DATA SHEET 9711

Data Compression Processor

hifn

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.

As of October 1, 1998, our new address will be:

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

1 Product Description

The Hi/fn ™ 9711 is a high-performance lossless data compression processor which may be used in a variety of applications, including data communication applications. This product implements the LZS® and MPPC™ data compression algorithms. The LZS algorithm has been standardized by many standards organizations including ANSI (X3.241), QIC (122), IETF (RFC-1967, RFC1974), TIA/EIA (655), and the Frame Relay Forum (FRF.9).

Multiple compression histories are supported, providing the maximum compression ratio possible for multi-stream data communication applications.

CPU intervention is not required between operations, eliminating any latency issues. Operations are pipelined, and will switch automatically when an operation completes.

Features

- Supports the LZS and MPPC data compression algorithms
- Maximum 8 Mbyte/s compression, 15 Mbyte/s decompression speed
- No CPU intervention required between operations (no latency)
- Simple operation
- Multiple compression histories (up to 2048 simultaneous histories)
- Single bus DMA slave interface

Applications

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

Part Number	Package
9711 PT4	100-pin thin plastic quad flat pack
9711 PT6	144-pin thin plastic quad flat pack

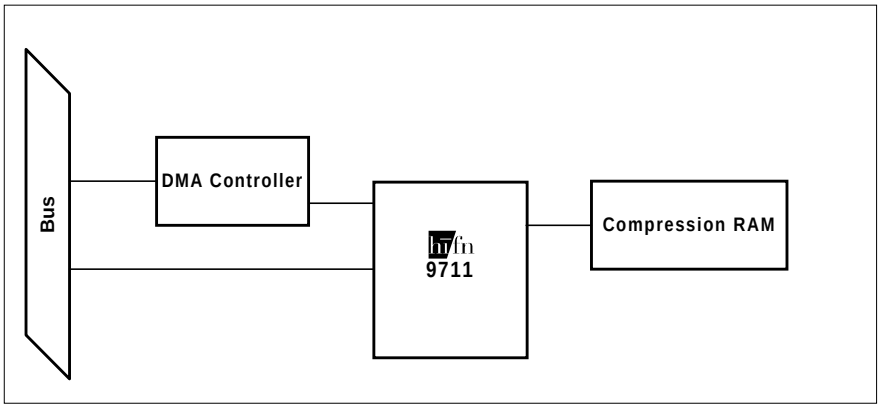


Figure 1. System Concept

2 Pin descriptions

100-pin TQFP	Signal	Type	Description
CPU Interface			
18-7,5,100-82	DATA31-0	PI/O4	32-bit data bus. (16-bit option)
3-1	ADDR2-0	I	Register select
23	R/W	I	Read/Write
24	CS#	I	Command Strobe
26	IRQ#	OD2	Interrupt output
25	DS#	I	Data Strobe
DMA Interface			
28	SDREQ#	O4	Source FIFO DMA request
30	SDACK#	I	Source FIFO DMA acknowledge
27	STC#	O6/OD6	Source FIFO terminal count
34	DDREQ#	O4	Dest FIFO DMA request
33	DDACK#	I	Dest FIFO DMA acknowledge
35	DTC#	O6/OD6	Dest FIFO terminal count
Compression RAM			
32,19,75-73	CADDR19-15	O6	CRAM Address
72-70	CADDR14-12	O8	CRAM Address
69,67,65-56	CADDR11-0	O6	CRAM Address
54,52-38	CDATA15-0	I/O4	CRAM Data
76	WE#	O8	Write enable
77	OE#	O8	Output enable
80	LB#	O4	Lower Byte enable
78	UB#	O4	Upper Byte enable
Miscellaneous Signals			
21	RESET#	SI	Reset
37,36	BMODE1-0	I	Bus mode
22	BCLK	CI	Bus clock
20	PCLK	CI	Processing Unit clock
6,31,55,68,81	Vdd		+5 volts
4,29,53,66,79	GND		Ground

I=TTL input, SI=schmitt input; CI=clock input; Ox=output; I/Ox=bidirectional; ODx=open drain output. Numerals indicate I_{OL} levels. See the *Electrical Specifications* section for more details.

Figure 2. 100-pin TQFP Pin Connections

144-pin TQFP	Signal	Type	Description
CPU Interface			
21,20,17-14,12-7,5,143-139,137,134-128,126,124-120	DATA31-0	PI/O4	32-bit data bus. (16-bit option)
3-1	ADDR2-0	I	Register select
30	R/W	I	Read/Write
31	CS#	I	Command Strobe
37	IRQ#	OD2	Interrupt output
32	DS#	I	Data Strobe
DMA Interface			
39	SDREQ#	O4	Source FIFO DMA request
41	SDACK#	I	Source FIFO DMA acknowledge
38	STC#	O6/OD6	Source FIFO terminal count
54	DDREQ#	O4	Dest FIFO DMA request
53	DDACK#	I	Dest FIFO DMA acknowledge
55	DTC#	O6/OD6	Dest FIFO terminal count
Compression RAM			
43,22,107-105	CADDR19-15	O6	CRAM Address
104-102	CADDR14-12	O8	CRAM Address
100,98,96-93,91,89-85	CADDR11-0	O6	CRAM Address
76,74-65,63-59	CDATA15-0	I/O4	CRAM Data
109	WE#	O8	Write enable
110	OE#	O8	Output enable
113	LB#	O4	Lower Byte enable
111	UB#	O4	Upper Byte enable
Miscellaneous Signals			
26	RESET#	SI	Reset
57,56	BMODE1-0	I	Bus mode
28	BCLK	CI	Bus clock
24	PCLK	CI	Processing Unit clock
138,125,114,99,90,83,78,77,52,51,46,42,29,25,19,18,6	Vdd		+5 volts
144,136,135,127,119,116,112,108,101,97,92,84,79,75,64,58,50,45,44,40,27,23,13,4	Vss		Ground
118,117,115,82-80,49-47,36-33	NC		No Connection

I=TTL input, SI=schmitt input; CI=clock input; Ox=output; I/Ox=bidirectional; ODx=open drain output. Numerals indicate I_{OL} levels. See the *Electrical Specifications* section for more details.

Figure 3. 144-pin TQFP Pin Connections

2.1 CPU Interface

2.1.1 Address (A2-A0)

Address input signals for the CPU Interface. These inputs are significant only for Register accesses, not for DMA accesses.

2.1.2 Data (D31-D0)

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. They are terminated internally with high impedance pull-ups.

2.1.3 Command Strobe (CS#)

Active low input. While this signal is active, a register access will take place. The data direction is determined by the R/W signal.

2.1.4 Read/Write Command (R/W)

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. See Figure 4 for proper polarity.

2.1.5 Interrupt (IRQ#)

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}

2.1.6 Data Strobe (DS#)

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 as shown in Figure 4. See the AC Timing section for timing details.

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.

2.1.7 Bus Mode (BMODE1-0)

These pins select the timing mode of the CPU interface. See Figure 4 for a summary of the bus modes. Refer to the BCLK pin description for a brief description 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 9711.

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 4. Bus Modes and DS# usage

2.2 DMA Interface

2.2.1 Source FIFO DMA Request (SDREQ#)

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.

2.2.2 Source FIFO DMA Acknowledge (SDACK#)

Active low input. While this signal is active, a DMA write operation will take place to the Source FIFO.

2.2.3 Source FIFO Terminal Count (STC#)

Active low output. This signal can be either an open drain or a totem pole output, depending on the setting of the TC DRIVE bit in the configuration register. The pin is open drain after a reset.

This signal will become active just before and during the last DMA transfer to the Source FIFO of the current command. Any further Source DMA activity will be for the next command in the Command pipeline.

2.2.4 Dest FIFO DMA Request (DDREQ#)

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.

2.2.5 Dest FIFO DMA Acknowledge (DDACK#)

Active low input. While this signal is active, a DMA read operation will take place from the Dest FIFO.

2.2.6 Dest FIFO Terminal Count (DTC#)

Active low output. This signal can be either an open drain or a totem pole output, depending on the setting of the TC DRIVE bit in the configuration register. The pin is open drain after a reset.

This signal will become active just before and during the last DMA transfer from the Dest FIFO of the current command. Any further Dest DMA activity will be from the next command in the Command pipeline.

2.3 Compression RAM Interface

2.3.1 Compression RAM Address Bus (CADDR19-CADDR0)

These output signals select the address of the Compression RAM to be accessed. Either SRAM or DRAM may be used as set in the Configuration register.

If SRAM is used, these 20 address signals may be tied directly to the address bus of the Compression RAM.

If DRAM is used, CADDR11-CADDR0 are attached to the multiplexed address bus of the Compression 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.

2.3.2 Compression RAM Data Bus (CDATA15-CDATA0)

This 16-bit bi-directional data bus for the Compression RAM is used to perform the compression and decompression functions. 16-bit compression RAM must be used.

2.3.3 Compression RAM Write Enable (WE#)

Active low output. If SRAM is used, this signal will become active during each write access to the Compression RAM.

If DRAM is used, this signal must be left unconnected. The CADDR15 signal is used for the DRAM WE#.

2.3.4 Compression RAM Output Enable (OE#)

Active low output. If SRAM is used, this signal will become active during each read access to the Compression RAM.

If DRAM is used, this signal must be left unconnected. The CADDR16 signal is used for the DRAM OE#.

2.3.5 Upper Byte Enable (UB#)

Active low output. This signal will become active during each access to the upper byte of the Compression RAM if SRAM is used.

This signal is not used for DRAM. It must be left unconnected.

2.3.6 Lower Byte Enable (LB#)

Active low output. This signal will become active during each access to the lower byte of the Compression RAM if SRAM is used.

This signal is not used for DRAM. It should be left unconnected.

2.4 Miscellaneous Signals

2.4.1 Reset (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 Compression History should be cleared on its first use. Do not access the chip after RESET#is deasserted until after 12 BCLK + 30 PCLK cycles have occurred.

2.4.2 Processing Unit Clock (PCLK)

This input drives the clock of all the internal logic, including bus interface logic if configured for asynchronous (BCLK=PCLK/2) mode of operation (see below). The clock frequency determines the speed of the Processing Unit. The speed of the Processing Unit is directly linear with the clock frequency. The duty cycle of this clock input must be at least 60/40.

2.4.3 Bus Clock (BCLK)

This input drives the clock of the bus, which determines the speed of the bus. The duty cycle of this clock input must be at least 60/40. There are 4 modes of operation.

The asynchronous mode is used when the system bus timing interface is not relative to any clock. The asynchronous bus interface uses the BCLK for generating bus timing. In this case the input frequency of BCLK must not be greater than 33MHz, and the input frequency of PCLK must not be greater than 66MHz.

The asynchronous (BCLK=PCLK/2) mode uses the PCLK signal internally divided by 2 rather than BCLK. In this case the BCLK pin must be tied high or low. The bus timing is the same as asynchronous mode bus timing.

The synchronous mode is used when the system bus timing interface is relative to a clock. In this case BCLK is used to generate bus interface timing and PCLK is used to generate compression engine timing.

The 68360 bus mode is used when a Motorola 68360 is connected to the 9711. This bus mode utilizes the 68360's CPUCLK signal tied directly to the 9711's BCLK pin and provides 68360 compatible bus timing and PCLK is used to generate compression engine timing.

3 Product Overview

3.1 Source FIFO Data Flow

The Source FIFO will not request any data unless there is an active command and there is room in the Source FIFO for additional data. After a command is issued, the Source FIFO will request data. The request will remain active until the Source FIFO becomes full, or until one of the command termination conditions listed in the Command Stack register description occurs. The most common command termination condition is when the Source Counter reaches zero.

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.

Most of the command termination conditions affect the input side of the Source FIFO. For example, when the number of bytes entering the Source FIFO matches the value set in the SOURCE COUNT field in the Command Stack register, or if the Stop command is issued, the last byte in the Source FIFO at the time the termination condition occurs is allowed to flow through to the output side of the Source FIFO (and through the Processing Unit) before the command actually terminates.

The Source FIFO will stop requesting data for the current command when the number of bytes entering the Source FIFO matches the value set in the SOURCE COUNT field of the Command Stack. The STC# signal will be asserted during the last data transfer into the source FIFO.

If the Source FIFO has accepted the number of bytes specified by the SOURCE COUNT field, and if there is a pipelined command, then the DMA Source Counter (on the input side of the Source FIFO) will be set to the new value of SOURCE COUNT, and data will continue to be requested by the Source FIFO. In other words, the Source FIFO may contain data from two different commands simultaneously. The 9711 will handle this condition automatically. The signal will still be asserted during the last data transfer of the command, even though the request will remain active in this case.

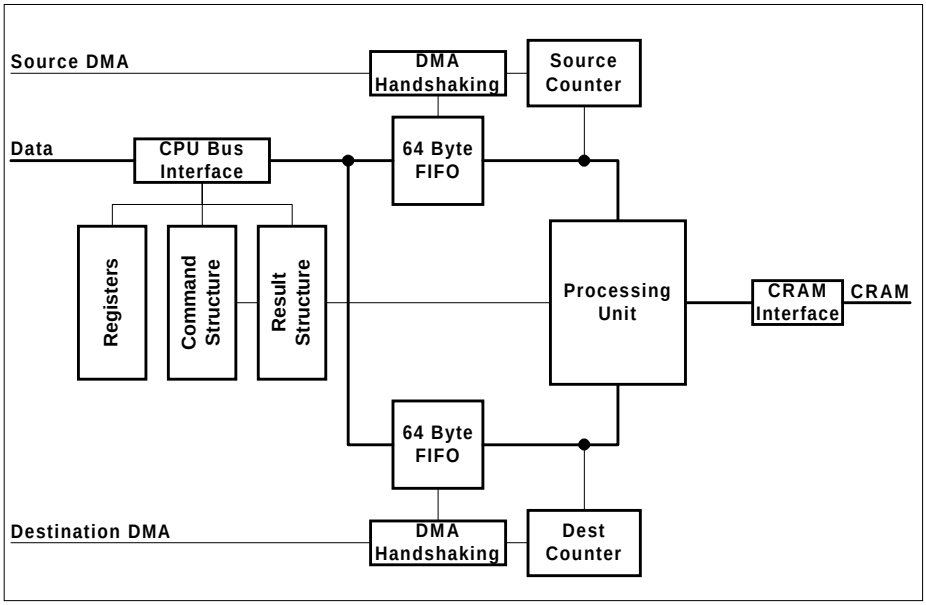


Figure 5. Internal block diagram

When the last byte of a source data for a given command exits the Source FIFO and enters the Processing Unit (as indicated by the Source Counter on the output side of the Source FIFO reaching zero), any trailing bytes that are not 32-bit (or 16-bit) aligned will be discarded. The first byte of data for the next command will come from the next 32-bit (or 16-bit) aligned value.

3.2 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 a command has entered the Dest FIFO, the Dest FIFO will request data transfers, independent of the FIFO threshold set by the FIFO Configuration register, until the FIFO is empty. The DTC# signal will be asserted during the transfer of the last byte out of the Dest FIFO.

After the last byte of data from a command has entered the Dest FIFO, the next pipelined command may begin to transfer data to the Dest FIFO immediately. In other words, the Dest FIFO may contain data from two different commands simultaneously. The 9711 will handle this condition automatically. The DTC# signal will still be asserted during the transfer of the last byte of the current command, even though the request will remain active in this case.

When the last byte of a data enters the Dest FIFO from the Processing Unit, the data is padded with dummy data to fill out an entire 32-bit (or 16-bit) value. The value of any dummy byte is undefined. The first byte of data from the next command will be 32-bit (or 16-bit) aligned.

3.3 Command/Result Pipeline

In order to eliminate latency while issuing commands and reading results, the command and result registers are architected into a multi-stage pipeline. Commands and results are pipelined to allow the CPU to issue commands and read results without severe timing constraints.

The pipeline consists of three stages. The first stage is the Command Stack. The second stage is the Processing Unit. The third stage is the Result Stack
When a command is written to the chip, it is written to the Command Stack in the pipeline. If the Processing Unit is available (no command is active), the command will be transferred to the Processing Unit. When a command is trans-

ferred to the Processing Unit, the Command Stack is ready to accept another command. The COMMAND READY bit in the Status register will be set to one.

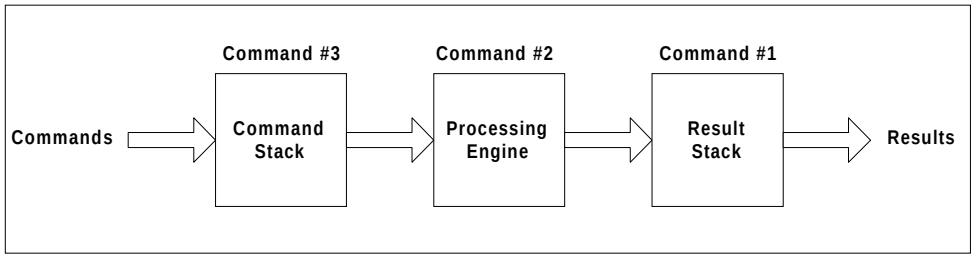


Figure 6. Command/Result Pipeline

Once the Processing Unit has completed the command, the result information will be transferred to the Result Stack. The RESULT READY bit in the Status register will be set to one. When a result is transferred to the Result Stack, the Processing Unit is ready to accept another command from the Command Stack.

After detecting the RESULT READY bit (via an interrupt, or polling), the CPU may read the result from the Result Stack.

The three stages allow buffering of up to three operations. The pipeline allows the CPU ample time to issue commands and read results without slowing down the Processing Unit.

3.4 Compression RAM

The 9711 utilizes 16-bit RAM to maintain history information for each full-duplex channel of data. Either SRAM or DRAM may be used for this storage. At a 66MHz PCLK speed, a 12ns or faster SRAM must be used or a 60ns or faster EDO DRAM must be used. The 9711 only supports EDO DRAM. The PCLK speed may be adjusted to accommodate slower RAM speeds. This will affect compression and decompression performance.

Each full-duplex compression/decompression history requires 16Kbytes of RAM storage in LZS mode and 32Kbytes in MPPC mode. When using the maximum of 2Mbytes of SRAM, 128 LZS (or 64 MPPC) full-duplex histories may be maintained. When using the maximum 32Mbytes of DRAM, up to 2048 LZS (or 1024 MPPC) full-duplex histories may be maintained.

Please refer to Figure 11 for selecting the proper DRAM configuration. Suitable DRAMs must support 16-bit data bus widths and byte-write capability with single WE- and upper and lower byte CAS-.

4 Register Description

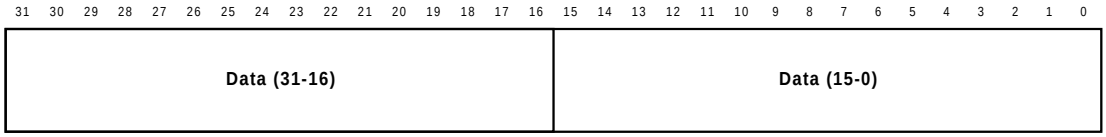
4.1 Overview

The 9711 uses 8 address locations for configuration and use. All 9711 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
Command Stack	1
Result Stack	2
Configuration	3
Interrupt Enable	4
Status	5
FIFO Status	6
FIFO Configuration	7

Figure 7. Register list

4.2 Data (0)



This register can be read or written. It is used to transfer data to the Source FIFO and 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.

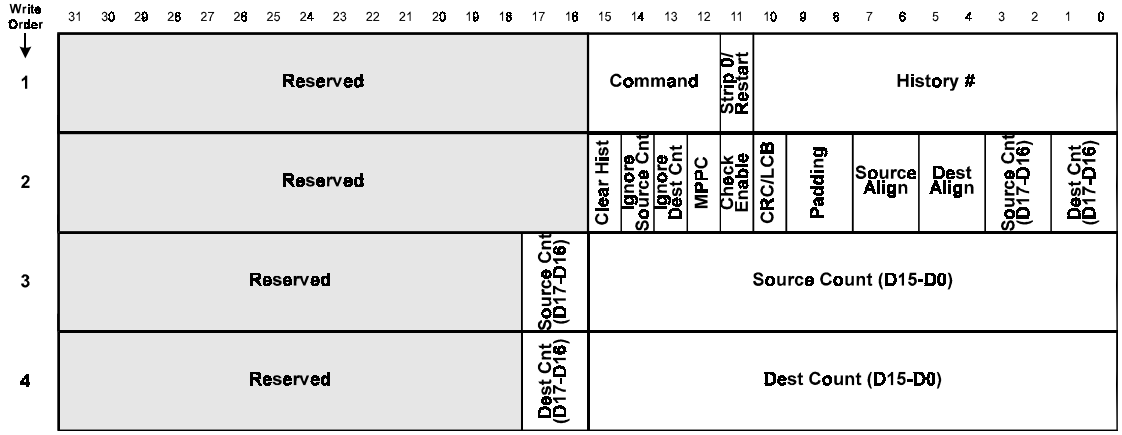
A CPU bus write operation will transfer four (or two) bytes to the Source FIFO (depending on the data bus width). A CPU bus read operation will transfer four (or two) bytes from the Dest FIFO.

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.

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

If too much data is written, or too much data is read, the command will terminate, and the DATA ERROR bit in the Result Stack and in the Status register will be set to one. This error will produce undefined results in the data stream, so the data should be ignored, and the associated history should be cleared.
This register can be read or written. It is used to specify a command to be executed. In addition, several parameters related to the command are set here.

4.3 Command Stack (1)



This register operates as a stack. Four write operations are required to properly start a command. The COMMAND IN PROGRESS bit in the Status register may be used to verify the synchronization of write operations. All four writes must take place unless specified differently in the command description.

A new command can only be issued if the COMMAND READY bit in the Status register is set to one before the first write to the Command register.

Most commands will produce result information when they are completed. The result information is read from the Result Stack.

See the Command/Result Pipeline section for further details of the operation of the Command/Result pipeline.

Although not normally required, the command stack may be read by the CPU. This may be done for testing purposes. The last command written may be read after written. All four words must be written to the command stack before the Command Stack may be read. The COMMAND IN PROGRESS bit in the Status register may also be used to verify the synchronization of read operations.

4.4 Command Termination Conditions

All the commands (except the Reset command) will terminate normally when any of the following conditions occur. A “normal termination” means that the Results Stack contains valid status from the previously executed command, as indicated by the RESULT READY bit in the Status register. The Reset command does not terminate normally, and causes an immediate termination and results are not generated.

1) Source Counter reaches zero (unless the IGNORE SOURCE CNT bit is set to one or the currently executing command is Read CRAM). In the Result Stack, the SOURCE DONE bit will be set to one. The Read CRAM command terminates when the Dest counter reaches zero.

2) The Stop command is issued. The currently executing command will terminate when the last byte currently in the Source FIFO at the time the Stop command is issued, enters the Processing Unit. At command termination,

the Result Stack contains the results of the currently executing command. The Stop command generates no results of its own.

3) The Reset command is issued. There will be no Result Stack. Everything stops immediately, and the data in the FIFOs will be discarded. The associated history should be reset.

4) An attempt is made to write data to a full Source FIFO or an attempt is made to read data from an empty Dest FIFO. In the Result Stack, the DATA ERROR bit will be set to one. The command will terminate when the last byte in the Source FIFO at the time of the data error, enters the Processing Unit. The associated history should be reset.

Note: The Dest Counter cannot terminate a command. If the Dest Counter decrements below zero during the operation of any command, the command will continue to process incoming data, but no additional data will be output to the Dest FIFO. The DEST OVERRUN bit in the Result Stack will be asserted in this case.

4.5 Commands

The COMMAND field specifies the command to be executed. The valid values are listed in Figure 8. All other values are invalid.

Command	Command field
Compress	0
Decompress	1
Update History	2
Pass-Through	3
Reset	4
Stop	5
Read CRAM	6
Write CRAM	7

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

Figure 8. Commands

Compress
This command will compress a block of data using the Compression History specified in the HISTORY # field. The compression algorithm used will be either LZS or MPPC, depending on the MPPC bit.

At termination, the Processing Unit will flush out its internal data. If in LZS mode, an End Marker will be appended to the compressed data stream.
This command will optionally append padding and an LCB or CRC to the compressed data stream as set by the other options in this command.

Decompress
This command will decompress a block of data using the Decompression History specified in the HISTORY # field. The decompression algorithm used will be either LZS or MPPC, depending on the MPPC bit.

In LZS mode, if the End Marker is found before the command terminates, all the source data between the End Marker and the last source byte (except the check

field, if there is one) 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 last source byte must be the last byte produced by the Compress command. Otherwise the Decompression History will become invalid.

In LZS mode, this command will optionally analyze the compressed data stream by stripping padding and verifying LCB or CRC as set by the other options in this command.

Update History
This command will pass a block of uncompressed data from the source to the destination without modifying the data in any way. The Decompression History specified in the HISTORY # field will be updated with the uncompressed source data. The update function used will be either LZS or MPPC, depending on the MPPC bit.

Pass-Through
This command will pass a block of data from the source to the destination without modifying the data in any way. No Compression or Decompression History is affected.

This command is mainly used for debugging purposes.

Reset
This command will cause the chip to immediately enter a known start-up state. This is a single word command. Any additional writes to the Command Stack register will be interpreted as the start of the next command. The effect of a Reset command is identical to asserting the RESET pin, except that the Configuration register will not be initialized.

When this command is issued, all chip operations will immediately Stop. Any pipelined commands will be discarded. The Source and Dest FIFOs will be cleared. Any intermediate information still residing in the chip will be lost. If a Compress operation is stopped in this manner, the Compression History will be corrupt and must be cleared next time it is used.

This command is not pipelined, and will operate immediately. The COMAND IN PROGRESS bit in the Status register must be zero when this command is issued. No result data is generated by this command.

Do not access the chip after a Reset command is issued until after 12 BCLK + 30 PCLK cycles have occurred.

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

The CRAM starting address is combined in the HISTORY # field and the SOURCE COUNT field. CRAM address bits 24-14 are set by the entire HISTORY # field. CRAM address bits 13-1 are set by bits 13-1 of the SOURCE COUNT field. Bit 0 of the SOURCE COUNT field must be zero.

The DEST COUNT field determines the number of bytes to read. Since the HISTORY # field does not increment, CRAM address bits 24-14 are fixed and cannot be incremented. So, the value of SOURCE CNT, DEST COUNT, and HISTORY # fields must be chosen to prevent the CRAM address bits 24-14 from incrementing. Bit 0 in the DEST COUNT field must be set to zero. The value of the entire DEST COUNT field must not be set to zero.

Write CRAM
This command is used to write data directly to the CRAM (Compression RAM). This is normally used for debugging purposes or for CRAM memory testing.

The CRAM starting address is combined in the HISTORY # field and the DEST COUNT field. CRAM address bits 24-14 are set by the entire HISTORY # field. CRAM address bits 13-1 are set by bits 13-1 of the DEST COUNT field. Bit 0 of the DEST COUNT field must be zero.

The SOURCE COUNT field determines the number of bytes to write. Since the HISTORY # field does not increment, CRAM address bits 24-14 are fixed and cannot be incremented. So, the value of SOURCE COUNT, DEST COUNT, and HISTORY # fields must be chosen to prevent the CRAM address bits 24-14 from incrementing. Bit 0 in the SOURCE COUNT field must be set to zero. The value of the entire SOURCE COUNT field must not be set to zero.

Stop
This command will cause the currently executing Compress, Decompress, Update History, or Pass-Through command to terminate as soon as the last byte currently in the Source FIFO at the time the Stop command is issued, enters the Processing Unit. All normal termination processing will take place. This is a single word command. Any additional writes to the Command Stack register will be interpreted as the start of the next command.

This command will not affect any command residing in the pipeline. If there is no command currently operating, then the Stop command will be ignored.

This command is useful to terminate a command if the IGNORE SOURCE COUNT bit is set.

This command is not pipelined, but will operate immediately. No result data is generated by this command (except for the result data of the command that is being stopped).

Typical use of this command assumes that there are no data transfers occurring (for example, DMA).

4.6 Command Stack Fields

4.6.1 Strip 0/Restart

This bit has two functions depending on the mode of operation of the 9711. 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. 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.

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 9711 must know the exact number of source bytes so that it can insert the zero at correct location in the data stream. The IGNORE SOURCE COUNT bit in the Command Stack cannot be used, and the Stop command cannot be used.

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/RESTART bit is set to zero, this feature will be disabled.

In MPPC mode the STRIP 0/RESTART bit is used to signal the decompression engine to move the packet to the front of the history buffer, 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 history buffer. If this bit is set to zero the packet is not moved to the front of the history buffer.

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

4.6.2 History

This field is utilized in the Compress, Decompress, and Update History commands. This field specifies the history number to be used. The memory organization depends on the state of the MPPC bit, as explained below. Do not specify a history number that is not supported by the size of the Compression RAM.

LZS Mode
When the LZS format is selected for this history (the MPPC bit is set to zero), each full duplex history requires 16 Kbytes of Compression RAM (CRAM). Although an individual Compression History is functionally independent of a Decompression History, the associated memory areas are allocated together in a single 16 Kbyte memory block.

History type	History # field value	Actual CRAM location
1st compression history	0	0
1st decompression history	0	0
2nd compression history	1	16K
2nd decompression history	1	16K

Figure 9. Example LZS CRAM usage

MPPC Mode
When the MPPC format is selected for this history (the MPPC bit is set to one), each half-duplex history requires 16 Kbytes of Compression RAM (CRAM). Each Compression History requires 16 Kbytes, and each Decompression History requires an additional 16 Kbytes. Both cannot exist in a single 16 Kbyte block of memory.

History type	History # field value	Actual CRAM location
1st compression history	0	0
1st decompression history	1	16K
2nd compression history	2	32K
2nd decompression history	3	48K

Figure 10. Example MPPC CRAM usage

4.6.3 Clear History

This bit is utilized in the compress command for LZS and MPPC modes and the decompress command in the MPPC mode. If this bit is set to one, the compression (or decompression) history specified by the History # field will be cleared before the operation begins.

If this bit is set to zero, the compression history will not be cleared. The compress operation will use history information from previous compress operations which used the same history number.

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

Due to the nature of the LZS compression format, unless there are data corruption or data loss errors, 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.

4.6.4 Ignore Source Count

This bit is utilized in the Compress, Decompress, and Update History commands only in LZS mode, or in the Pass-Through command. If this bit is set to one, the Source Counter will not terminate the command when the counter reaches zero. The Source Counter will wrap from zero to 0x3FFFF.

If this bit is set to zero, the specified command will terminate when the Source Counter reaches zero.

In either case, the Source Counter will be initialized to the value set in the SOURCE COUNT field. Also, the command may terminate from other conditions listed previously.

4.6.5 Ignore Dest Count

This bit is utilized in the Compress, Decompress, and Update History commands in either LZS or MPPC modes, or in the Pass-Through command. If this bit is set to one, the Dest Counter will not cause the Processing Unit to stop producing data when the counter reaches zero. The Dest Counter will wrap from zero to 0x3FFFF.

If this bit is set to zero, the Processing Unit will stop producing data when the Dest Counter reaches zero. Also, the dest counter will not decrement further. In either case, the Dest Counter will be initialized to the value set in the DEST COUNT field.

4.6.6 MPPC

This bit is utilized for the Compress , Decompress, and Update History commands. It selects the compression format used for operation. 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.

Each history must be initialized for a particular compression algorithm, either LZS or MPPC. This bit must be used consistently for each history number, until the history is initialized.

This bit also affects the compression RAM memory map as defined in the description of the History field.

4.6.7 Check Enable

This bit is utilized only in the Compress and Decompress commands. It enables the in-line generation and verification of a check field (with optional padding in LZS mode). If this bit is set to zero, there will be no padding or in-line check fields generated or verified. The CHECK ENABLE bit does not affect the calculation of the CHECK FIELD in the Result Stack register.

For a Compress command in either LZS or MPPC modes, if this bit is set to one, padding (as defined by the PADDING field) and a check field (as defined by the CRC/LCB field) will be generated and appended to the compressed destination data stream. If the check field is set to be a CRC, the low byte of the CRC is appended first, followed by the high byte.

For a Decompress command, if this bit is set to one in LZS mode all data between the End Marker and the check field will be ignored, and the check field will be verified. For a Decompress command, if this bit is set to one in MPPC mode the last 1 or 2 bytes of data (the check field) will be verified.

If the check value calculated for the current block of data does not match the check field embedded in the data stream or if the check value is missing from the data stream, the CHECK ERROR bit in the Result Stack will be set to one. Also, the

CHECK ERROR bit in the Status register will be set to one (and the associated interrupt may be generated).

4.6.8 CRC/LCB

This bit is utilized in the Compress, Decompress, Update History, and Pass-Through commands. It selects between the use of an LCB or a CRC check field. This affects the value reported in the Result Stack as well as the value optionally inserted or checked in the data stream.

If this bit is set to zero, an 8-bit LCB (Longitudinal Check Byte) will be used. The LCB is a byte-wise exclusive-OR sum of the uncompressed data. The LCB is initialized to FF_{16} before each compress or decompress operation.

If this bit is set to one, a 16-bit CRC will be used. The CRC is calculated on the uncompressed data. The CRC is initialized to $FFFF_{16}$ before each compress or decompress operation. The CRC polynomial is

$$x^{16} + x^{12} + x^5 + 1.$$

4.6.9 Padding

In the LZS mode (the MPPC bit is set to zero) this field is utilized in the Compress and Decompress commands only if the ENABLE CHECK bit is set to one. This field determines the number of bytes inserted or removed from the data stream between the last byte of compressed data, and the selected check field (LCB or CRC). In MPPC mode (the MPPC bit is set to one) the PADDING field is reserved and must be set to zero.

If the PADDING field is set to zero, no byte padding will be generated or stripped. The check field will immediately follow the last byte of compressed data.

For a Compress command, if the PADDING field is set to one, zero or one byte (with the value of zero) will be inserted between the last byte of compressed data and the check field so that the end of the check field is coincident with the end of a word (16-bit) boundary.

For a Compress command, if the PADDING field is set to 2, zero, one, two, or three bytes (with the value of 0x00) will be inserted between the last byte of compressed data and the check field so that the end of the check field is coincident with the end of a double-word (32-bit) boundary.

For a Decompress command, if the PADDING field is set to one or two, all the bytes between the last byte of compressed data and the check field will be discarded. In this case, the 9711 must know the exact number of source bytes so that it can locate the check field. The IGNORE SOURCE COUNT bit in the Command Stack cannot be used, and the Stop command cannot be used.

The PADDING value of 3 is reserved.

4.6.10 Source Align

This field is utilized in the Compress, Decompress, Update History, and Pass-Through commands. 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 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.

4.6.11 Dest Align

This field is utilized in the Compress, Decompress, Update History, and Pass-Through commands. 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.

4.6.12 Source Count

This field is utilized in all commands except the Reset and Stop commands. This is the initial value used for the Source Counter. The Source Counter is decremented for each source byte processed by the Processing Unit. The initial value for MPPC must not be greater than 8192 plus the number of any source-aligned bytes.

Source Count will decrement for every byte processed, including bytes specified by the SOURCE ALIGN field, data, padding, and a check value (if enabled). It will not decrement for any extra bytes inserted due to non-word alignment in the last transfer.

In 32-bit mode (as set by the BUS WIDTH bit in the Configuration register), all bits of the SOURCE COUNT may be set in the 3rd word of the Command Stack. In 16-bit mode, D17-D16 cannot be written in the 3rd word, and these bits are assumed to be zero. In this case, D17-D16 may be set in the SOURCE COUNT bits in the 2nd word of the Command Stack. In general, the Source Counter bits D17-D16 will be set to the logical OR of both D17-D16 bits in the Command Stack.

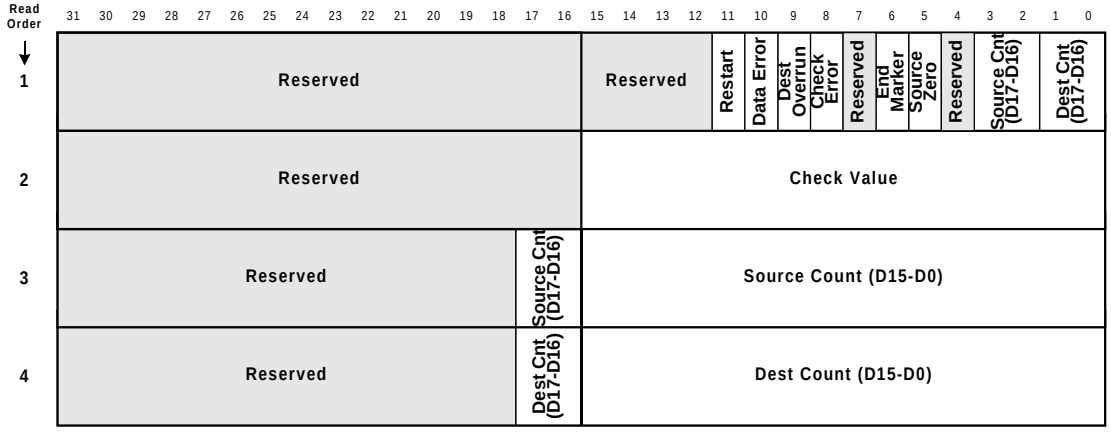
4.6.13 Dest Count

This field is utilized in all commands, except the Reset and Stop commands. This is the initial value used for the Dest Counter. The Dest Counter is decremented for each destination byte produced by the Processing Unit. The initial value for MPPC must not be greater than 8192 plus the number of any dest-aligned bytes.

Dest Count will decrement for every byte processed, including bytes specified by the DEST ALIGN field, data, padding, and a check value (if enabled). It will not decrement for any extra bytes inserted due to non-word alignment in the last transfer.

In 32-bit mode (as set by the BUS WIDTH bit in the Configuration register), all bits of the DEST COUNT may be set in the 3rd word of the Command Stack. In 16-bit mode, D17-D16 cannot be written in the 3rd word, and these bits are assumed to be zero. In this case, D17-D16 may be set in the DEST COUNT bits in the 2nd word of the Command Stack. In general, the Dest Counter bits D17-D16 will be set to the logical OR of both D17-D16 bits in the Command Stack.
This register may only be read. It is used to read the result of a completed operation.

4.7 Result Stack (2)



This register operates as a stack. Four read operations are required to properly read the result. The RESULT IN PROGRESS bit in the Status register may be used to verify the synchronization of read operations. All four reads must take place even if some of the information will not be used.

This register must only be read when the RESULT READY bit in the Status register is set, which occurs after the currently executing command terminates and the last data byte enters the dest FIFO.

See the Command/Result Pipeline section for further details of the operation of the Command/Result pipeline.

4.7.1 Restart

This bit is significant for only the Compress command in MPPC mode only. This bit is used to confirm that the compression engine moved the packet to the front of the history buffer, as specified in the MPPC protocol. This bit applies to the Compression command when the MPPC bit is set in the Command Stack. If this bit is set to one the packet was moved to the front of the history buffer. If this bit is set to zero the packet was not moved to the front of the history buffer.

4.7.2 Data Error

This bit is significant for all commands. This bit will be set to one if an attempt is made to write data to a full Source FIFO or an attempt is made to read data from an empty Dest FIFO.

This error condition will cause the automatic execution of a Stop command. The current command will terminate as soon as the Source FIFO becomes empty.

See the Stop command description in the Command Stack for additional information. Also, any pipelined commands will be cleared. No new commands may be written until the DATA ERROR bit in the status register is cleared.

This condition may be one reason why the operation terminated.

4.7.3 Dest Overrun

If this bit is set to one, the command produced more data than expected. The Dest Counter attempted to decrement below zero, and the IGNORE DEST COUNT bit was set to zero. This condition does not terminate a command, but no additional data was output to the Dest FIFO. Any further source data is discarded until the command terminates. The Dest Counter will remain at zero.

If this bit is set to zero, then the Dest Counter did not attempt to decrement below zero, or the IGNORE DEST COUNT bit was set to one.

4.7.4 Check Error

This bit is significant for the Decompress command only, and only if the CHECK ENABLE bit was set in the Decompress command. If the CHECK ERROR bit is set to one, the check value calculated for the current block of data did not match the check field embedded in the data stream, or the check value is missing from the data stream (due to the Source Count reaching zero prematurely, for example).

If this bit is zero, the check value matched correctly.

4.7.5 End Marker

This bit is significant for the Decompress command only 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.

4.7.6 Source Zero

This bit is significant for the Compress, Decompress, Update History, Write CRAM and Pass-Through commands. If this bit is set to one, the Source Counter reached zero. This condition may be one reason why the operation terminated.

If this bit is set to zero, the Source Counter did not reach zero before the operation terminated, or the IGNORE SOURCE COUNT bit was set to one.

4.7.7 Check Value

This field is significant for the Compress, Decompress, Update History, and Pass-Through commands. This is the calculated LCB or CRC (as set by the CRC/LCB bit in the Command Stack). If configured for LCB, then only bits 7 to 0 are significant. The other bits are unused. If configured for CRC, then bits 15 to 0 are significant.

This field is significant even if the CHECK ENABLE bit in the Command Stack register is set to zero.

4.7.8 Source Count

This field is significant for all commands, except the Reset and Stop commands. This is the final value of the source counter at command termination. The source counter is decremented for each source byte processed by the Processing Unit.

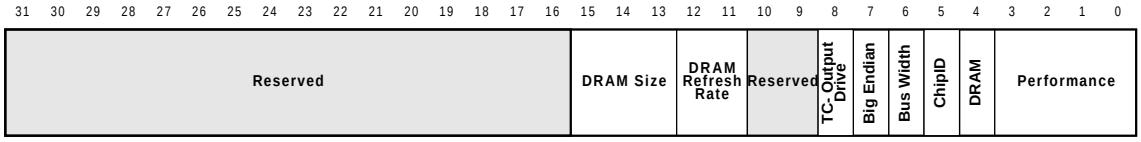
In 32-bit mode (as set by the BUS WIDTH bit in the Configuration register), bits D17-D16 of the Source Counter will be present both in the first and third words of the Result Stack. In 16-bit mode, bits D17-D16 will be present only in the first word of the Result Stack.

4.7.9 Dest Count

This field is significant for all commands except the Reset and Stop commands. This is the final value of the Dest counter at command termination. The Dest counter is decremented for each destination byte produced by the Processing Unit.

In 32-bit mode (as set by the BUS WIDTH bit in the Configuration register), bits D17-D16 of the Dest Counter will be present both in the first and fourth words of the Result Stack. In 16-bit mode, bits D17-D16 will be present only in the first word of the Result Stack.

4.8 Configuration (3)



This register may be read or written. It is used to configure chip options. The default value of all the fields in this register after a hardware reset is zero, except the PERFORMANCE field which defaults to a value of 8.

4.8.1 DRAM Size

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

Value	Size (16-bit words)	Rows	Columns
0 0 0	256K	9	9
0 0 1	512K	10	9
0 1 0	1M	10	10
0 1 1	2M	11	10
1 0 0	4M	11	11
1 0 1	8M	12	11
1 1 0	16M	12	12
1 1 1	Reserved		

Figure 11. EDO DRAM size values

4.8.2 DRAM Refresh

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

Value	Divisor
0 0	1024
0 1	512
1 0	256
1 1	Reserved

Figure 12. Refresh frequency

4.8.3 TC- Drive

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

If this bit is zero, then the outputs are open drain drivers. If this bit is one then the outputs are totem-pole drivers. The default is open drain drivers.

4.8.4 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 9711 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 9711 data bus will enter or leave the Processing Unit first.

4.8.5 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 registers affected by the setting of this bit are:

1) Data register.

2) Command Stack. The SOURCE COUNT and DEST COUNT fields.

3) Result Stack. The SOURCE COUNT and DEST COUNT fields.

4.8.6 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.

4.8.7 DRAM

This bit selects the RAM type used for the Compression 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 9711 compression and decompression performance is affected by the setting of the DRAM bit due to the memory speed differences. Figure 13 shows the difference in 9711 performance for a PCLK input of 66MHz under the different memories.

Compression RAM	compress (Mbytes/s)	decompress (Mbytes/s)
12ns SRAM	8	15
60ns EDO DRAM	2.5	5

Figure 13. 9711 performance

4.8.8 Performance

This field sets the compression performance for both the LZS or MPPC compression algorithms. 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 14 illustrates the effect of the PERFORMANCE field when compressing a text file containing the Constitution of the United States when the algorithm is LZS and the compression RAM is SRAM. Figure 15 illustrates the effect of the PERFORMANCE field when compressing a text file containing the Constitution of the United States when the algorithm is LZS and the compression RAM is DRAM.

9711 decompression speed is 15Mbytes/s with SRAM and 5MBytes/sec with DRAM. Some full-duplex applications require both compression and decompression operations to be performed simultaneously. These 9711 performance for these types of applications may be calculated, and is explained in the "9710/9711 SRAM vs. DRAM Full-Duplex Performance" application note (APP-0026).

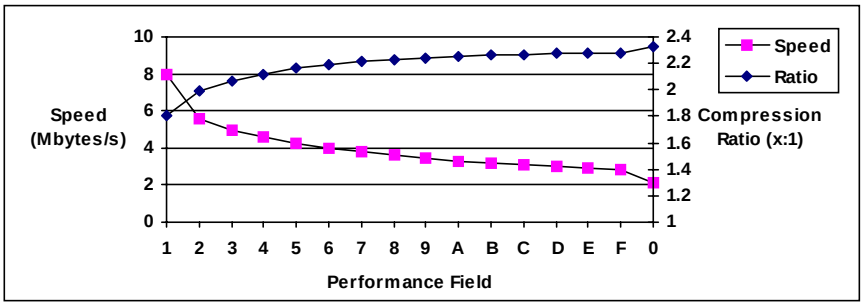


Figure 14. Effect of LZS Compression Performance field with SRAM

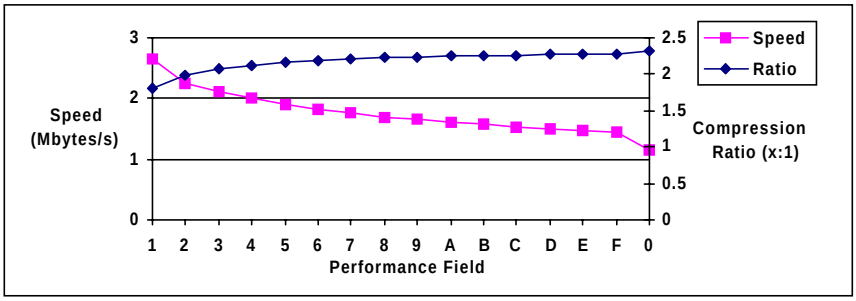
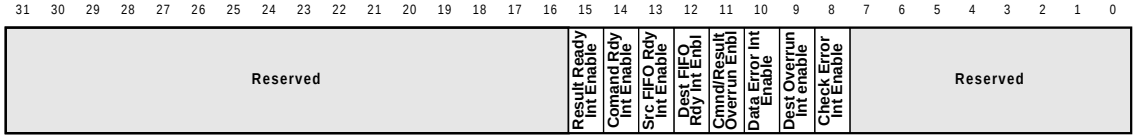


Figure 15. Effect of LZS Compression Performance field with DRAM

4.9 Interrupt Enable (4)



This register may be read or written. It is used to configure the chip. The default value of all the fields in this register after a reset is zero.

4.9.1 Result Ready Interrupt Enable

While this bit is set to one, the IRQ# signal will be asserted while the RESULT READY bit in the Status register is set to one.

4.9.2 Command Ready Interrupt Enable

While this bit is set to one, the IRQ# signal will be asserted while the COMMAND READY bit in the Status register is set to one.

4.9.3 Source FIFO Ready Interrupt Enable

While this bit is set to one, the IRQ# signal will be asserted while the SOURCE FIFO READY bit in the Status register is set to one.

4.9.4 Dest FIFO Ready Interrupt Enable

While this bit is set to one, the IRQ# signal will be asserted while the DEST FIFO READY bit in the Status register is set to one.

4.9.5 Command/Result Overrun Interrupt Enable

While this bit is set to one, the IRQ# signal will be asserted while the COMMAND/RESULT OVERRUN bit in the Status register is set to one.

4.9.6 Data Error Interrupt Enable

While this bit is set to one, the IRQ# signal will be asserted while the DATA ERROR bit in the Status register is set to one.

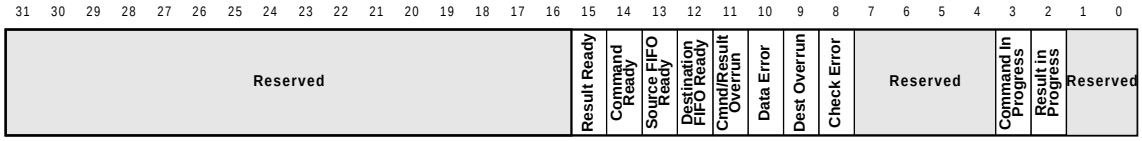
4.9.7 Dest Overrun Interrupt Enable

While this bit is set to one, the IRQ# signal will be asserted while the DEST OVERRUN bit in the Status register is set to one.

4.9.8 Check Error Interrupt Enable

While this bit is set to one, the IRQ# signal will be asserted while the CHECK ERROR bit in the Status register is set to one.

4.10 Status (5)



This register may be read or written. It is used to monitor the status of the chip. The default value of the bits in this register are the valid values for a quiescent chip, and are listed under each bit description.

4.10.1 Result Ready

This bit is set to one when a result is ready to be read from the Result Stack.

This bit may be cleared by writing a one to this bit. This bit may also be cleared by reading the first word of the Result Stack. When cleared, this bit will not be set to one again until the result of the next command is ready to be read (after reading the four words of the current result).

The default value of this bit is zero.

4.10.2 Command Ready

This bit is set to one when the Command Stack is ready to accept a new command.

This bit may be cleared by writing a one to this bit. This bit may also be cleared by writing the first word to the Command Stack. When cleared, this bit will not be set to one again until the chip is ready to accept another new command (after writing all four words of the current command).

The default value of this bit is one.

4.10.3 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 remain set until it is cleared by writing a one to this bit. The default value of this bit is zero.

4.10.4 Dest FIFO Ready

This bit is set to one when the number of bytes in the Dest FIFO exceeds the threshold programmed in the FIFO Configuration register.

This bit will remain set until it is cleared by writing a one to this bit. The default value of this bit is zero.

4.10.5 Command/Result Overrun

This bit is set to one if the Command Stack is written when it is not ready, or the Result Stack is read when it is not ready.

This error condition will not affect any command in operation, or any command successfully pipelined. However, no new commands may be written until this status bit is cleared.

This bit may be cleared by writing a one to this bit. The default value of this bit is zero.

4.10.6 Data Error

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

This error condition will cause the automatic execution of a Stop command. The current command will terminate as soon as the Source FIFO becomes empty. See the Stop command description in the Command Stack for additional information. Also, any pipelined commands will be cleared. No new commands may be written until this status bit is cleared.

The DATA ERROR bit in the Result Stack will also be set to one when reading the Result Stack of the currently executing operation.

This bit may be cleared by writing a one to this bit. The default value of this bit is zero.

4.10.7 Dest Overrun

This bit is set to one if the command produced more data than specified by the DEST COUNT field of the Command Stack, and the IGNORE DEST COUNT bit was set to zero. When the Dest Counter attempted to decrement below zero, the command continued to process incoming data, but no additional data was output to the Dest FIFO. This did not cause the command to terminate. The Dest Counter will remain at zero.
This bit may be cleared by writing a one to this bit. The default value of this bit is zero.

4.10.8 Check Error

This bit is set to one if the check value calculated for the current block of data does not match the check field embedded in the data stream.

This bit may be cleared by writing a one to this bit. The default value of this bit is zero.

4.10.9 Command In Progress

This bit indicates that a command is currently in the middle of being written or read. This bit becomes set to one after the first access to the Command Stack register. This bit returns to zero after the last (fourth) access to the Command Stack register. The default value of this bit is zero.

4.10.10 Result In Progress

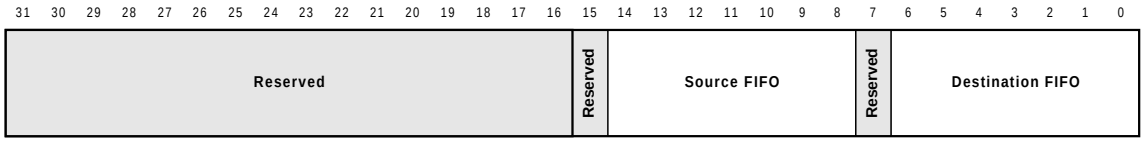
This bit indicates that a result is currently in the middle of being read. This bit becomes set to one after the first read from the Result Stack register. This bit returns to zero after the last (fourth) read from the Result Stack register. The default value of this bit is zero.

4.10.11 Chip ID

When the ChipID bit (bit 5) in the Configuration register is set, this register returns the Chip ID value 0x01XX. The upper 8 bits are defined as the product ID code and the lower 8 bits are reserved.

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

4.11 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 (e.g. CPU PIO), or for testing.

While in the asynchronous bus modes (BMODE=00 or 01), the value of this register is latched on each access to the 9711. The value read is always the previously latched value. Therefore, if it is unknown when the previous access to the 9711 took place, the user should read the value of this register twice, and use only the second value.

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

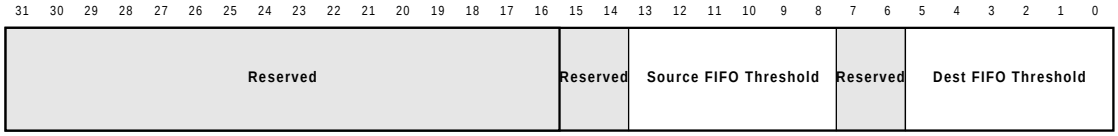
4.11.1 Source FIFO

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.

4.11.2 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.

4.12 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 one. 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.

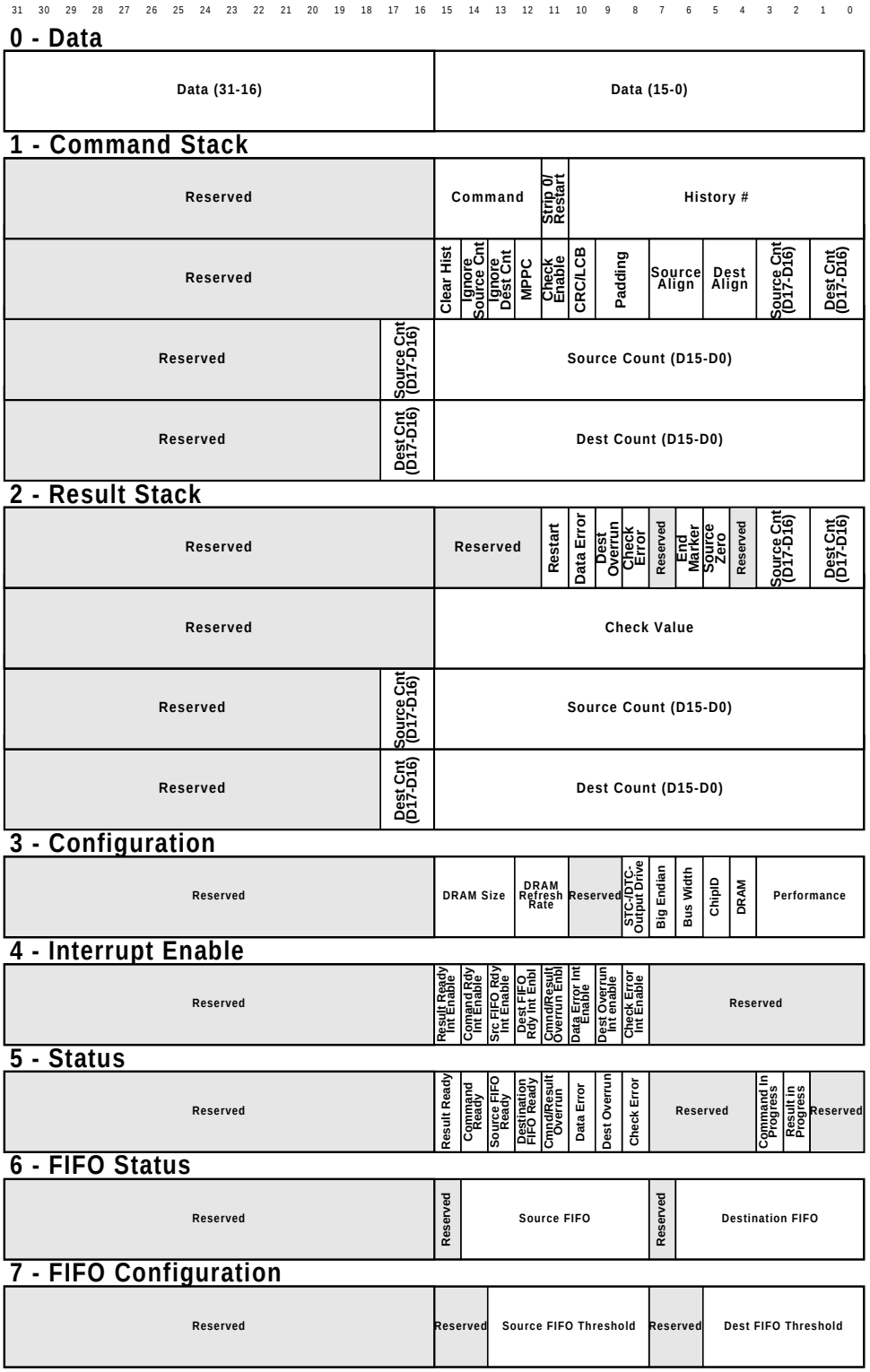
4.12.1 Source FIFO Threshold

This field sets the Source FIFO Threshold. When the number of bytes available to be written to the Source FIFO is greater than the value set in this field, the SOURCE FIFO READY bit in the Status register will be set to one and 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.

4.12.2 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 Register Summary



6 Timing Descriptions

6.1 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#.

6.1.1 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 9711 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. If DS# is not used, it must be tied low.

6.1.2 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 9711, 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 9711 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 9711 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 9711 will assert the $\overline{TC}$ signal after the trailing edge of DACK# on the second to last DMA transfer. The 9711 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. If the DS# is not used, it must be tied low.

6.1.3 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.

6.1.4 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 by the end of a clock. The following clock cycle will be either T2 or Tw based on the value of the DACK# signal at the end of the following cycle. If DACK# is active, then this cycle will be Tw. If DACK# 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 following clock cycle will be either T2 or Tw based on the value of the DACK# signal at the end of the following cycle. If DACK# is active, then this cycle will be Tw. If DACK# 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 DACK# signal at the end of the following cycle. If DACK# is active then this cycle will be T1. If DACK# is 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 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). 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.

6.1.5 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 9711 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. 9711 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.

6.1.6 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 9711.

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 9711 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 CSX# or CAS# deasserts. The 9711 will latch data on the trailing edge of DACK#.

The 9711 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 9711 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 9711 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.

7 Electrical Specifications

DC Supply Voltage (V_{DD})	-0.3V to +6.0V
DC Input Voltage	-0.3V to V_{DD} + 0.3V
DC Input Current	±10mA
Storage Temperature	-40°C to +125°C

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

Figure 16. Absolute maximum ratings

DC Supply Voltage	+4.75V to +5.25V
Operating Temperature	0°C to +70°C

Figure 17. 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)		$V_{IN} = V_{SS}$ $V_{DD} = 5.25V$	-10			µA
		With pullup (PI)		-40		-5	µA
I_{IH}	High level input current (I, SI)		$V_{IN} = V_{DD}$ $V_{DD} = 5.25V$			10	µA
V_{OL}	Low level output voltage		$V_{DD} = 4.75V$				
		(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		$V_{DD} = 4.75V$				
		(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} $V_{DD} = 5.25V$	-10		10	µA
I_{DD}	Quiescent supply current					300	µA
C_{IN}	Input capacitance		$V_{DD} = 5.0V$		2.4		pF
C_{OUT}	Output capacitance		$V_{DD} = 5.0V$		5.6		pF
$C_{I/O}$	I/O capacitance		$V_{DD} = 5.0V$		6.6		pF
P_A	Power dissipation		$V_{DD} = 5.25V$		0.5	1.0	W

Figure 18. DC electrical characteristics

Symbol	Parameter	Conditions
C_{L1}	Output load on CPU Data bus	47 pF
C_{L2}	Output load on CRAM Interface	5pF min., 40 pF max.
C_{L3}	Output load on $\overline{IRQ}$	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	5V ± 5%
V_{SS}	Ground potential	0V
T_A	Ambient operating temperature	0°C to +70°C

Figure 19. AC specification definition

8 Timing Specifications

Number	Description	Min	Max	Units
1	Reset width	t_{BCLK} + 7		ns
2	First 9711 access after Reset	12 t_{BCLK} + 30 t_{PCLK}		ns

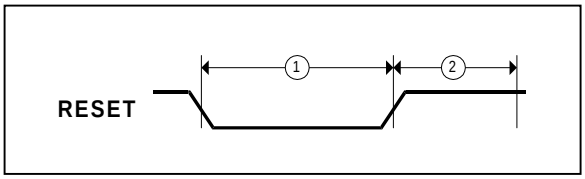


Figure 20. Reset timing

Number	Description	Min	Max	Units
	Oscillator frequency		66.67	MHz
1	Clock period	15		ns
2	Clock width high	6		ns
3	Clock width low	6		ns
4	Clock rise time from V_{IL} to V_{IH}		5	ns
5	Clock fall time from V_{IH} to V_{IL}		5	ns

Figure 21. External PCLK clock

Number	Description	Min	Max	Units
	Oscillator frequency		33.3	MHz
1	Clock period	30		ns
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 22. External BCLK clock

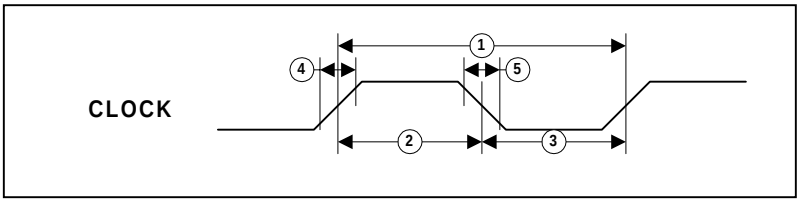


Figure 23. External clock

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

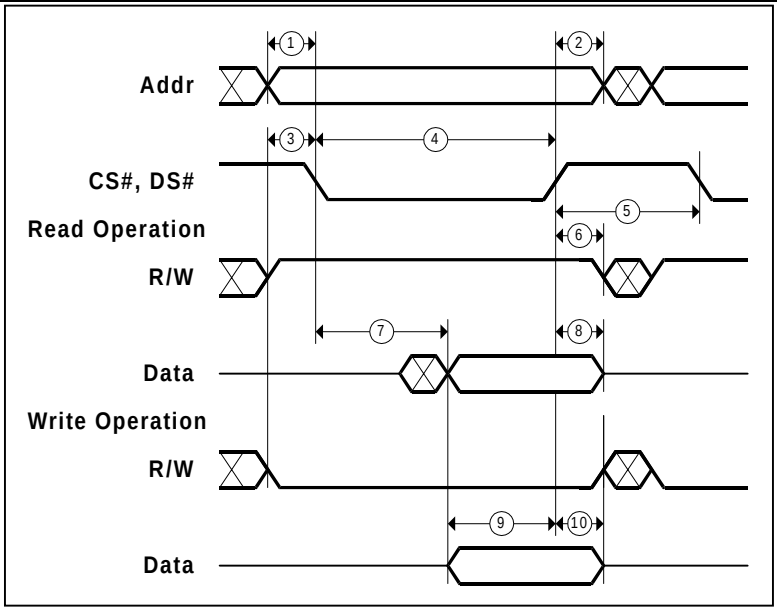


Figure 24. Asynchronous CPU Timing

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

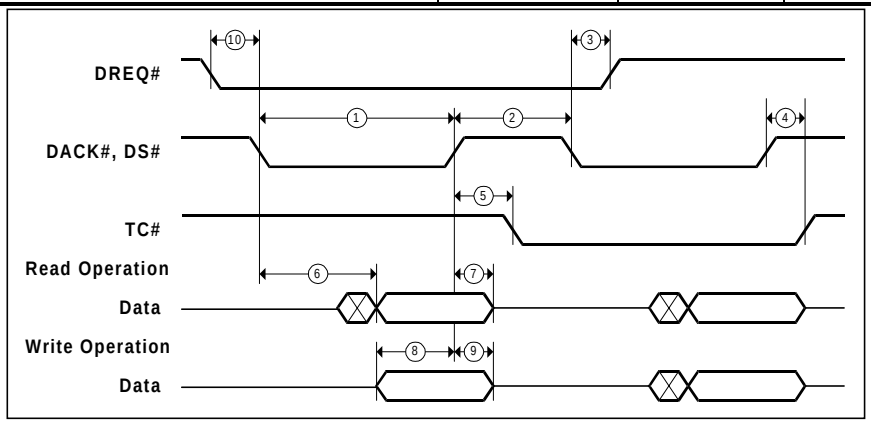


Figure 25. Asynchronous DMA Timing

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

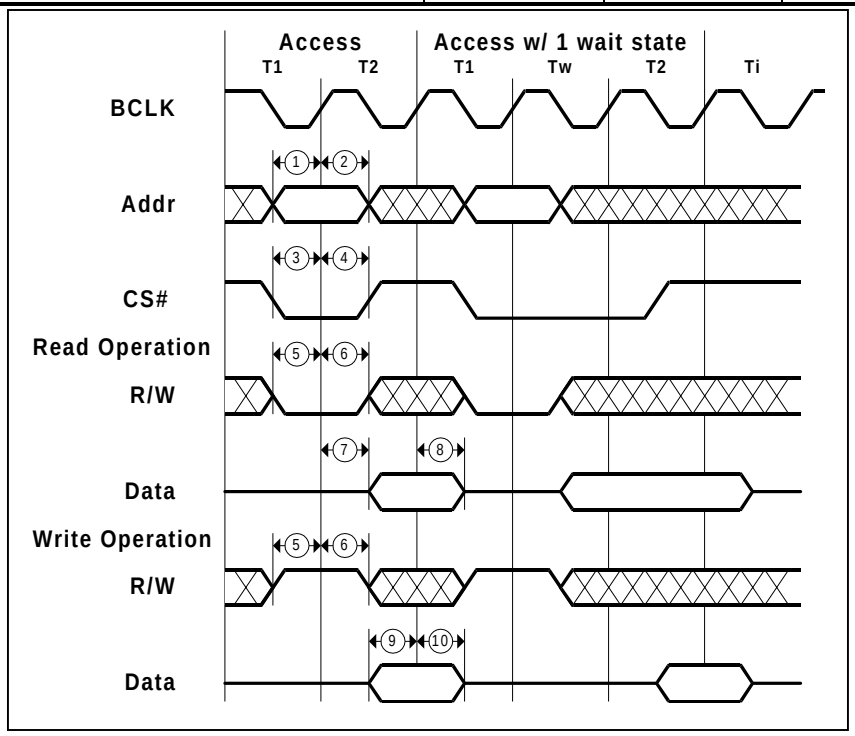


Figure 26. Synchronous CPU Timing

Number	Description	Min	Max	Units
1	DREQ# output valid delay		12	ns
2	DREQ# hold	5		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		12	ns
7b	TC# output valid delay - tri-state mode		14	ns
8	TC# hold	5		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

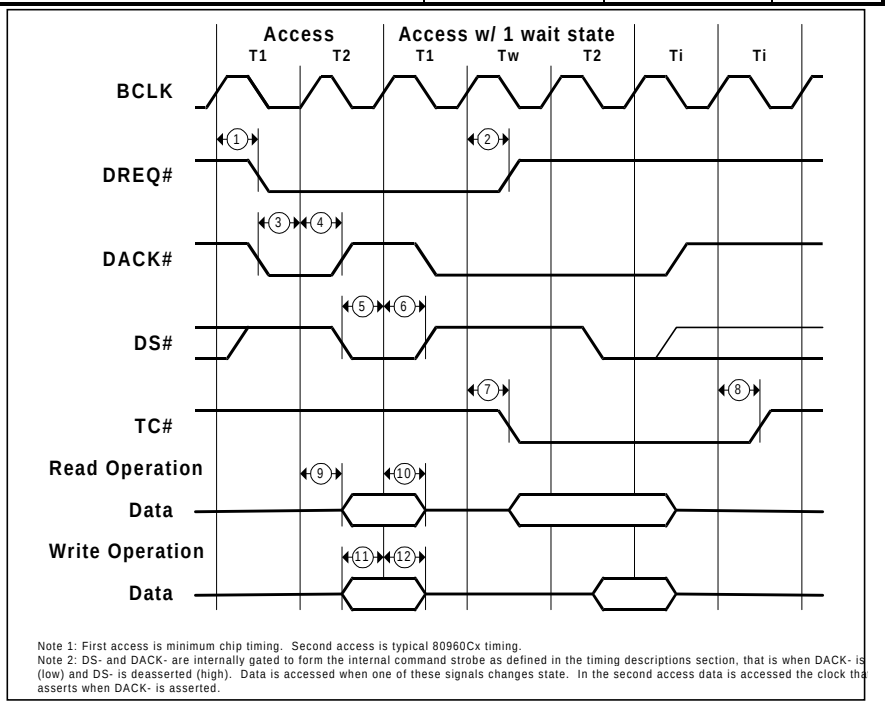


Figure 27. Synchronous DMA Timing

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

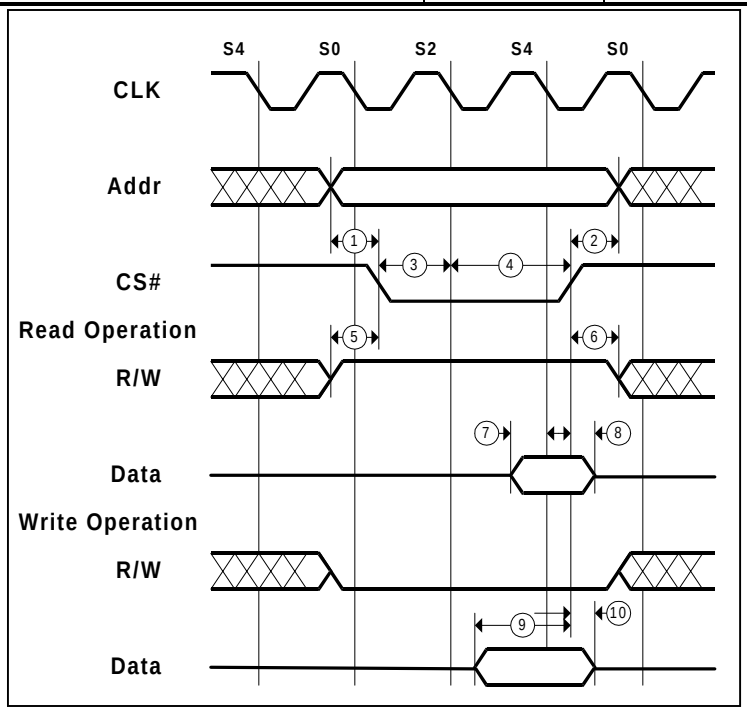


Figure 28. 68360 CPU Timing

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 valid 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		12	ns
8	DREQ# hold after falling edge of BCLK	4		ns
9a	TC# active delay after falling edge of BCLK (totem-pole mode)		12	ns
9b	TC# active delay after falling edge of BCLK (tri-state mode)		13	ns
10	TC# hold after falling edge of BCLK	4		ns

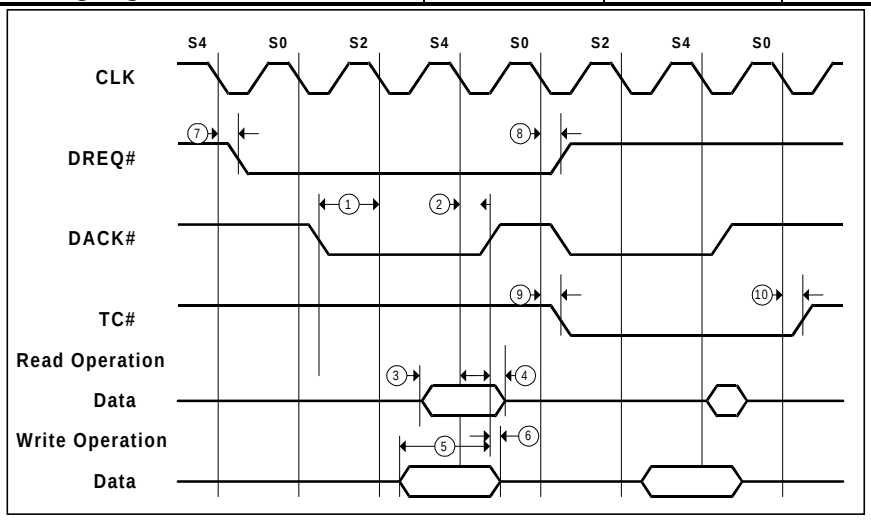


Figure 29. 68360 DMA Timing

Number	Description	Min	Max	Units
1	Data valid after Addr/LB/UB valid (access time)		2 t_{PCLK} - 18	ns
2	Data valid after OE active		1.5 * t_{PCLK}- 15	ns
3	Address valid width (cycle time)	2 t_{PCLK} - 6		ns
4	Data hold after OE inactive	0	t_{PCLK} - 10	ns

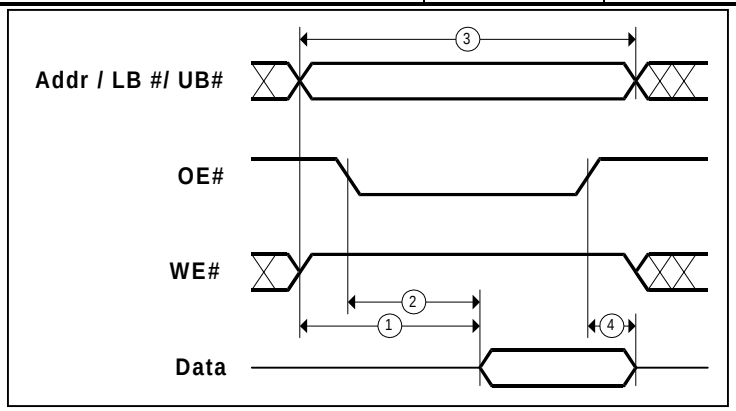


Figure 30. Compression SRAM Read Timing

Number	Description	Min	Max	Units
1	Addr/LB/UB setup to WE# valid	t_{PCLK} - 7		ns
2	WE# active width	t_{PCLK} - 2		ns
3	Addr hold after WE# inactive	2		ns
4	Data setup before WE# inactive	t_{PCLK} - 6		ns
5	Data hold after WE# inactive	1		ns
6	Addr valid to WE# inactive	2 t_{PCLK} - 7		

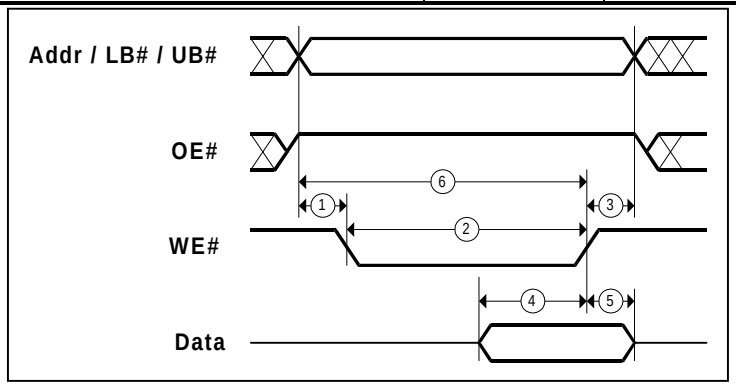


Figure 31. Compression SRAM Write Timing

Number	Description	Min	Max	Units
1	RAS# read / write cycle time	8 t_{PCLK}		ns
2	RAS# pulse width	5 t_{PCLK}		ns
3	RAS# precharge time	3 t_{PCLK}		ns
4	CAS# read / write cycle time	2 t_{PCLK}		ns
5	RAS# hold time from CAS precharge	3 t_{PCLK}		ns
6	CAS# hold time	3 t_{PCLK} - 1		ns
7	RAS# to CAS# delay	2 t_{PCLK}	3 t_{PCLK} + 1	ns
8	CAS# pulse width	t_{PCLK} - 2		ns
9	CAS# hpage precharge time	t_{PCLK}		ns
10	CAS# to RAS# precharge	3 t_{PCLK}		ns
11	CAS# precharge time	5 t_{PCLK}		ns
12	RAS# to col. Address delay time	t_{PCLK}	t_{PCLK} + 2	ns
13	Row address setup time	t_{PCLK} - 2		ns
14	Row address hold time	t_{PCLK}		ns
15	Column address setup time	t_{PCLK} - 4		ns
16	Column address hold time	t_{PCLK} + 1		ns
17	Column address to RAS	3 t_{PCLK} - 3		ns
18	Read command setup (read)	4 t_{PCLK}		ns
19	Read command hold from RAS# (read)	2 t_{PCLK}		ns
20	Read command hold from CAS# (read)	2 t_{PCLK}		ns
21	Access time from CAS precharge (read)		2 t_{PCLK} + 5	ns
22	Read data access time from OE#		2 t_{PCLK}	ns
23	Read data access time from RAS#		4 t_{PCLK}	ns
24	Read data access time from col. Address		2 t_{PCLK}	ns
25	Read data access time from CAS#		t_{PCLK}	ns
26	Read data output hold time	0		ns
27	Output buffer turn-off from OE# (read)		t_{PCLK}	ns
28	WE# setup to CAS# (write)	t_{PCLK} - 1		ns
29	WE# hold from CAS# (write)	t_{PCLK} - 1		ns
30	Write data setup to CAS#	t_{PCLK} - 5		ns
31	Write data hold from CAS#	t_{PCLK}		ns
32	Output buffer turn-off from WE# active (not drawn)		t_{PCLK}	ns
33	WE# Pulse Width	2 t_{PCLK}		ns
34	RAS# hold after CAS#	t_{PCLK} - 2		ns

Figure 32. Compression DRAM Read and Write Timing

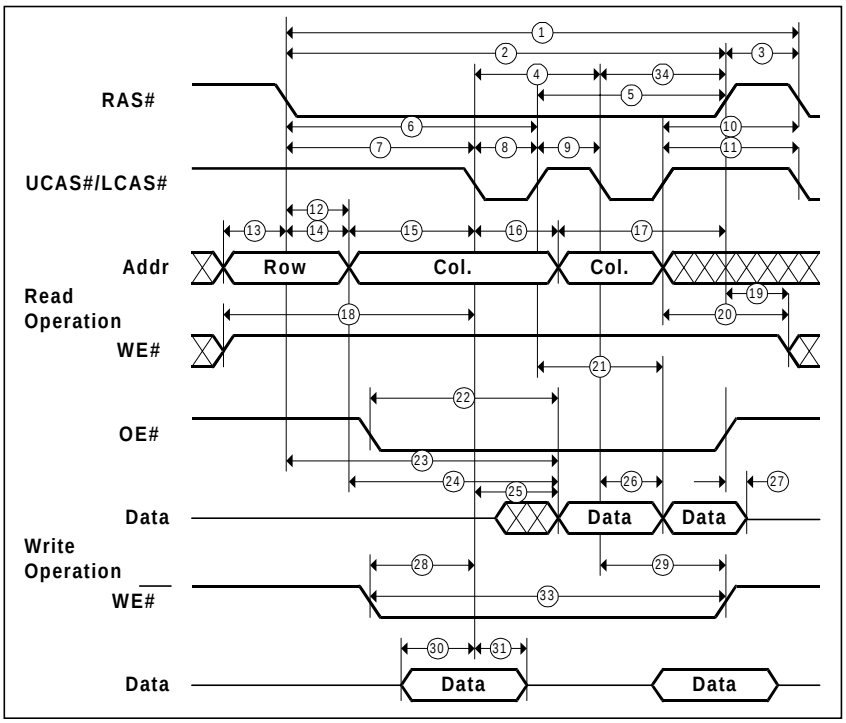


Figure 33. Compression DRAM Read and Write Timing Diagram

Number	Description	Min	Max	Units
1	CAS setup to RAS	t_{BCLK} - 4		ns
2	CAS hold from RAS	3 t_{BCLK}		ns

Note: All other relevant timing is referenced above.

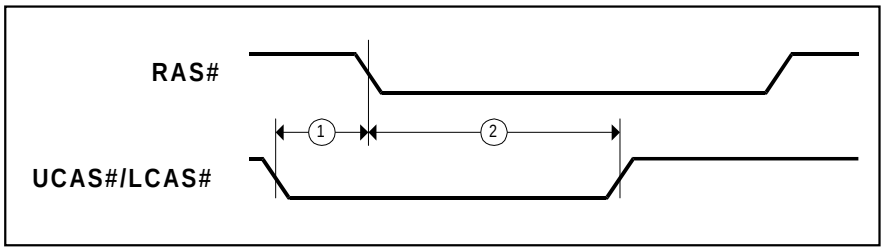


Figure 34. Compression DRAM Refresh Timing

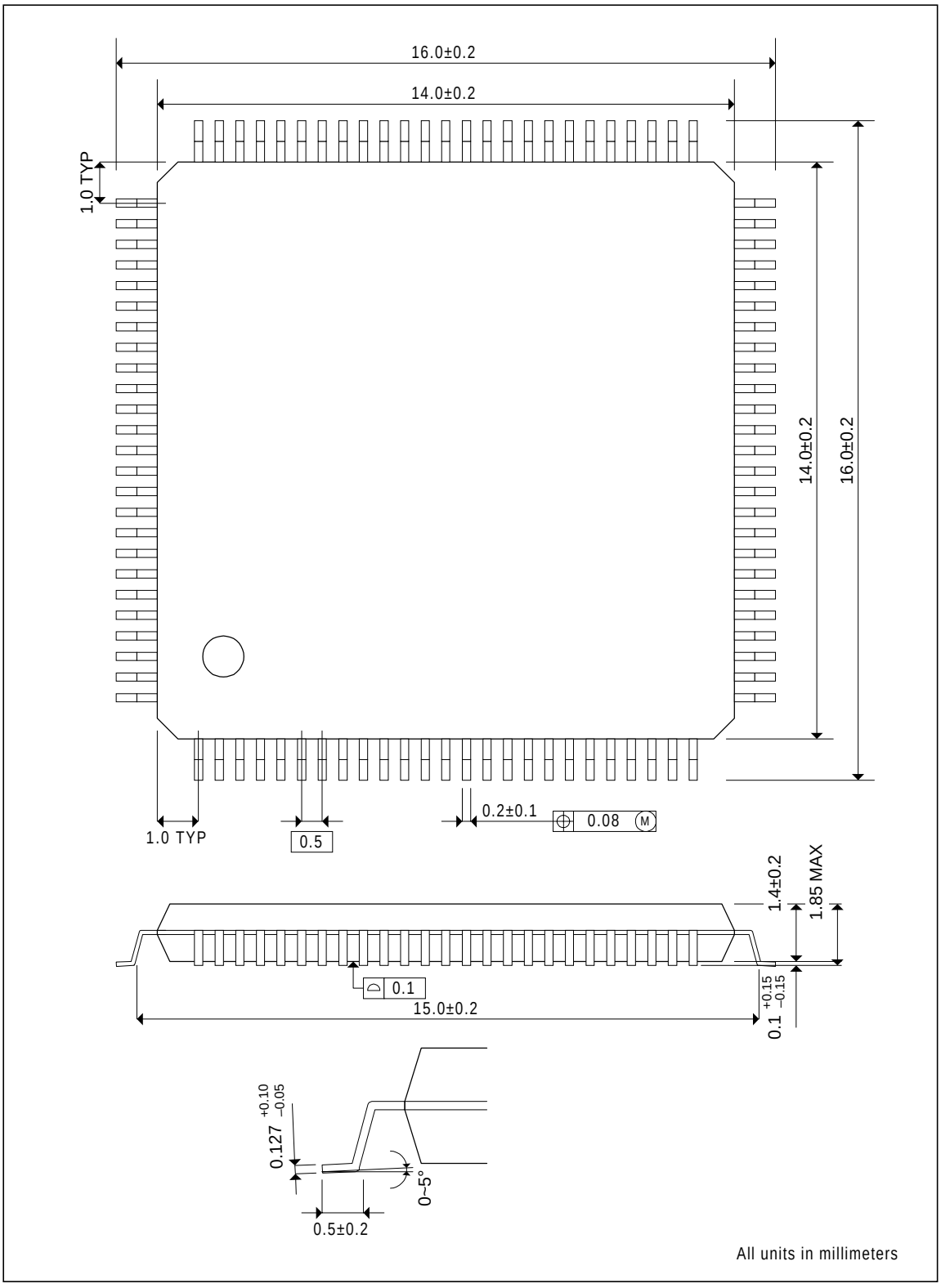


Figure 35. 100-pin TQFP package

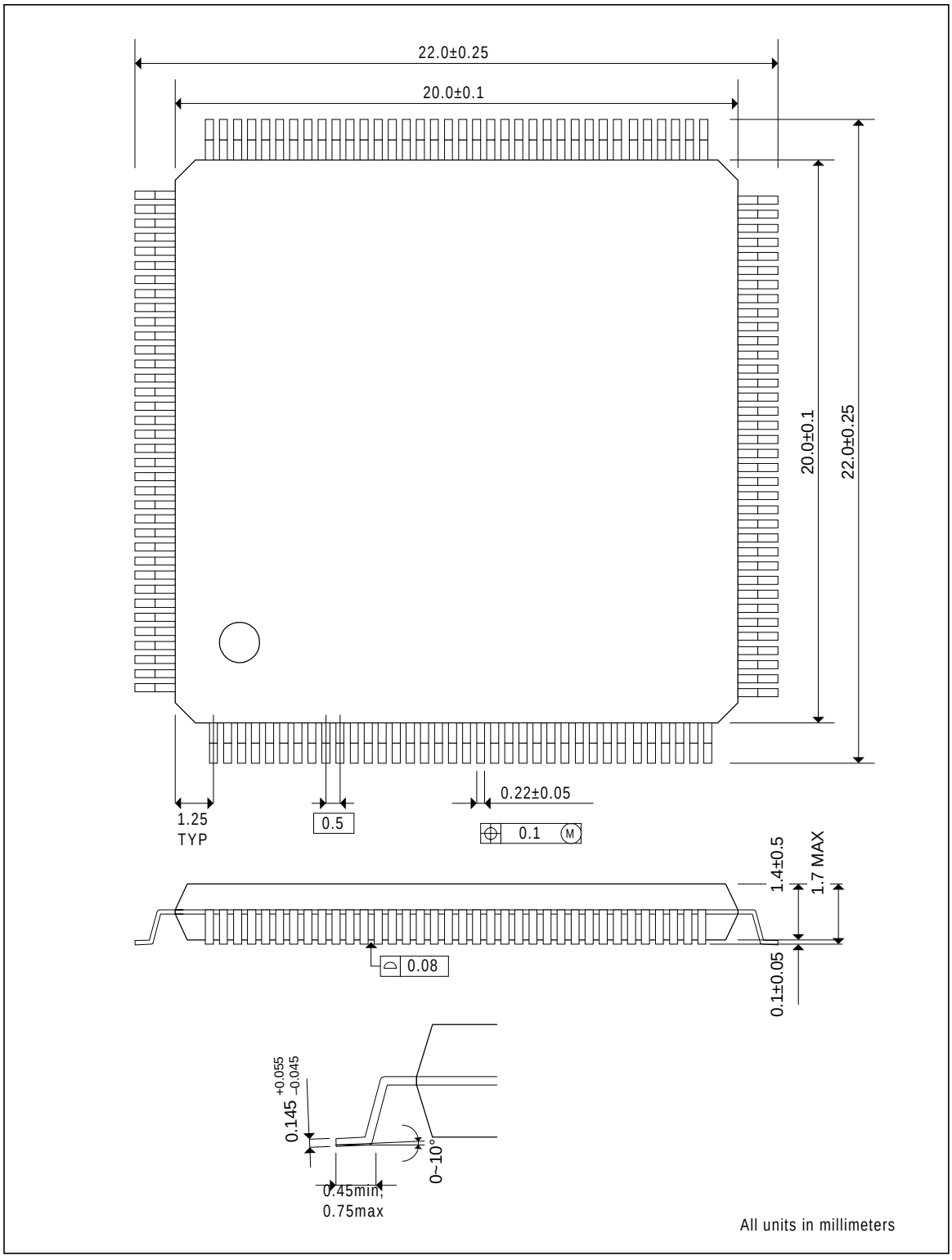


Figure 36. 144-pin TQFP package

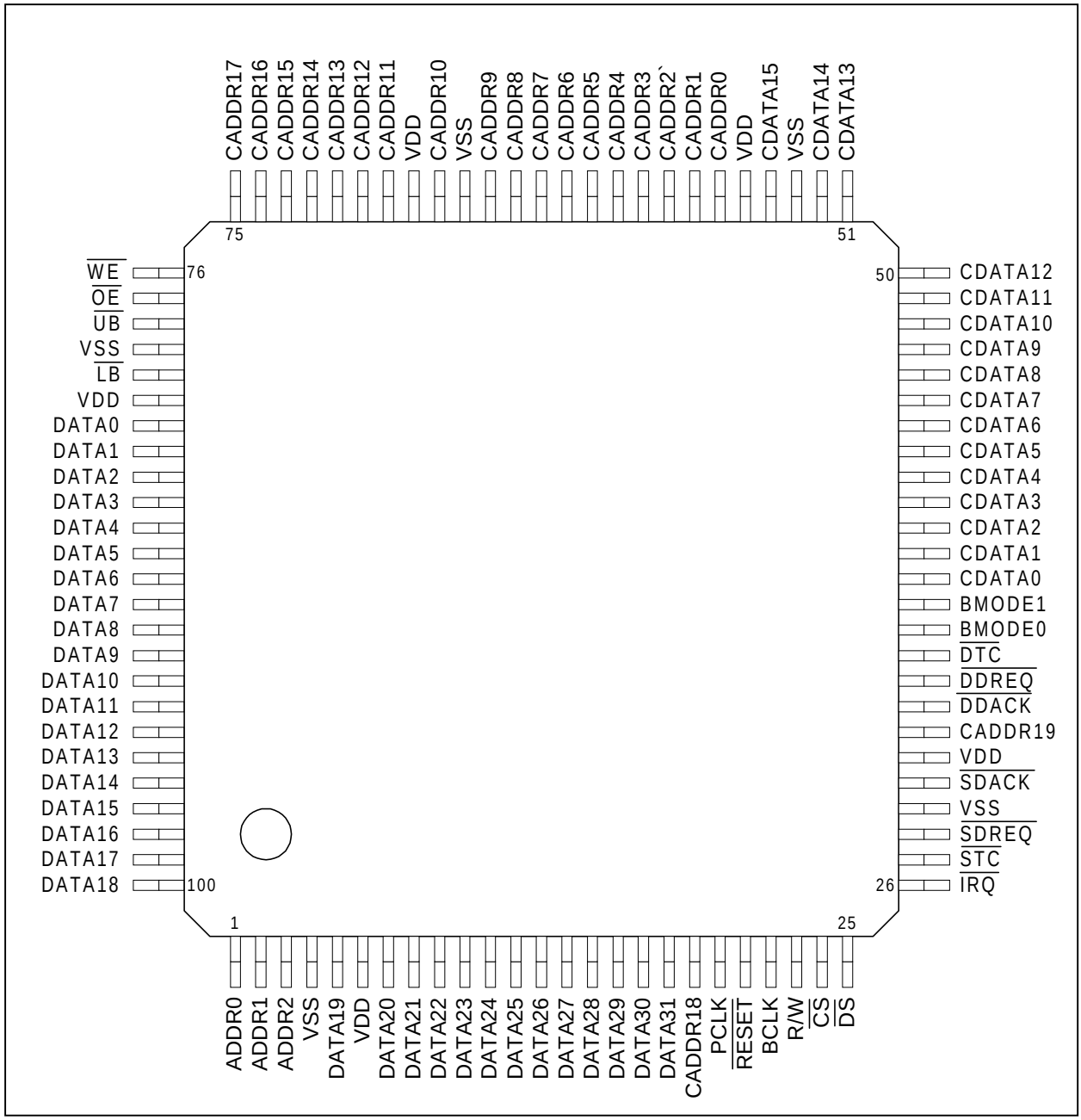


Figure 37. 100-pin TQFP pinout

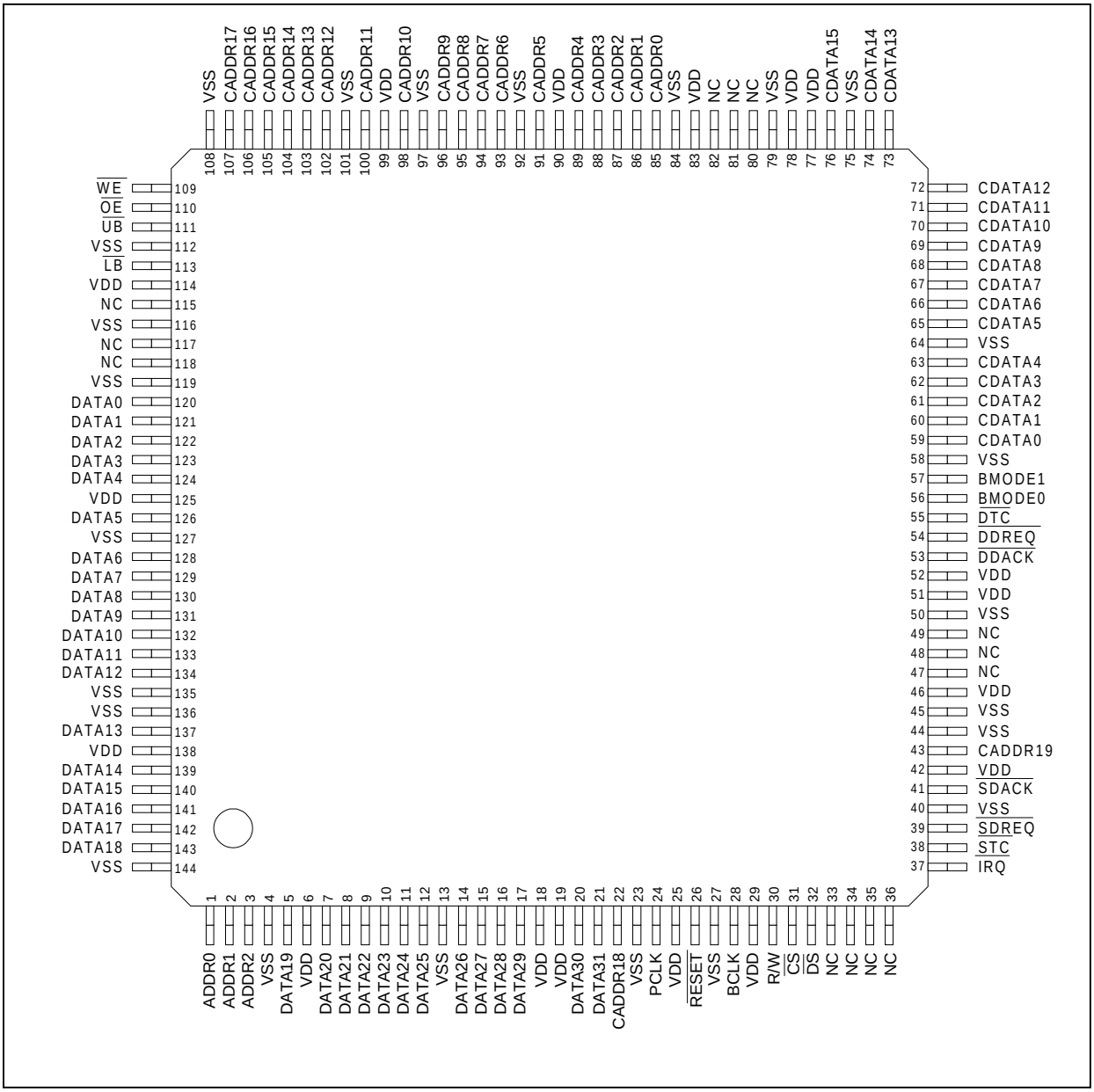


Figure 38. 144-pin TQFP pinout