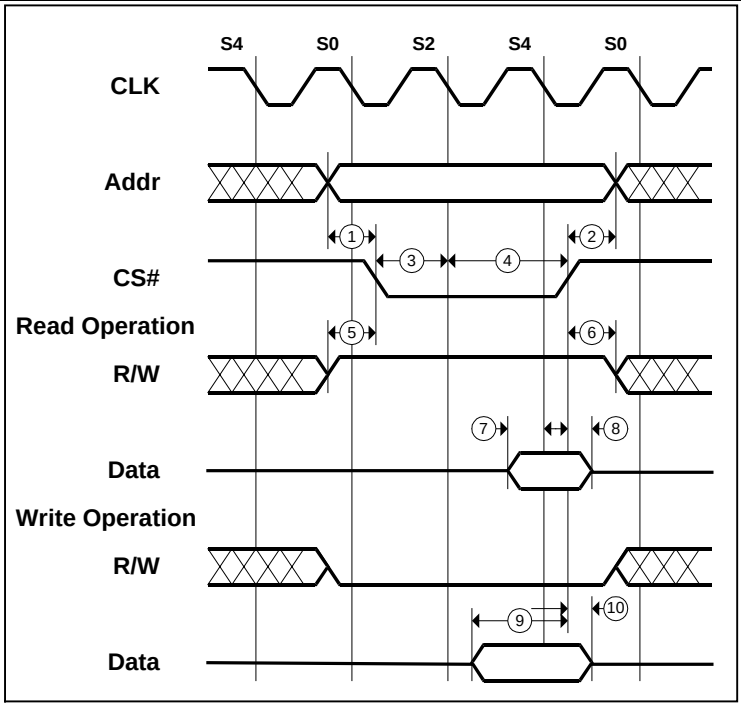


Figure 57. 68360 CPU Timing

Number	Description	Min	Max	Units
1	DACK# setup to falling edge of BCLK	12		ns
2	DACK# hold from falling edge of BCLK	4		ns
3	Read data output before S4 falling edge	6		ns
4	Read data hold after DACK# inactive	1		ns
5	Write data setup to DACK# inactive	20		ns
6	Write data hold after DACK# inactive	7.5		ns
7	DREQ# output delay after falling edge of BCLK		15	ns
8	DREQ# hold after falling edge of BCLK	4		ns
9a	TC# active delay after falling edge of BCLK (totem-pole mode)		13	ns
9b	TC# active delay after falling edge of BCLK (tri-state mode)		15	ns
10	TC# hold after falling edge of BCLK	4		

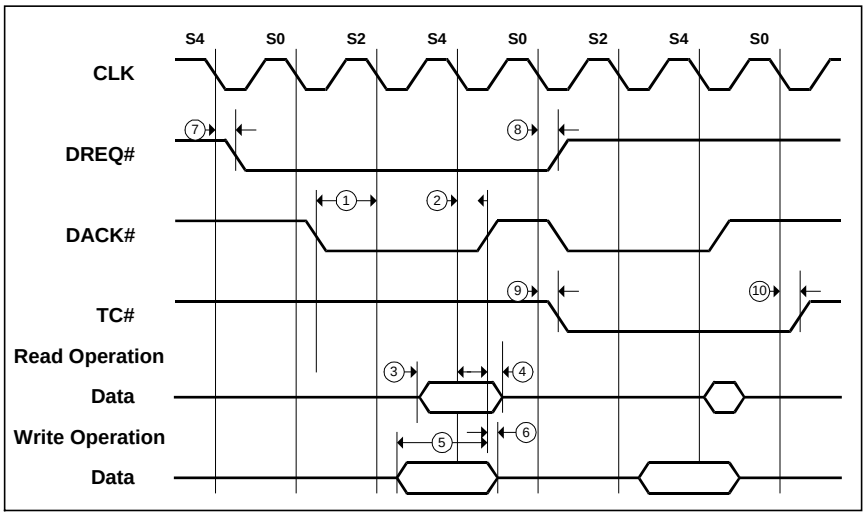


Figure 58. 68360 DMA Timing

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

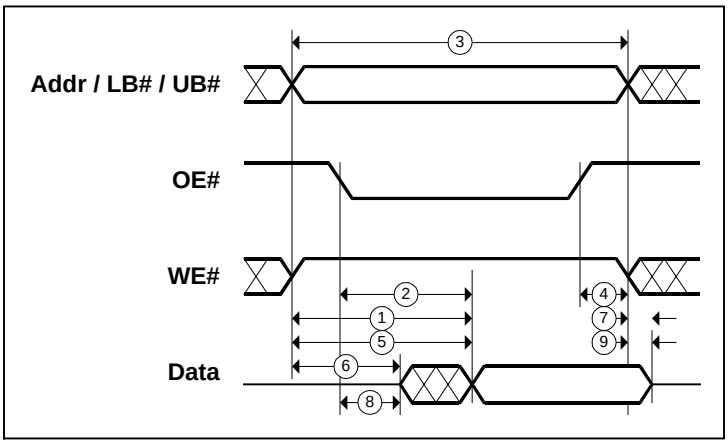


Figure 59. Compression SRAM Read Timing

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	$0.5\ t_{ECLK} - 5$		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

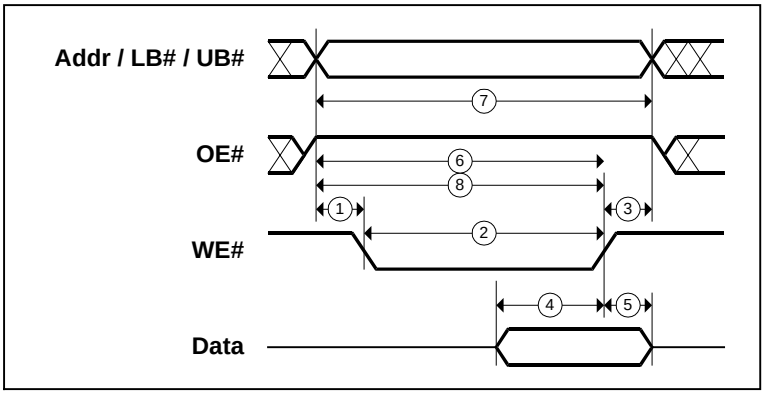


Figure 60. Compression SRAM Write Timing

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}		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} - 3		ns
12	RAS# 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}		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 col. 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} + 1		ns
30	Write data setup to CAS#	0.5 t_{ECLK} – 13		ns
31	Write data hold from CAS#	0.5 t_{ECLK}		ns
32	Output buffer turn-off from WE# active (write)		15	ns
33	WE# Pulse Width	t_{ECLK} - 2		ns
34	RAS# hold after CAS#	0.5 t_{ECLK}		ns
35	RAM buffer turn off from RAS- (read)	0	15	ns

Figure 61. Compression DRAM Read and Write Timing

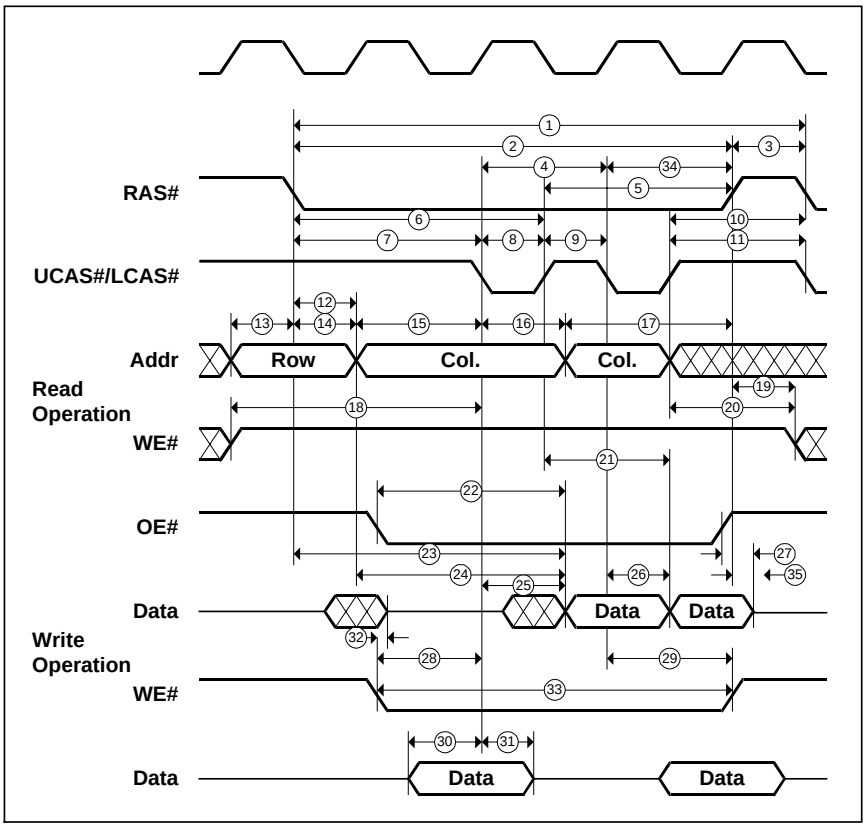


Figure 61. Compression DRAM Read and Write Timing
(more on the previous page)

Number	Description	Min	Max	Units
1	CAS- setup to RAS-	0.5 t_{ECLK}		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.

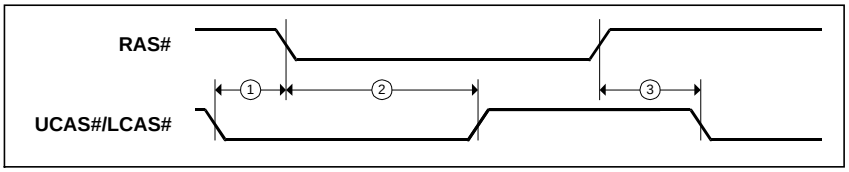


Figure 62. Compression DRAM Refresh Timing

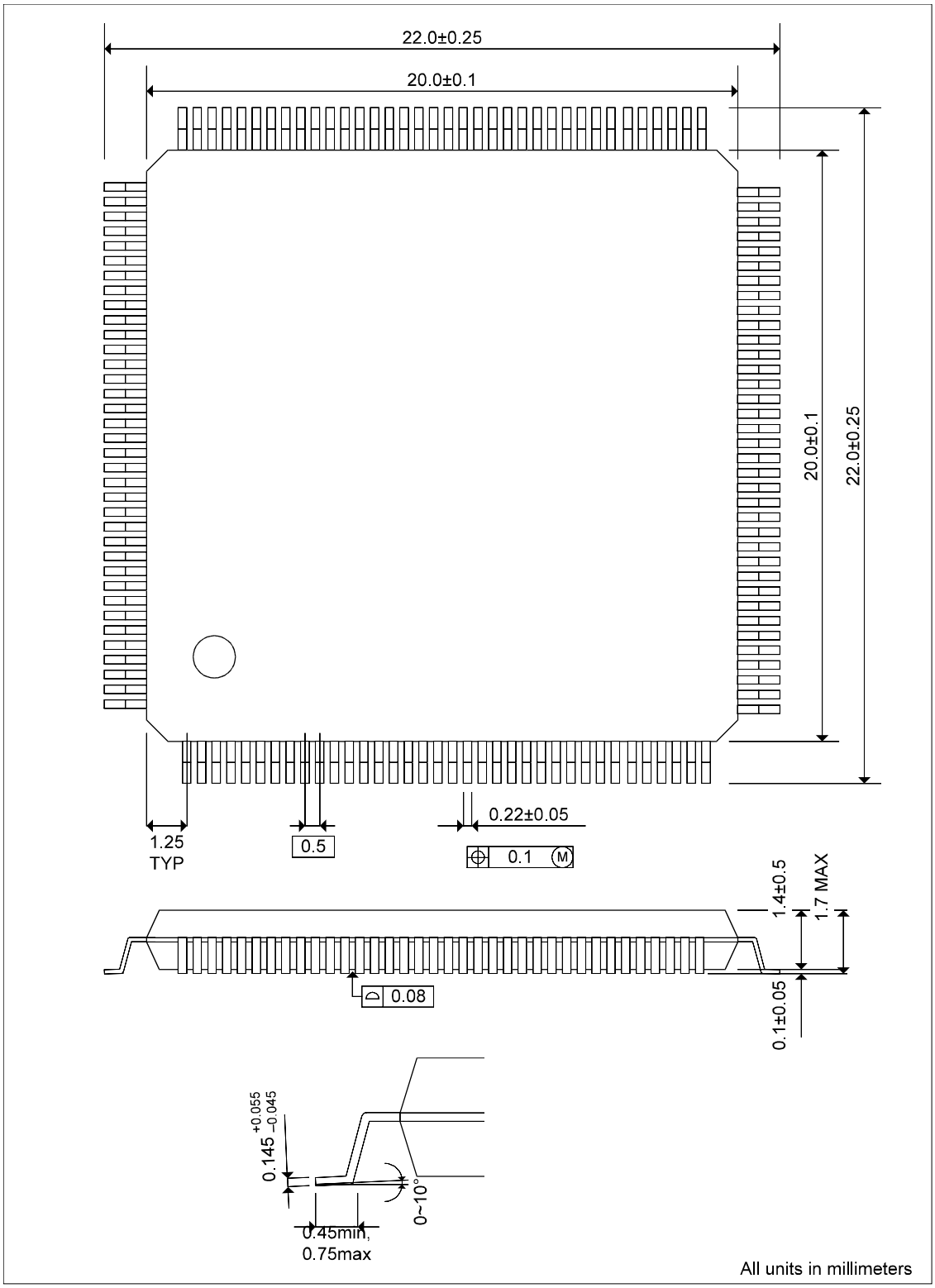


Figure 63. 144-pin TQFP package

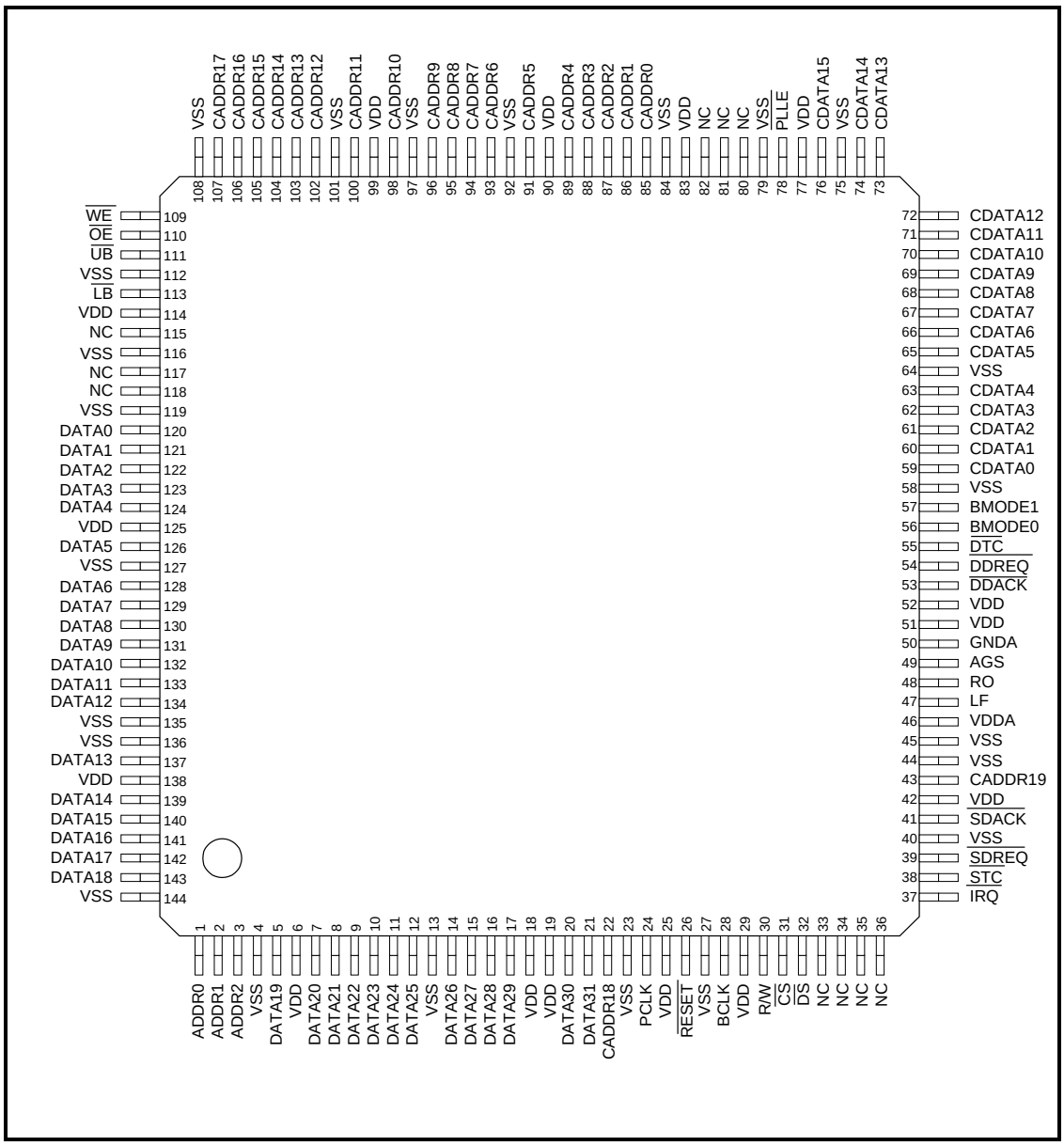


Figure 64. 144-pin TQFP pinout