VMS115

Data Sheet

Revision: 2.0

Table of Contents

1 Introduction

This document describes the VMS115, a high performance cryptographic coprocessor performing data encryption, integrity verification and authentication functions. The chip contains IPSec processing functionality, including DES/3DES encryption and MD5/SHA-1 hashing. An exponentiator is also available on this chip.

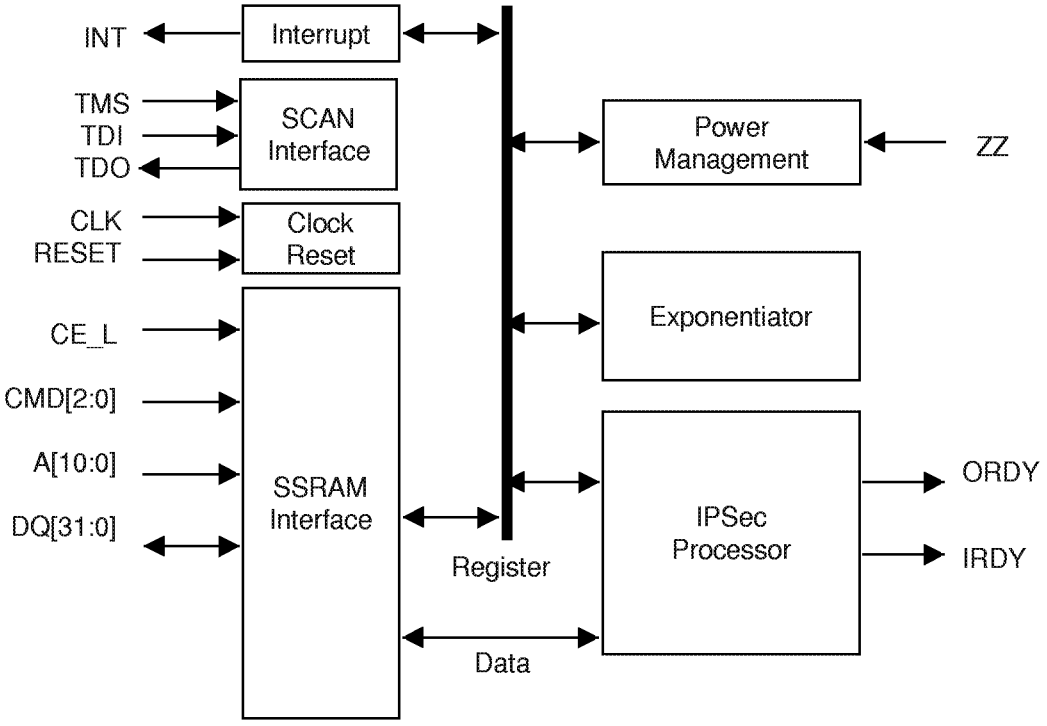
This security coprocessor is intended for applications in which physical security is not of major concern in the deployed system, since keys are loaded into the device in the clear.

This document will describe, in detail, each security block and how each block is used via an external processor. A memory map is provided as well as a description of the data buffering and movement through the chip. Performance figures for each of the functional blocks is included along with AC/DC timings, timing diagrams, power dissipation figures and device pinout.

2 Overview

The VMS115 is composed of two major functional blocks; IPSEC block and exponentiator. The architecture of the chip is illustrated below, showing the relationship between the various hardware components. VLSI's IPSEC block is a streamlined design customized to meet IPSEC protocol requirements. The block is composed of a high-speed hash and triple-DES engine. The hash engine in the IPSEC block supports the MD5 and SHA-1 algorithms as well as the HMAC functionality required by IPSEC.

VMS115 Block Diagram



2.1 Features

The VMS115 performs two major functions; IPSec processing and exponentiation.

- Supports Electronic Codebook (ECB) and Cipher Block Chaining (CBC) ciphering operation.
- Supports two operating modes: Single and triple DES (two key and three key types)
- Based on a FIPS-PUB 46-2 compliant design validated by the National Institute of Standards and Technology (NIST)
- HMAC processing supported
- MD5 or SHA-1 hashing of clear and encrypted data.
- 1024-bit modular exponentiation

3 Memory Map

This section describes the VMS115 memory map. This map refers to the internal memory address locations of the specified registers.

VMS115 Memory/Register Map

Function	Address Map		Function	Address Map
General Purpose Registers			**Exponentiator**	
Interrupt Status Register	0x400		K Register	0x200
Interrupt Mask Register	0x401		Reserved	0x201
Configuration Register	0x402		Reserved	0x202
Context Read Pointer Register	0x404		Reserved	0x203
Context Write Pointer Register	0x405		Control & Status Register	0x204
			Number of Exponentiation Bits	0x205
			Reserved	0x206
			Operand RAM	0x080-0x0FF

3.1 Address Decoding

The following definitions indicate how the address bits are allocated. The upper address bits (bits 7 - 10) are used to pre-decode the register address space to relax timing.

A[6:0] Decode inside VMS115
A[7] Expo operand RAM select
A[8] Expo result RAM select
A[9] Expo register space
A[10] Other registers

4 Data Structure

This section describes the packets that will be processed by the VMS115. It is also evident from this section that it is not necessary for the VMS115 to "know" much about the packets being processed other than where to begin, end and what function to perform.

Prior to performing DES or Hash operations, the VMS115 will receive context data followed by the packet data. The context data that is provided to the VMS115 prior to packet processing dictates what will happen with the incoming packets. The VMS115 can perform any of the following three functions on the data: nothing; Hash some or all of the data; DES some or all of the data. Regardless of what function is performed, when complete the entire data fragment will be sent back out.

4.1 Context Data

The Context data consists of the elements described in the following table and is applicable for IPSec processing only. The order is significant, as the more dynamic elements are listed first and those elements expected to change less frequently are listed later. A context write will flush all previous states of the IPSec processor. A context write does not overwrite the entire context area, thus it's only necessary to modify those elements in the context that have actually changed since the previous write.

The Context Read/Write Pointer Registers are used by the VMS115 to determine the location within the context block for read and write accesses. The context pointers point to the word (4 bytes) of data that will be read or written for a context access. The Context Read/Write Pointer registers are reset to the first location (zero) under the following conditions:

Read Pointer Reset Conditions:

1. Any packet data access (read or write).
2. A change from context write to context read.
3. Soft Reset
4. Chip Reset

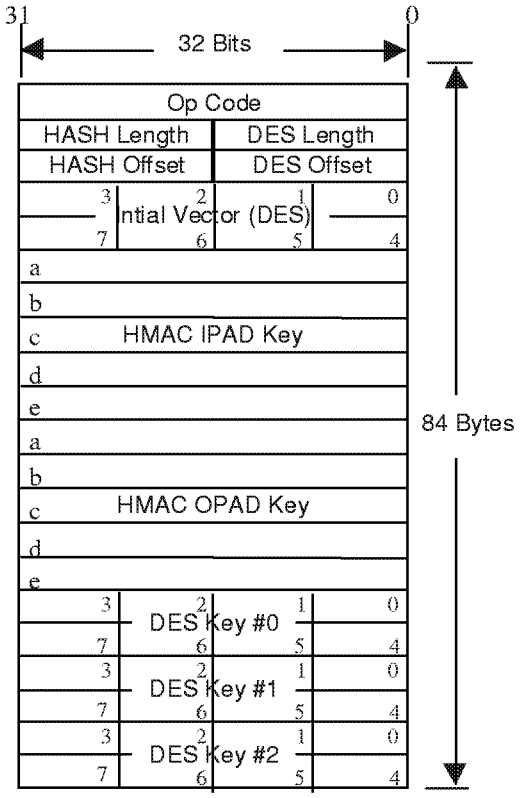
Write Pointer Reset Conditions:

1. Any packet data access (read or write).
2. A change from context read to context write.
3. Soft Reset
4. Chip Reset

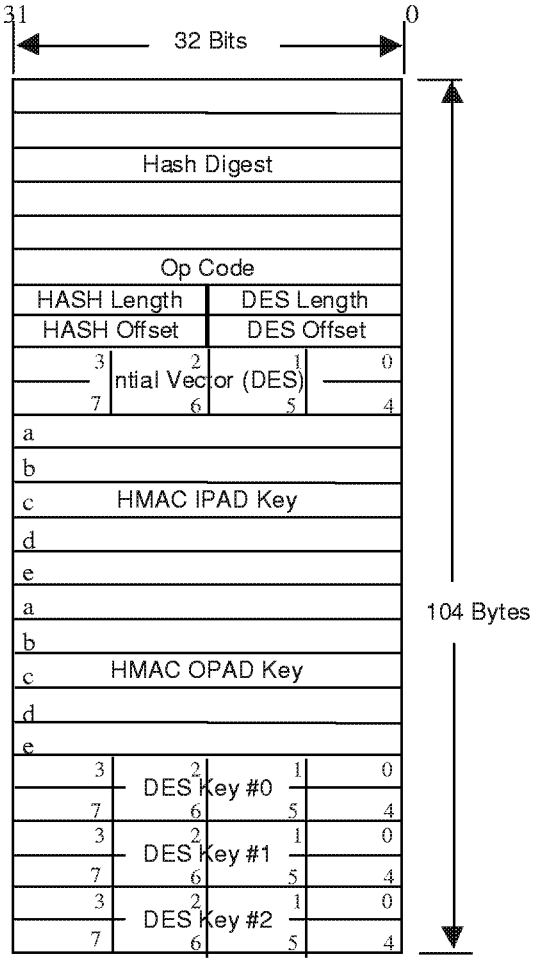
The context pointers may be directly accessed through the VMS115 memory space by the external processor and they may be set to any value via this interface. A context data read results in 20 bytes of Hash Digest being added to the beginning of the context data. The Hash Digest is stored in the Context Buffer such that the beginning of the digest (first word) is at the beginning of the Context.

Separate pointer registers exist for read (Context Read Pointer Register) and writes (Context Write Pointer Register). Each can be accessed and set separately by the external processor.

Context Data Write



Context Data Read



4.2 Packet Data

Incoming packets must begin on 4 byte boundaries and all packet data must be contiguous. Packet data must always be transferred to the VMS115 4 bytes at a time. The end of the packet will be determined by the Total Length field in the Opcode Register. Bytes in the last 4 byte segment beyond the Total Length value will be ignored by the VMS115.

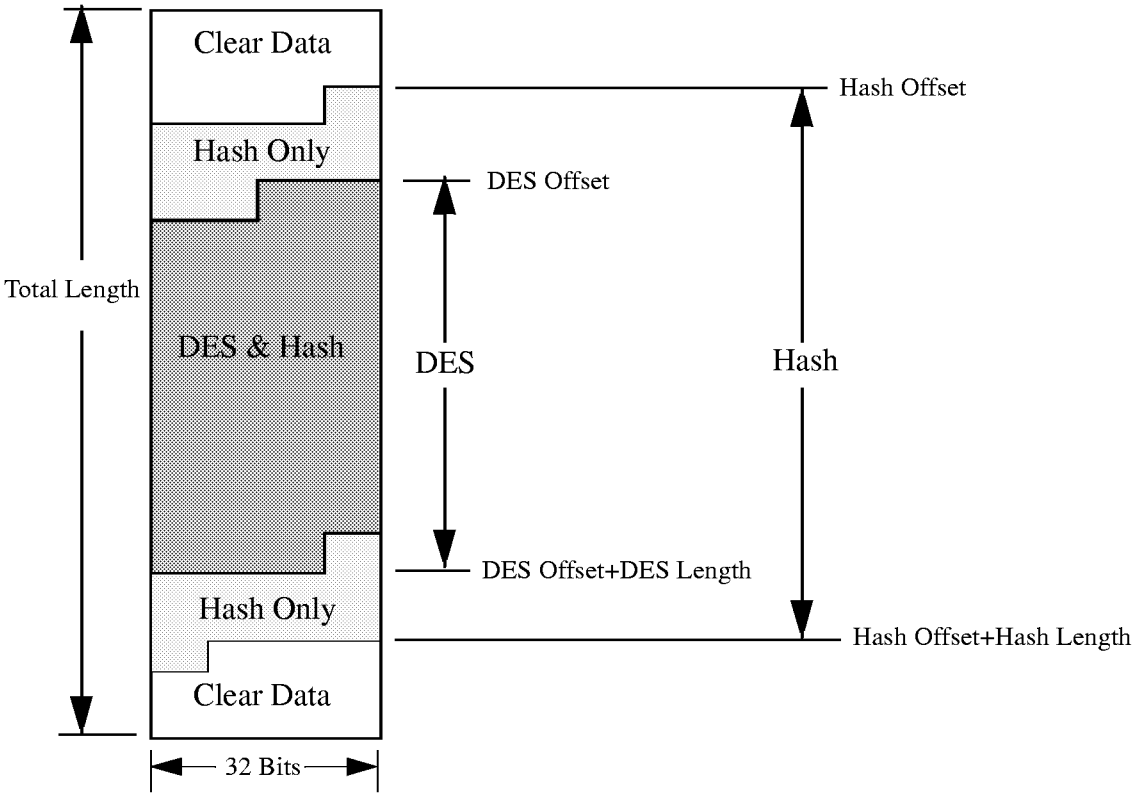
When DES is disabled in the Opcode Register, DES Length and DES Offset are ignored. Similarly, when the Hash function is disabled in the Opcode Register, Hash Length and Hash Offset are ignored.

The Hash Offset value must be less than or equal to the DES Offset value when both the Hash and DES functions are enabled in the Opcode Register.

The DES Length must be a multiple of 8 bytes, thus bits 0,1 and 2 of DES Length must be set to 0. If DES Length equals 0, then no packet data will be processed by the DES engine. Likewise, if Hash Length equals 0, then no packet data will be processed by the Hash engine. If not disabling

the Hash Engine via the Hash Length value, then the Hash Length must be at least 4 bytes.

Packet Data Partitioning



5 Functional Description

This section will describe the functionality of the chip, interfaces to the functions and a register summary for the interfaces.

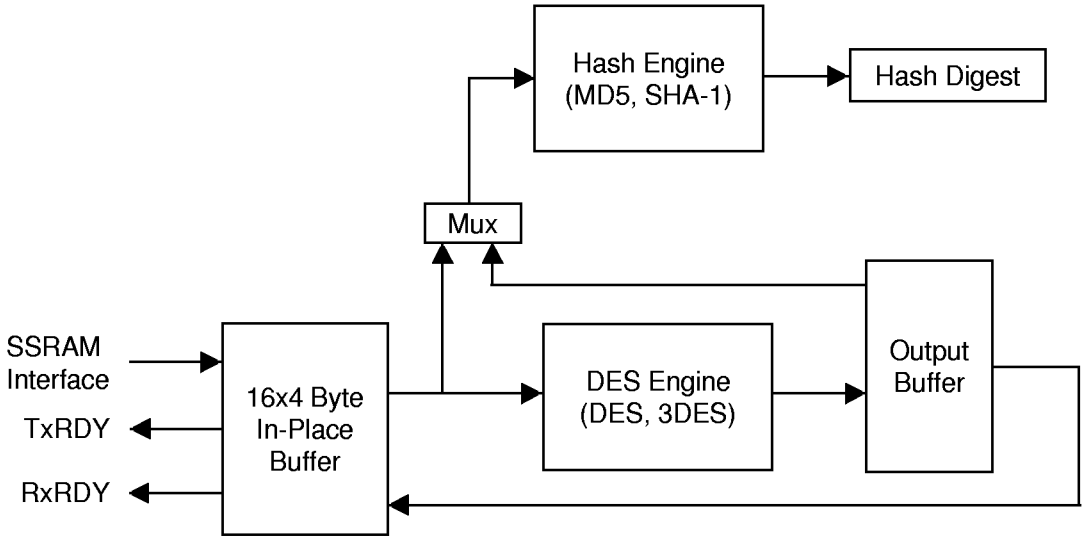
5.1 IPSec Processor

This functionality includes DES, 3DES, SHA-1 hashing and MD5 hashing. Parallel processing of data through the encryption and hash engines is supported. IPSec HMAC processing is also supported with this block.

5.1.1 Main Features

- Supports Electronic Codebook (ECB) and Cipher Block Chaining (CBC) ciphering operation.
- Supports two operating modes: Single and triple DES (two key and three key types).
- Based on a design validated by the National Institute of Standards and Technology (NIST)
- HMAC processing supported.
- MD5 or SHA-1 hashing of clear and encrypted data.

IPSec Block Diagram



5.1.2 DES Engine

This DES core is a subset of VLSI's standard VK110 DES core. This block is a high speed ciphering engine that supports the Data Encryption Standard (DES) algorithm as specified by the National Institute of Standards and Technology's (NIST) Federal Information Processing Standards Publication #46-2 (FIPS PUB 46-2). It offers two modes of ciphering, Electronic Code Book (ECB) and Cipher Block Chaining (CBC), and two operating modes, single and triple DES. The triple DES function supports both two-key and three-key. The aforementioned ECB mode will be implemented in accordance with FIPS-PUB 81.

5.1.2.1 Block Architecture

The design is an optimized DES engine that can be utilized for both single and triple DES operations. The single DES operates per the FIPS-PUB standard. The triple DES function operates as either a two-key or three-key operation. The triple DES encryption cycle includes three ciphering sessions which start with the encryption of the plain text using Key 0. The result is, then, decrypted with Key 1 before encrypted again with Key 2. For two-key triple DES operations, Key 0 will be the same as the Key 2. The triple DES decryption cycle is the opposite of the encryption cycle.

Cipher_Text = Encrypt_Key2[Decrypt_Key1[Encrypt_Key0[Plain_Text]]]

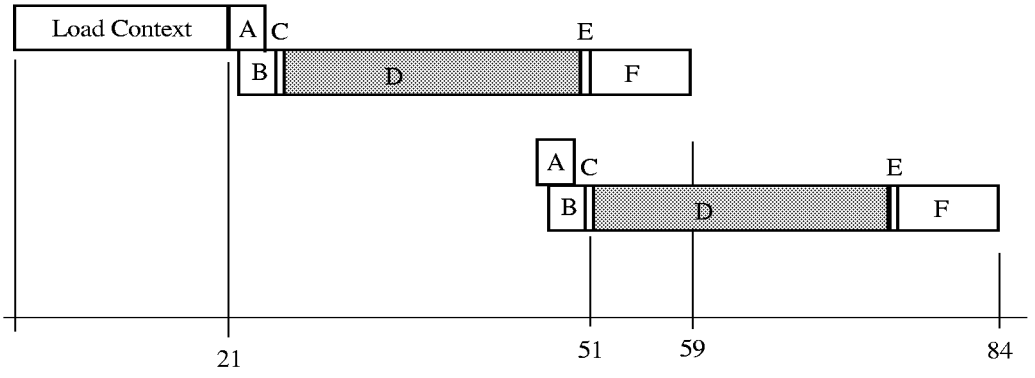
Plain_Text = Decrypt_Key0[Encrypt_Key1[Decrypt_Key2[Cipher_Text]]]

The block uses a 64 bit Initialization Vector (IV) register for the CBC mode.

5.1.2.2 Triple DES Throughput

The DES Core requires 24 clock cycles for Triple DES processing through the DES engine. The internal state machine takes one clock cycle to latch a 64 bit block of data into the engine and another cycle to latch a 64 bit of data out of the engine.

Triple DES Pipeline Diagram



Triple DES Processing Cycles

Function	Description	Clock Cycles
	Load Context	21
A	Load In-Place Buffer.	3
B	Load Word Align Buffer.	3
C	Load DES Core.	1
D	Triple DES Processing.	24
E	Unload DES Core.	1
F	Load Data Into In-Place Buffer.	8

5.1.2.2.1 Triple DES Throughput Calculations

These throughput figures assume the 4-Byte In Place Buffer is not starved for data.

P = packet size in bytes
n = number of 64-bit blocks = Round up to nearest integer [(P * 8)/64]
f = frequency = 80 MHz
Cycles Required = Load Context + ((D+E) * n) + 4 + C + F
= 21 + ((D+E) * n) + 13
Throughput = (f * P * 8)/Cycles Required

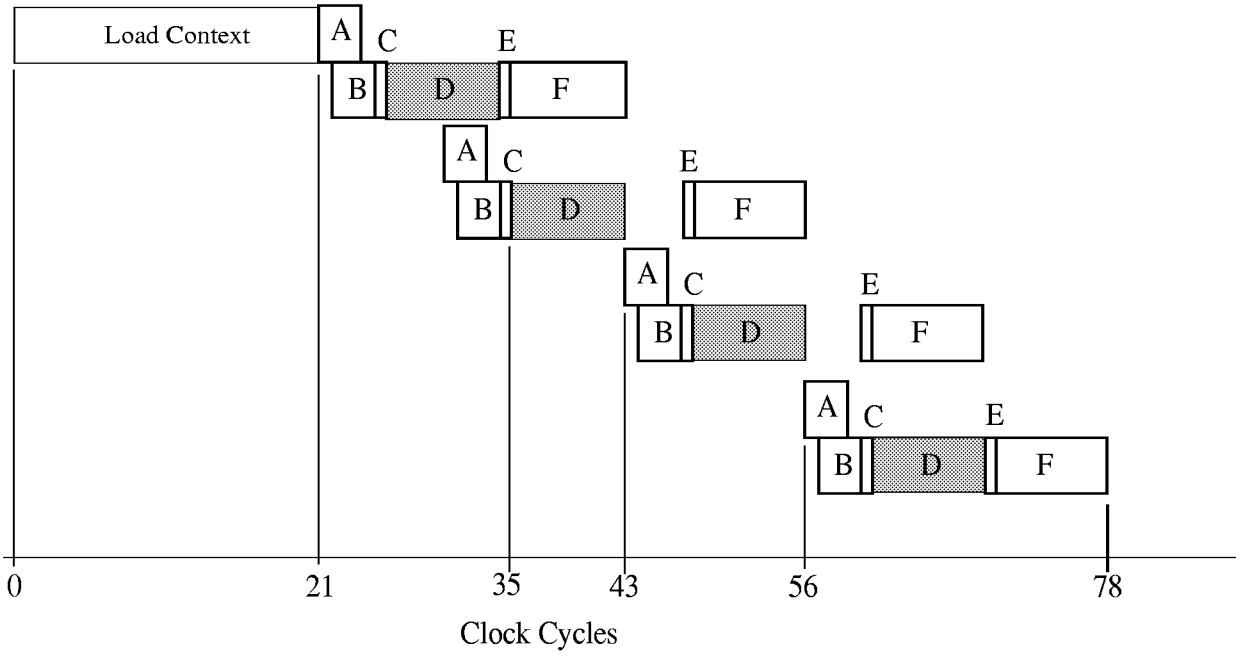
Triple DES Performance Table

Packet Size (Bytes)	3DES (Mbits/sec)
64	175
256	196
1500	203
Packet Size => Infinity	205

5.1.2.3 Single DES Throughput

The DES Core requires 8 clock cycles for a single pass through the DES engine. The internal state machine takes one clock cycle to latch a 64 bit block of data into the engine and another cycle to latch a 64 bit block of data out of the engine.

Single DES Pipeline Diagram



Single DES Processing

Function	Description	Clock Cycles
	Load Context	21
A	Load In-Place Buffer.	3
B	Load Word Align Buffer.	3
C	Load DES Core.	1
D	DES Processing.	8
E	Unload DES Core.	1
F	Load Data Into SSRAM Buffer.	8

5.1.2.3.1 Single DES Throughput Calculations

These throughput figures assume the 4-Byte In Place Buffer is not starved for data.

P = packet size in bytes
n = number of 64-bit blocks = Round up to nearest integer [(P * 8)/64]
f = frequency = 80 MHz

For n = 1 or 2

Cycles Required = Load Context + ((D+E) * n) + 4 + C + F
= 21 + ((D+E) * n) + 13

For n >= 3

Cycles Required = Load Context + [(n - 2)*(4 + (D+E))] + (2 * (D+E)) + C + 4 + F
= 21 + [(n - 2)*(4 + (D+E))] + (2 * (D+E)) + 13

Throughput = (f * P * 8)/Cycles Required

Single DES Performance Table

Packet Size (Bytes)	DES (Mbits/sec)
64	315
256	370
1500	389
Packet Size => Infinity	393

5.1.3 Hash Engine

The hash engine implements the FIPS 180-1 compliant Secure Hash Algorithm (SHA-1) and the MD5 hash algorithm. These algorithms are used for computing condensed representations of a message or data file, called a message digest. SHA-1 generates a 160-bit message digest while MD5 generates a 128-bit message digest. The algorithm is designed to have the following properties: it is computationally infeasible to find a message which corresponds to a given message

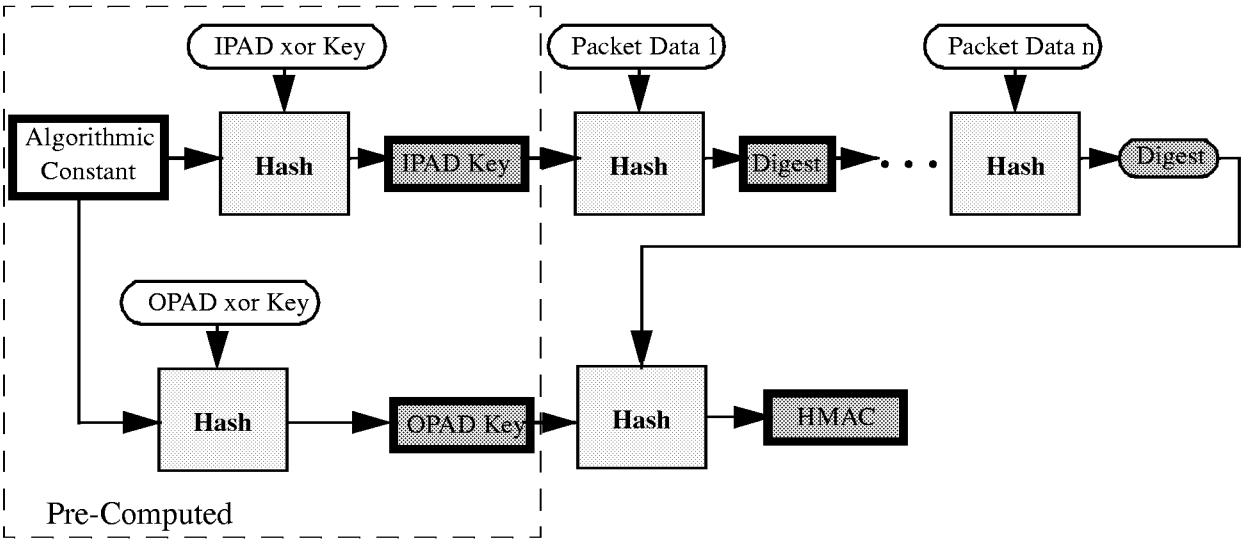
digest, or to find two different messages which produce the same message digest.

The following diagram illustrates the typical flow required to compute a Hash Digest. Both the MD5 and SHA-1 hashing algorithms generate a hash digest from two inputs; a state value and a data value. The data value is 512-bits and the state value varies depending upon the hash algorithm. The computed digest is 128 bits for MD5 and 160 bits for SHA-1.

Hash Flow Diagram

Generating an HMAC (Hash Message Authentication Code) digest requires several hash operations. If n is the number of 512-bit blocks which need to be hashed, then the total number of hash operations is n + 3. Two of the extra three operations can be precomputed prior to data being sent to the VMS115. The precomputed hash operations result in the IPAD Key and OPAD Key values. The third extra hash operation is required to actually generate the HMAC Digest, which is the result of hashing the OPAD Key and the digest result from the last packet data hash operation.

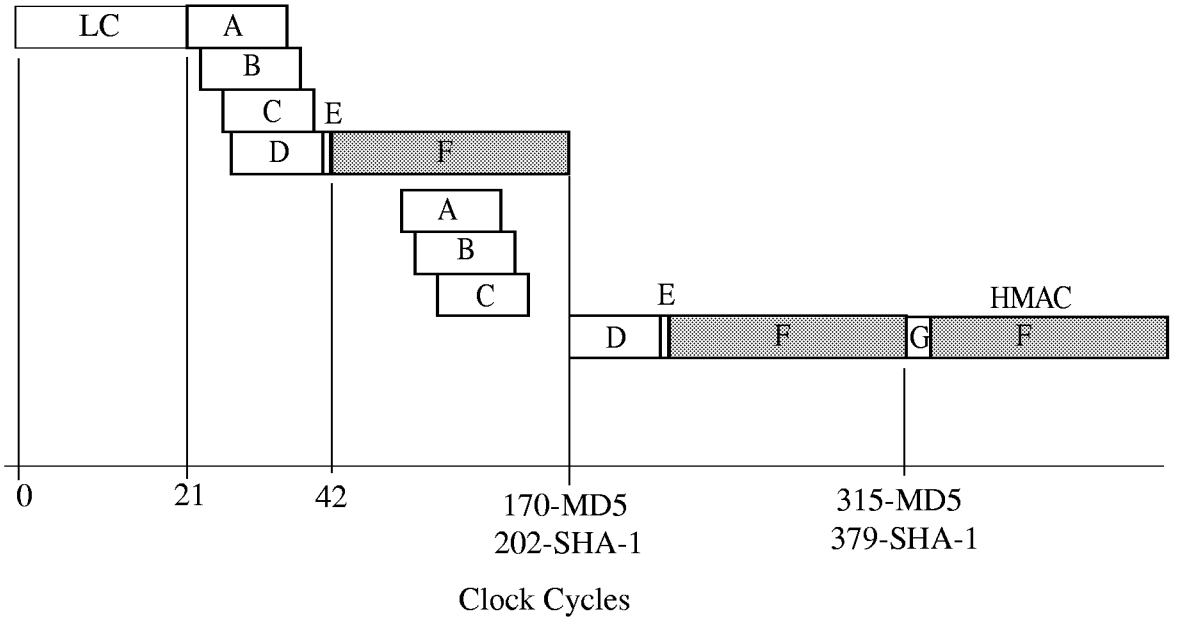
HMAC Flow Diagram



5.1.3.1 Hash Throughput

The following diagram illustrates the hash pipeline when hashing prior to performing DES.

Pre-DES Hash and Hash Only Processing



Hash Processing

Function	Description	Clock Cycles
LC	Load Context	21
A	Load 16 x 4 bytes into In-Place Buffer	17
B	Word Align and Assemble	17
C	Load Hash FIFO	16
D	Load Hash Block	16
E	Startup Cycle	1
F (MD5)	MD5 Hash Processing	128
F (SHA-1)	SHA-1 Hash Processing	160
G	HMAC Feedback	6

5.1.3.2 HMAC Throughput Calculations

These calculations are for the HMAC performance. The raw numbers in the table are applicable to Hash only operations and DES Encrypt/Hash operations. The throughput for DES Decrypt/ Hash must account for the injected delay of decrypting data prior to sending it to the Hash engine. This delay will be 30 clock cycles.

P = packet size in bytes
n = number of 64-byte (512-bit) blocks = Round up to nearest integer [(P + 8)/64]
f = frequency = 80 MHz

Cycles Required = Load Context + 21 + ((F + D + E) * n) + G + F

Throughput = (f * P * 8)/Cycles Required

.

HMAC Only and Pre-DES Hash Performance Table

Packet Size (Bytes)	MD5 (Mbits/sec)	SHA-1 (Mbits/sec)
64	88	73
256	182	150
1500	262	215
Packet Size => Infinity	282	231

The following table defines the throughput figures for HMAC when decrypting and hashing data.

HMAC Post-DES Hash Performance Table

Packet Size (Bytes)	MD5 (Mbits/sec)	SHA-1 (Mbits/sec)
64	83	69
256	176	146
1500	260	214
Packet Size => Infinity	282	231

5.1.4 Register Summary

The table below describes the internal registers of the DES Core.

IPSEC Register Summary

Register	Access	Description	Reset Value
Opcode Register	R/W	IPSEC Opcode Register	00000000h

5.1.4.1 Opcode Register

Opcode Register Configuration

31	30	29	28	27	26	25	24
Total Length of Packet (Bytes)							

23	22	21	20	19	18	17	16
Total Length of Packet (Bytes)							

15	14	13	12	11	10	9	8
Reserved							Hash No Pad

7	6	5	4	3	2	1	0
Enable HMAC	Post-DES/ Pre-DES	Hash Enable	DES Enable	MD5/ SHA-1	CBC/ECB	Encrypt/ Decrypt	3DES/DES

5.1.4.1.1 Opcode Register Bit Definitions

Bits 31:16 Total length, in bytes, of packet - used to determine the end of packet. Minimum packet size is 0 and maximum packet size is $(2^{16}- 4)$.

Bits 15:9 Reserved

Bit 8 Hash No Pad

0 - Hash No Pad Off: Enables padding of the input data, hence the input data does not need to be a multiple of 64 bytes. The IPAD and OPAD keys are not used in this mode. Uses the Hash initial state constants.

1- Hash No Pad On: Disables padding of the input data, hence the input data must be a multiple of 64 bytes. The IPAD and OPAD keys from the Context Write data structure are used as the initialization state.

Bit 7 Enable HMAC - Enables the HMAC functionality to allow hashing with the HMAC computation. When HMAC is enabled (bit=1) the Hash No Pad must be disabled (Hash No Pad = 0).

0 - HMAC Disabled

1 - HMAC Enabled

Bit 6 Pre-DES/Post-DES - Indicates whether data is to be hashed before or after DES processing.

0 - Pre DES Hashing: Hashing occurs before the DES operation.

1 - Post DES Hashing: Hashing occurs after the DES operation.

Bit 5 Hash Enable - Enables hashing without the HMAC computation. This bit is ignored when the HMAC functionality is enabled (Bit 7 = 1).

0 - Hash Disabled

1 - Hash Enabled

Bit 4 DES Enable - Indicates that the data will be routed through the DES block.
0 - DES Disabled
1 - DES Enabled

Bit 3 SHA-1/MD5 - If bit 5 is set, this bit indicates the hash algorithm to be used.
0 - SHA-1
1 - MD5

Bit 2 CBC/ECB - If bit 4 is set, this bit indicates the DES mode in which to operate.
0 - Electronic Code Book
1 - Cipher Block Chaining

Bit 1 Encrypt/Decrypt - If bit 4 is set, this bit indicates whether the DES block should perform an encrypt or decrypt operation.
0 - Decrypt
1 - Encrypt

Bit 0 DES/3DES - If bit 4 is set, this bit indicates a single or triple DES operation.
0 - Single DES
1 - Triple DES

Hash Configuration Bits

Bit 8	Bit 7	Bit 5	Mode
x	1	x	HMAC Using Pre Compute
0	0	1	Hash Algorithm
1	0	1	Pre Compute IPAD and OPAD Keys for HMAC

Note: '101' may be used to chain hash data larger than 2^{16} bytes.

5.1.5 DMA Control Signals

5.1.5.1 VMS115 Rules for IRDY

The VMS115 will set the IRDY high whenever there are at least 8 word (32 bytes) locations available in the IPB. Otherwise IRDY will be low.

EXCEPT

The VMS115 will clear IRDY (low) when the last 8 or less words (32 bytes) of the packet have started transfer to the VMS115. The IRDY line is set low when the host writes the first word into the VMS115. The IRDY line will remain low until the context is READ from the VMS115.

5.1.5.2 Host Rules for IRDY

The host will sample IRDY prior to a DMA access to the VMS115. When IRDY is high, the host will always write 8 words (32 bytes) of information unless it is the end of the packet. The host will only write the remaining words (8 or less) at the end of a packet.

The host may break up the 8 words of data transfer into multiple DMA accesses to the VMS115. However, the DMA should only check IRDY after the entire 8 words (32 bytes) have been transferred.

5.1.5.3 VMS115 Rules for ORDY

The VMS115 will set ORDY high whenever there are at least 8 words (32 bytes) of data that have been processed and are available to be read from the IPB. Otherwise ORDY will be low.

EXCEPT

The VMS115 will set ORDY high when the last 8 words or less are processed and available to be read from the IPB. The VMS115 must clear ORDY (low) when the first word of the last transfer is read. ORDY will remain low until the context data is valid. ORDY will then be set high when the context data (digest) is valid. The context data will be valid immediately if there is no hash operation. ORDY will be cleared (low) when the first word of context information is READ from the VMS115.

NOTE: This implies that the DMA must read the context even if there is no digest.

5.1.5.4 Host Rules for ORDY

The host will sample ORDY prior to DMA access to the VMS115. When ORDY is high, the host will always read 8 words (32 bytes) of information unless it is the end of the packet. The host will read only the remaining words in a packet (8 or less) at the end of a packet.

The host may break up the 8 words of data transfer into multiple DMA accesses to the VMS115. However, the DMA should only check ORDY after the entire 8 words (32 bytes) have been transferred.

5.2 Exponentiator

This block performs a complete exponentiation of a 1024 bit number, a 1024-bit modulus and a 1024-bit base. The user must precompute 2 values that are dependent only upon the modulus.

The main 128-word (512 bytes) RAM is used to store the operands for the Exponentiate functions. The contents of this RAM are not changed by a hardware/software block level reset. Since this RAM appears in the full chip memory map, this area can be used as a general-purpose scratchpad RAM when the Exponentiator block is not in use.

The following equation is considered when exponentiating:

$Y = A^E \bmod N$

Where:

- A = base
- E = exponent
- N = modulus
- $K = -N[0]^{-1}$
- $B = 2^{2n} \bmod N$
- n = number of bits in the modulus

The process of exponentiation destroys the value in the A location and the value in the B location, therefore A and B have to be reloaded after each exponentiation. The done bit is set when the hardware completes the exponentiation function and the interrupt output pin is driven active until the done bit is written to a zero. The result (in integer form) is returned in the B location of the input RAM.

The exponentiation function requires that the K and B values be precomputed (per the equations above), then loaded into the multiplier's input RAM. The software must then load the Initial A (base), N (modulus) and E (exponent) values into their respective input RAM locations. The number of bits in the exponent must be set in the Number of Bits Register prior to exponentiation and the lower 8 bits of the Control/Status Register must be loaded with the value of 40H to start exponentiation.

5.2.1 Performance

Exponentiation Performance

Exponent Bits	Modulus Bits	Base Bits	Exponentiation Time	
			66 MHz	80 MHz
1024	1024	1024	< 55 ms	< 45 ms
224	1024	1024	< 12 ms	< 10 ms
160	1024	1024	< 9 ms	< 7 ms

5.2.2 Register Summary

Exponentiator Register Summary

Register	Address	Access	Description	Reset Value
K Register	0x200	R/W	K Value	00000000h
Reserved	0x201	R/W	Reserved	00000000h
Reserved	0x202	R/W	Reserved	00000000h
Reserved	0x203	R/W	Reserved	00000000h
Control & Status Register	0x204	R/W	Control and Status Register	00000000h
Number of Bits	0x205	R/W	Number of bits in the exponent Register	00000000h
Reserved	0x206	R/W	Reserved	00103210h
Reserved	0x100-0x120	R/W	Reserved	N/A
Operand RAM	0x080-0x0FF	R/W	Operand	N/A

5.2.2.1 K Register

The K value (32 bits) is stored in this register. The value is the pre-computed value of $K = -N[0]^{-1}$. K is the negative modular inverse of the least significant word of the modulus N. Word size is determined by the bit width of the underlying multiplier, i.e. 32-bits. The value K must be reloaded into this register every time the modulus changes.

5.2.2.1.1 K Value Computation

The following code segment is sample C code used to calculate the K value. The code is not optimized and is provided for explanatory purposes.

```
unsigned long kcalc (unsigned long n0)
{
    int i;
    unsigned long k;
    unsigned long pwr1, pwr2;
    unsigned long x;
```

```
    /* get the two's compliment */
    x = ~n0 + 1;
    k = 1;
    pwr1 = 2;
    pwr2 = 1;
    for (i=2; i < 32; i++)
    {
        pwr1 = pwr1 << 1;
        pwr2 = pwr2 << 1;
        if (((x*k) % pwr1) < pwr2)
            k = k;
        else
            k =k + pwr2;
    }
    pwr2 = pwr2 << 1;
    if ( (x*k) < pwr2 )
        k = k;
    else
        k = k + pwr2;
    return (k);
}
```

5.2.2.2 Control and Status Register

The Control/Status register provides the command interface to an external processor. The bits in this register are active during the exponentiation even if the functions are used by the exponentiator state machine.

Control/Status Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Done	Exponent	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Bit 31:8 Reserved

Bit 7 Done - State Machine Finished - This bit reports the completion of a function operation. The value of this bit corresponds to the interrupt output pin value.

Bit 6 Exponent - Exponentiate - Writing a one to this bit begins exponentiation. The

other functions should not be enabled when doing an exponentiation. The exponentiation state machine uses the A, B, N, and E values.

Bit 5:0 Reserved

5.2.2.3 Number of Bits In the Exponent Register

Only the lower 10 bits (bits[9:0]) of this register are used. In this register a value of zero represents a 1024 bit exponent. All other values are equal to their real value. The number of bits that are in the exponent is written to the register prior to an exponentiation. *This allows software to limit or prescan the exponent to increase performance of exponentiation time by not requiring the hardware to do squaring operations on the final exponents.*

5.3 External Interface

The SSRAM interface is the main interface between the security coprocessor chip, external memory and the external processor. Address, data, commands and control signals will flow through this interface.

5.3.1 Register Summary

This section describes the registers required to interface with the security coprocessor.

External Interface Register Summary

Register	Address	Access	Description	Reset Value
Interrupt Status Register	0x400	R/WOC	Interrupt Status Register	00000000h
Interrupt Enable Register	0x401	R/W	Interrupt Enable Register	00000000h
Configuration Register	0x402	R/W	Configuration Register	00000000h

WOC - Write one to clear.

5.3.1.1 Interrupt Status Register

The interrupt output is generated from the bitwise logical AND of the Interrupt Enable and Status registers and a logical OR of the resulting bits. That is, if any bit in the resulting bitwise AND is set, then an interrupt is generated. The interrupt output is active high and should be used as a level sensitive interrupt by the host.

Interrupt Status Register Configuration

7	6	5	4	3	2	1	0
Expo Done	Reserved	Reserved	ORDY	IRDY	Packet Done	Hash Done	DES Done

Bits 31-8 Reserved

Bit 7 Exponentiator Done - Exponentiator calculation is complete and output is ready. This bit can be cleared in the Multiplier Control and Status Register by setting bit7 to '0'.
0 - Exponentiation is not complete or has not been started.
1 - Exponentiation is complete.

Bit 6-5 Reserved

Bit 4 ORDY - Processed data is available to be read. This is equivalent to the ORDY output pin. The ORDY output pin is not latched in this register.
0 - ORDY is low.
1 - ORDY is high.

Bit 3 IRDY - The VMS115 is ready to receive incoming data. This is equivalent to the IRDY output pin. The IRDY output pin is not latched in this register.
0 - IRDY is low.
1 - IRDY is high.

Bit 2 Packet Done - Indicates the entire packet (Total Length in Opcode Register value) has been processed, read back and HMAC is done. This bit is reset with a Context Write or can write '1' to clear (WOC).
0 - Total Length is not expired or HMAC is not done with the packet.
1 - Total Length has expired and HMAC is done with the packet.

Bit 1 Hash Done - Indicates Hash Digest is valid. This bit is reset with a Context Write or can write '1' to clear (WOC).
0 - Hash Digest is not valid.
1 - Hash Digest is valid.

Bit 0 DES Done - Indicates the DES processing has completed. This bit is reset with a Context Write or can write '1' to clear (WOC).
0 - DES is not complete.
1 - DES is complete.

5.3.1.2 Interrupt Enable Register

Interrupt Enable Register Configuration

7	6	5	4	3	2	1	0
Expo Done	Reserved	Reserved	ORDY	IRDY	Packet Done	Hash Done	DES Done

Bits 31-8 Reserved

Bit 7 Exponentiator Done - Bit to enable the Exponentiator Done interrupt.
0 - Expo Done Interrupt is disabled.
1 - Expo Done Interrupt is enabled.

Bits 6-5 Reserved

Bit 4 ORDY - Bit to enable the ORDY interrupt.
0 - ORDY Interrupt disabled.
1 - ORDY Interrupt enabled.

Bit 3 IRDY -Bit to enable the IRDY interrupt.
0 - IRDY Interrupt disabled.
1 - IRDY Interrupt enabled.

Bit 2 Packet Done - Bit to enable the Packet Done interrupt.
0 - Packed Done Interrupt disabled.
1 - Packet Done Interrupt enabled.

Bit 1 Hash Done - Bit to enable the Hash Done interrupt.
0 - Hash Done Interrupt disabled.
1 - Hash Done Interrupt enabled.

Bit 0 DES Done - Bit to enable the DES Done interrupt.
0 - DES Done Interrupt disabled.
1 - DES Done Interrupt enabled.

5.3.1.3 Configuration Register

Configuration Register Configuration

15	14	13	12	11	10	9	8
Chip ID							

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Soft Reset

Bits 31-16 Reserved
Bits 15:8 Chip ID - This byte identifies the version of the chip.
Bits 7-1 Reserved
Bit 0 Soft Reset - Reset the VMS115 chip. Equivalent to the Reset input pin. When the reset is complete, this bit is set to 0.

5.3.2 Reset

Reset can be initiated via the input Reset pin going high or the Reset bit in the Configuration Register above. When resetting via the input Reset pin, the input must be held high for a minimum of one clock cycle to ensure that the VMS115 will exit Reset properly.

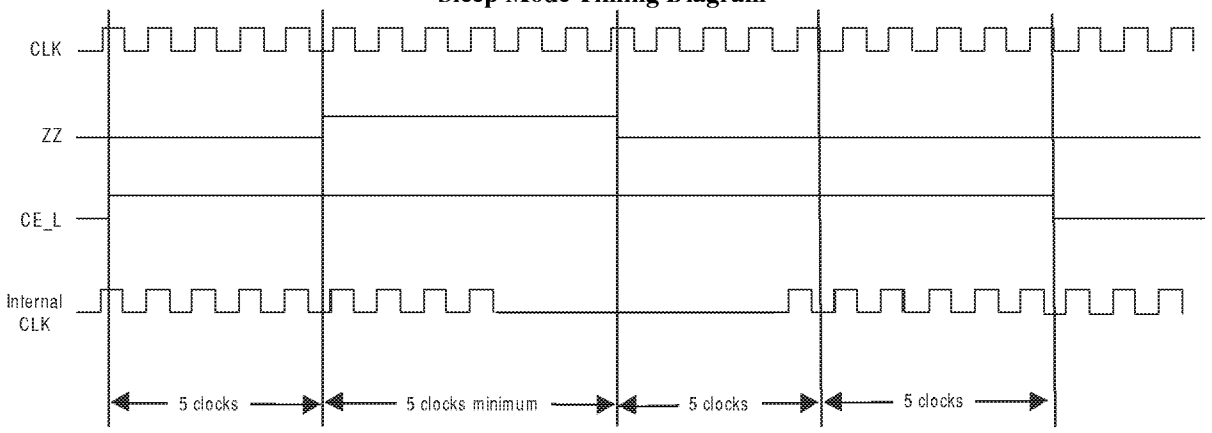
The RESET input signal can be asserted and de-asserted asynchronously to the clock input signal. The VMS115 will re-synchronize the RESET input and hold the part in reset for at least 3 clocks. Thus, the VMS115 will be ready to respond to external commands (register, context, or packet) within 4 clocks of RESET being de-asserted or 6 clocks after a write to the soft reset bit in the Configuration register.

5.3.3 Power Management

The ZZ input (active high) is used to put the VMS115 part into a low power mode. The ZZ input can be asserted asynchronously to the clock input signal. The ZZ input will be re-synchronized to the clock input and then gated into the master clock line prior to the clock tree of the device. Internal logic will guarantee that the clock is stopped on a low phase and effectively 'stretched' without glitching. The internal VMS115 clock will be stretched within 5 clocks of the assertion of the ZZ input. The internal VMS115 clock will resume within 5 clocks of the de-assertion of the ZZ input.

The VMS115 makes no assumption of the contents of the internal data after a power down cycle unless the interface is not active throughout the power down cycle. The chip select input must be de-asserted 5 clocks prior to the assertion of ZZ and remain inactive until 5 clocks after the de-assertion of ZZ to ensure that the internal data states are not corrupted by the power-down cycle. One exception to this is the Exponentiator Result RAM contents which are NOT valid following a sleep mode cycle.

Sleep Mode Timing Diagram



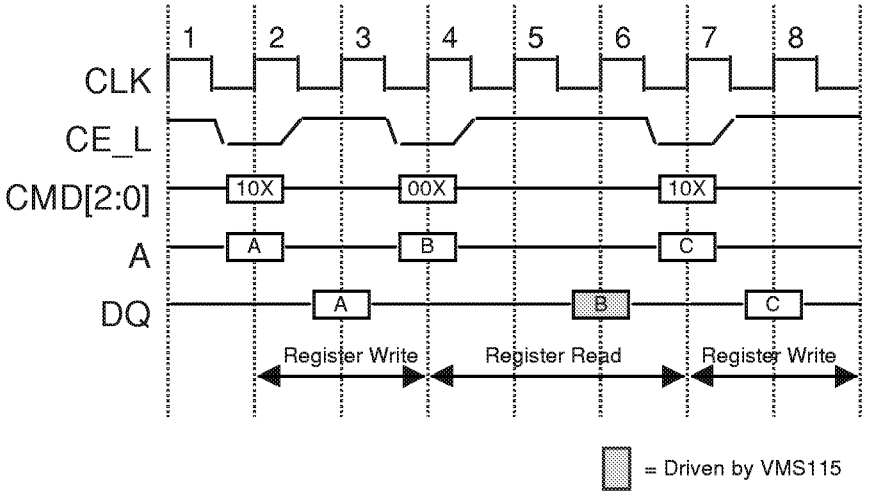
5.3.4 Data Transfer

This section includes timing diagrams for external access of the VMS115 chip. Diagrams are included for Single Register Access, Multiple Register Access, Context Write, Context Read, Packet Data Write, Packet Data Read and Back-to-Back Access.

5.3.4.1 Single Register Access

The external interface to the VMS115 consists of three types of commands in which data can be transferred into and out of the VMS115 device. The commands are determined by the CMD[2:0] inputs. The first command type is the register access. The register command will use the address A[10:0]to decode the location within the register space of the VMS115. The address input to the VMS115 is ignored for all commands other than the register command.

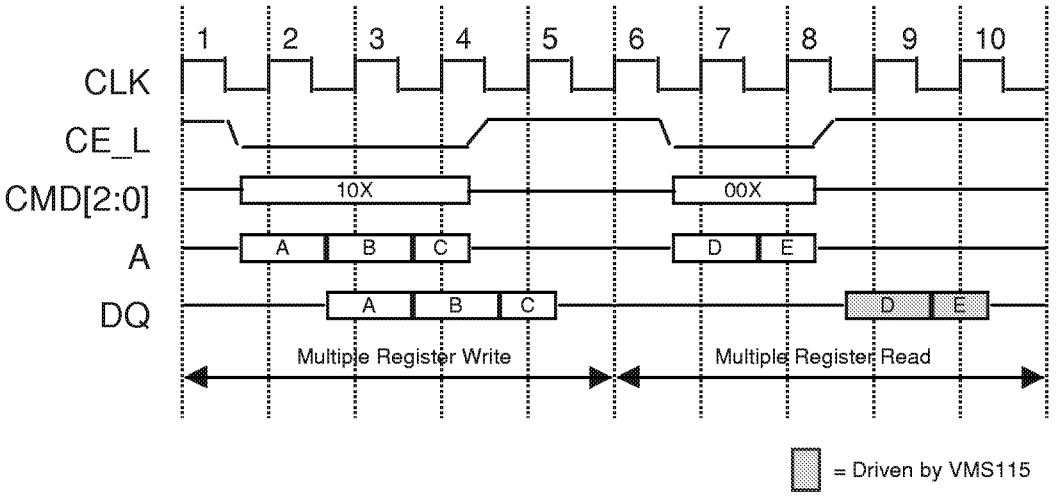
Single Register Access



5.3.4.2 Multiple Register Access

The register access can be extended for multiple clocks by keeping the CE_L chip select low and a constant value on the CMD[2:0] inputs. The address is decoded for each clock of a multi-register command access.

Multiple Register Access



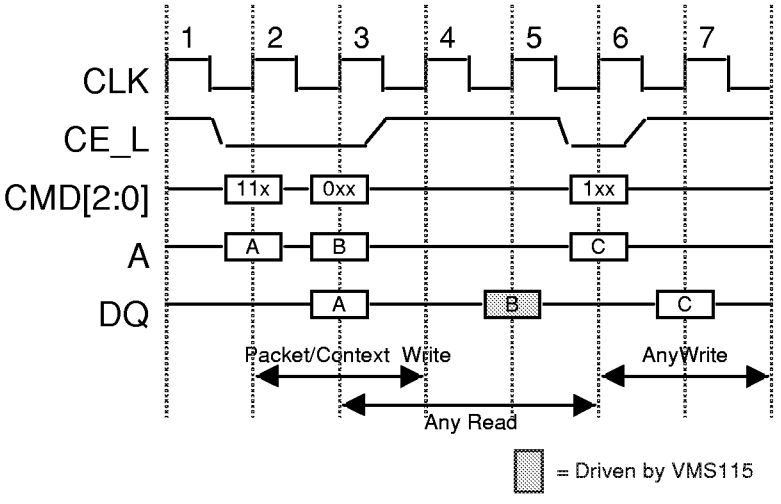
5.3.4.3 Back-to-Back Access

The CMD[2] bit can be treated as a read/write signal for the VMS115 device. A back to back access is defined as an access where the CMD[2] bit changes. The CMD[1:0] bits can change on any clock cycle.

5.3.4.3.1 Back-to-Back Packet/Context Access

The transition from a Packet/Context write (CMD[2:1] = '11') to a any read (CMD[2:1] = '0x') can occur from one clock cycle to the next. A transition from any read to any write must be separated by two (2) clocks of CE_L = '1' and no bus activity to ensure that there is no data contention.

Back-to-Back Packet/Context Access

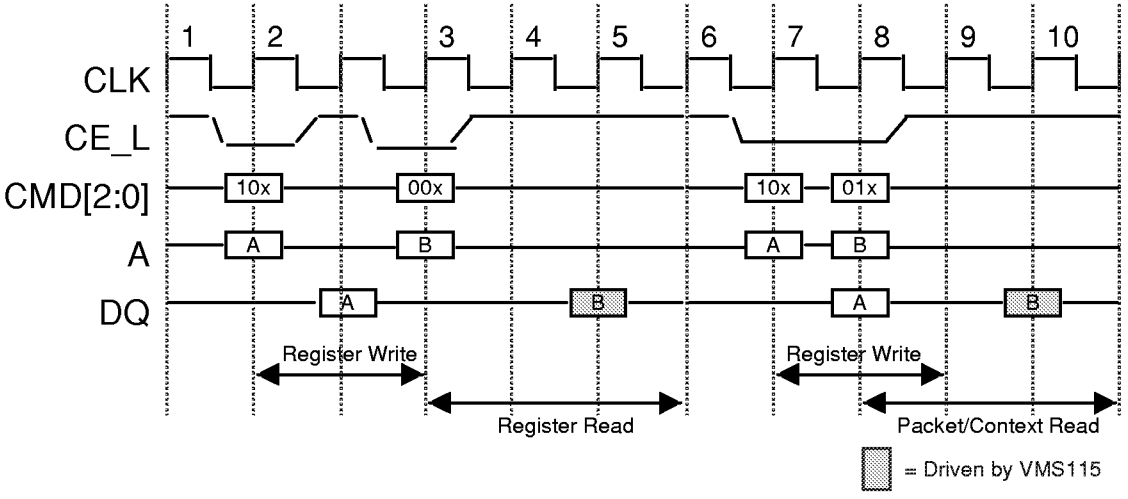


NOTE: Read followed by a write requires 2 clocks to ensure no data contention

5.3.4.3.2 Back-to-Back Register Access

The transition from a Register write (CMD[2:0] = '10') to a Register read (CMD[2:1] = '00') must be separated by one clock of CE_L = '1'. A transition from a Register Write (CMD[2:1] = '10') to a Packet/Context Read (CMD[2:1] = '01') can occur from one clock cycle to the next. When performing a register write to either Context Pointer, the Context Pointer will be updated two clocks following the register write data.

Back-to-Back Register Access

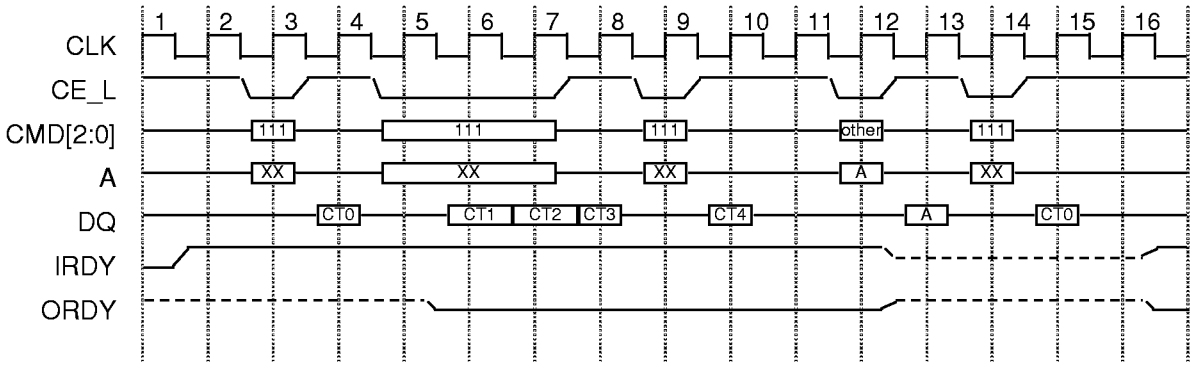


5.3.4.4 Context Write

The context data write can be performed at any time without regard to the IRDY signal (see timing diagram). However, the IRDY signal is cleared after a packet has been processed and the context (HMAC digest) has been read from the VMS115 device. Therefore, the IRDY signal can be used by the DMA to queue up the next context to be sent once IRDY is asserted. The context write uses the context write pointer to determine the word (4 bytes) within the context table that is updated. A register command (read or write) will not clear the context write pointer. The context write pointer will be reset to zero when any command other than a context data write or register command occurs.

The context data write command will flush the state machines and ensure that the VMS115 returns to a beginning of packet state. The contents of the context data are not flushed on a context write. Therefore, only the information that changes from packet to packet needs to be changed in the context write table.

Context Write Timing Diagram

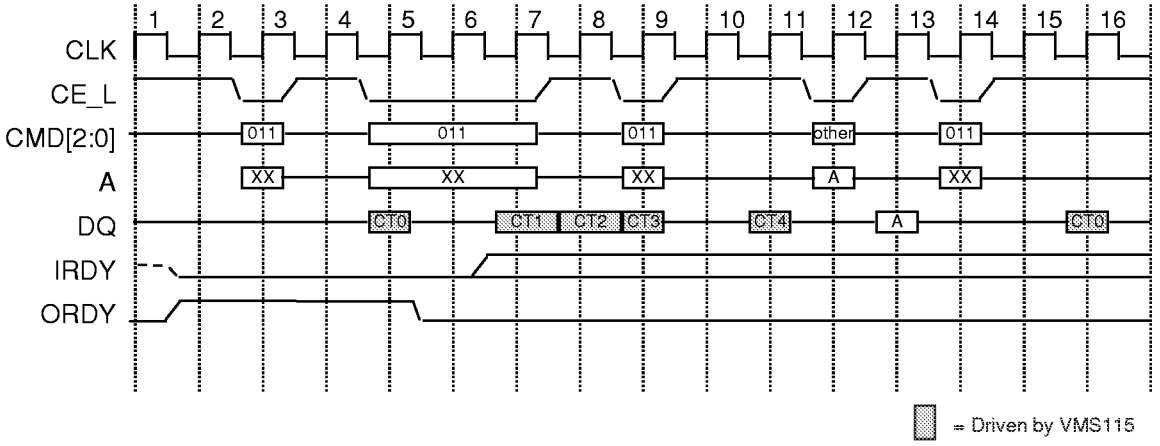


Note: 'other' is any code other than context write (110) or register (X00) and context pointer will reset to zero
Note: context write will flush all state machines and reset ORDY and IRDY

5.3.4.5 Context Read

The context data read can be performed at any time without regard to the ORDY signal (see timing diagram). However, the ORDY signal is used at the end of a packet to signal that the HMAC digest is ready to be read as part of the context. Therefore, the ORDY signal can be used to queue a DMA transaction that will read the HMAC digest at the proper time. The context read pointer is used to determine the value that is read from the context read table. The context read pointer is incremented after every read of the context read table. A register access will not reset the context read pointer.

Context Read Timing Diagram



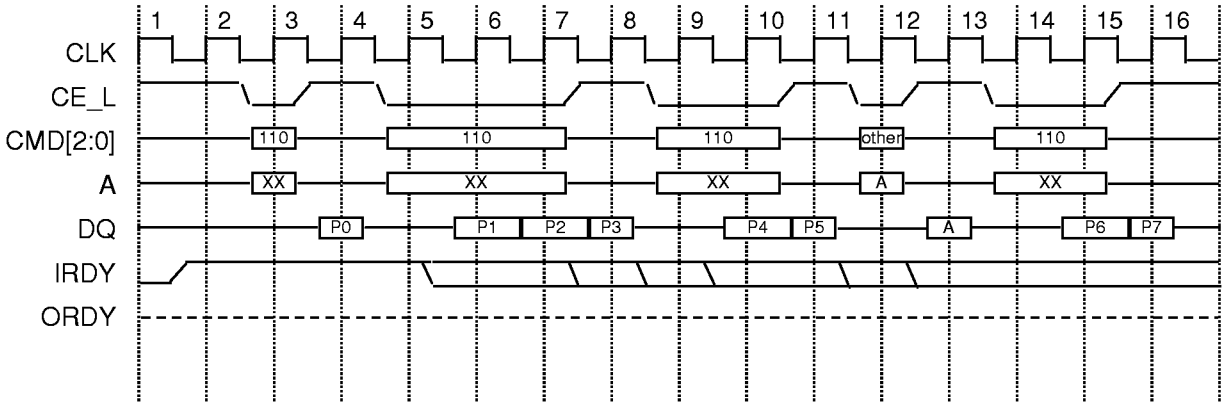
Note: 'other' is any code other than context read (010) or register (X00) and context pointer will reset to zero

5.3.4.6 Packet Data Write

The packet data is transferred into the VMS115 device with the packet data write command. The IRDY signal must be high before starting the packet data write command (see timing diagram). If

IRDY is high, the host can (and should unless at end of packet) transfer 8 words (32 bytes) of information into the VMS115 before re-sampling IRDY. The packet data write may ignore the ORDY signal. The IRDY will be updated based on having 32 or more bytes available in the In-Place Buffer. The IRDY signal will be valid one clock after the write data is sampled.

Packet Data Write Timing Diagram

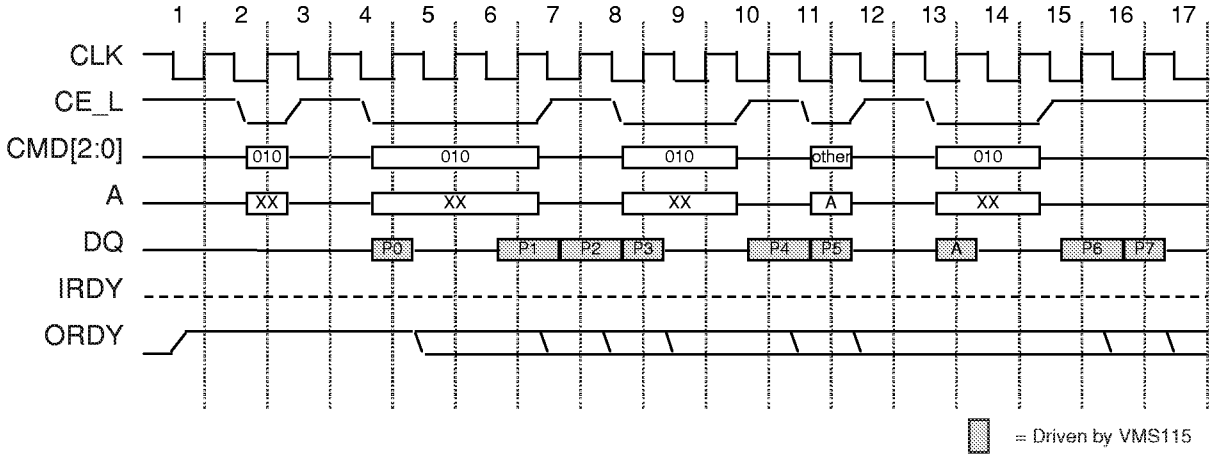


Note: IRDY should only be sampled before P0-P7 transfer is started
Note: 8 words (32 bytes) must be transferred once a write has started before IRDY is re-sampled

5.3.4.7 Packet Data Read

The packet data read is used to retrieve data from the VMS115. The ORDY signal used to signal that 32 bytes or more (except for end of packet) are ready to be read from the VMS115 (see timing diagram). The ORDY output will be updated one clock after the read data is transferred out of the VMS115. The IRDY signal can be ignored for packet data read commands. If ORDY is high, the host can (and should except for end of packet) transfer 8 words (32 bytes) of information out of the VMS115 before re-sampling the ORDY output.

Packet Data Read Timing Diagram

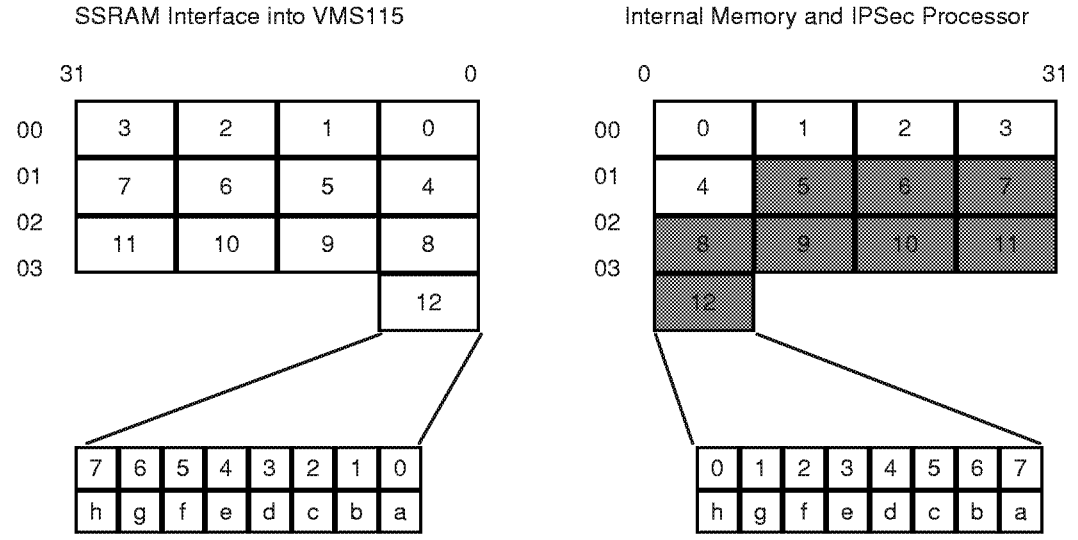


Note: ORDY should only be sampled before P0-P7 transfer is started
Note: 8 words (32 bytes) must be transferred once a read has started before ORDY is re-sampled

5.3.5 Data Format

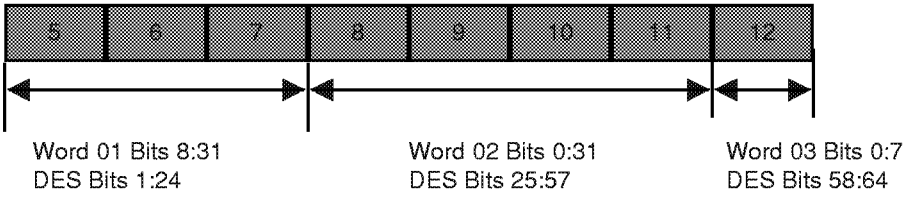
The data format into and out of the VMS115 chip is little endian. The data will be switched to big endian for storage internally to the chip. All of the register accesses will be little endian access (since there are no byte enables, the bit placement is the only information that has bearing).

Data Format (Endianess) Diagram



Bit 'a' is the first bit received within the byte. The data will be stored in the Hash Block of the IPSec Processor in the same format as the internal memory. The MD5 (little) and SHA-1 (big) algorthim will process the data according to the hash specification for endianness.

DES operation on bytes 5 to 12 will generate a word per the FIPS DES specfication



5.3.5.1 Data Format Examples and Test Cases

The following test cases represent the data format of data into and out of the VMS115 chip. These test cases simulate realistic operations that the VMS115 chip may perform and provide the input and output data in its proper format. The test cases provided are:

1. Encrypt - ECB - DES
2. Encrypt - CBC- DES
3. Encrypt - CBC - 3DES
4. Decrypt - CBC - 3DES
5. Hash - SHA-1
6. Hash - MD5

7. HMAC - SHA-1 (with pre-compute)
8. HMAC - MD5 (with pre-compute)

5.3.5.1.1 Test Case 1: Encrypt - ECB - DES

Context Data Values

Context Field	Value
Total Byte Length	0x001D (29)
HASH Byte Length	0x0000 (0)
HASH Byte Offset	0x0000 (0)
DES Byte Length	0x0018 (24)
DES Byte Offset	0x0003 (3)
HASH Mode	DISABLED-ZERO-LENGTH
HASH Type	MD5
DES Mode	ENCRYPT-ECB-DES

Context Write/Read Data

Context Field	Write Data	Read Data
Digest	n/a n/a n/a n/a n/a	0x67452301 0xEFCDAB89 0x98BADCFE 0x10325476 0xF0E1D2C3
OpCode	0x001D001A	same
Length	0x00000018	same
Offset	0x00000003	same
DES IV	0x78563412 0xEFCDAB90	same same
IPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
OPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same

Context Write/Read Data

Context Field	Write Data	Read Data
DES Key 0	0x67452301 0xEFCDAB89	same same
DES Key 1	0x00000000 0x00000000	same same
DES Key 2	0x00000000 0x00000000	same same

Packet Information

Packet	Write Data	Read Data
d . . .	4E 33 32 31	3F -- -- --
d d d d	69 20 77 6F	98 8A 0E A4
d d d d	68 74 20 73	6A 15 48 4D
d d d d	69 74 20 65	AB 87 17 27
d d d d	66 20 65 6D	89 F9 83 88
d d d d	61 20 72 6F	4B EC 51 3D
. d d d	2E 20 6C 6C	-- 53 3B 56
* * * .	** ** ** 0A	** ** ** --

5.3.5.1.2 Test Case 2: Encrypt - CBC - DES

Context Data

Context Field	Value
Total Byte Length	0x001D (29)
HASH Byte Length	0x0000 (0)
HASH Byte Offset	0x0000 (0)
DES Byte Length	0x0018 (24)
DES Byte Offset	0x0003 (3)
HASH Mode	DISABLED-ZERO-LENGTH
HASH Type	SHA-1
DES Mode	ENCRYPT-CBC-DES

Context Write/Read Data

Context Field	Write Data	Read Data
Digest	n/a n/a n/a n/a n/a	0x1234567 0x89ABCDEF 0xFEDCBA98 0x76543210 0xF0E1D2C3
OpCode	0x001D0016	same
Length	0x00000018	same
Offset	0x00000003	same
DES IV	0x78563412 0xEFCDAB90	0x49883768 0xF6057C9A
IPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
OPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
DES Key 0	0x67452301 0xEFCDAB89	same same

Context Write/Read Data

Context Field	Write Data	Read Data
DES Key 1	0x00000000 0x00000000	same same
DES Key 2	0x00000000 0x00000000	same same

Packet Information

Packet	Write Data	Read Data
d . . .	4E 33 32 31	E5 -- -- --
d d d d	69 20 77 6F	87 DE CD C7
d d d d	68 74 20 73	43 7C F2 2B
d d d d	69 74 20 65	8C 00 34 E9
d d d d	66 20 65 6D	68 0F 9C 38
d d d d	61 20 72 6F	9A 49 88 37
. d d d	2E 20 6C 6C	-- F6 05 7C
* * * .	** ** ** 0A	** ** ** --

5.3.5.1.3 Test Case 3: Encrypt - CBC - 3DES

Context Data

Context Field	Value
Total Byte Length	0x001D (29)
HASH Byte Length	0x0000 (0)
HASH Byte Offset	0x0000 (0)
DES Byte Length	0x0018 (24)
DES Byte Offset	0x0003 (3)
HASH Mode	DISABLED-ZERO-LENGTH
HASH Type	SHA-1
DES Mode	ENCRYPT-CBC-3DES

Context Write/Read Data

Context Field	Write Data	Read Data
Digest	n/a n/a n/a n/a n/a	0x1234567 0x89ABCDEF 0xFEDCBA98 0x76543210 0xF0E1D2C3
OpCode	0x001D0017	same
Length	0x00000018	same
Offset	0x00000003	same
DES IV	0x78563412 0xEFCDAB90	0x2DFE3DA2 0xE1818BAD
IPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
OPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
DES Key 0	0x67452301 0xEFCDAB89	same same

Context Write/Read Data

Context Field	Write Data	Read Data
DES Key 1	0x07060504 0x03020100	same same
DES Key 2	0x88776655 0x44332211	same same

Packet Information

Packet	Write Data	Read Data
d . . . d d d d d d d d d d d d d d d d d d d d . d d d * * * .	4E 33 32 31 69 20 77 6F 68 74 20 73 69 74 20 65 66 20 65 6D 61 20 72 6F 2E 20 6C 6C ** ** ** 0A	1A -- -- -- CC 6F 26 8E E6 75 46 F3 9C B6 F5 42 A2 01 D7 B8 AD 2D FE 3D -- E1 81 8B ** ** ** --

5.3.5.1.4 Test Case 4: Decrypt - CBC - 3DES

Context Data

Context Field	Value
Total Byte Length	0x001D (29)
HASH Byte Length	0x0000 (0)
HASH Byte Offset	0x0000 (0)
DES Byte Length	0x0018 (24)
DES Byte Offset	0x0003 (3)
HASH Mode	DISABLED-ZERO-LENGTH
HASH Type	SHA-1
DES Mode	DECRYPT-CBC-3DES

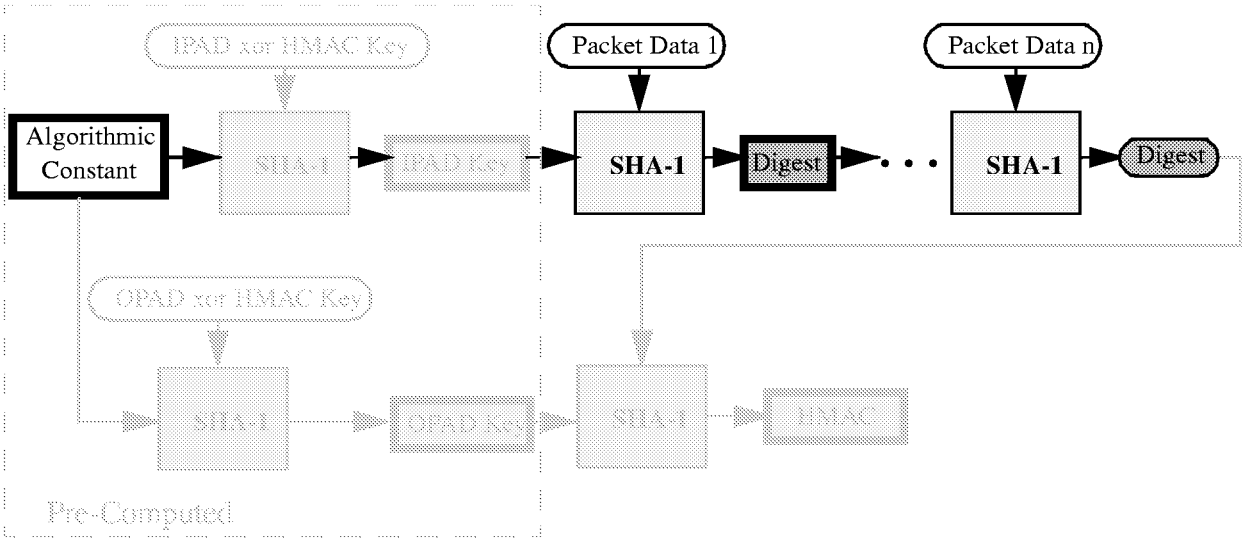
Context Write/Read Data

Context Field	Write Data	Read Data
Digest	n/a n/a n/a n/a n/a	0x01234567 0x89ABCDEF 0xFEDCBA98 0x76543210 0xF0E1D2C3
OpCode	0x001D0015	same
Length	0x00000018	same
Offset	0x00000003	same
DES IV	0x78563412 0xEFCDAB90	0x2DFE3DA2 0xE1818BAD
IPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
OPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
DES Key 0	0x67452301 0xEFCDAB89	same same
DES Key 1	0x07060504 0x03020100	same same
DES Key 2	0x88776655 0x44332211	same same

Packet Information

Packet	Write Data	Read Data
d . . .	1A 33 32 31	4E -- -- --
d d d d	CC 6F 26 8E	69 20 77 6F
d d d d	E6 75 46 F3	68 74 20 73
d d d d	9C B6 F5 42	69 74 20 65
d d d d	A2 01 D7 B8	66 20 65 6D
d d d d	AD 2D FE 3D	61 20 72 6F
. d d d	2E E1 81 8B	-- 20 6C 6C
* * * .	** ** ** 0A	** ** ** --

5.3.5.1.5 Test Case 5: Hash - SHA-1



Context Data

Context Field	Value
Total Byte Length	0x0040 (64)
HASH Byte Length	0x0038 (56)
HASH Byte Offset	0x0000 (0)
DES Byte Length	0x0000 (0)
DES Byte Offset	0x0000 (0)
HASH Mode	HASH
HASH Type	SHA-1
DES Mode	DISABLED-ZERO-LENGTH

Context Write/Read Data

Context Field	Write Data	Read Data
Digest	n/a n/a n/a n/a n/a	0x443E9884 0x6ED23B1C 0xA14AAEBA 0xE52951F9 0xF17046E5
OpCode	0x00400020	same

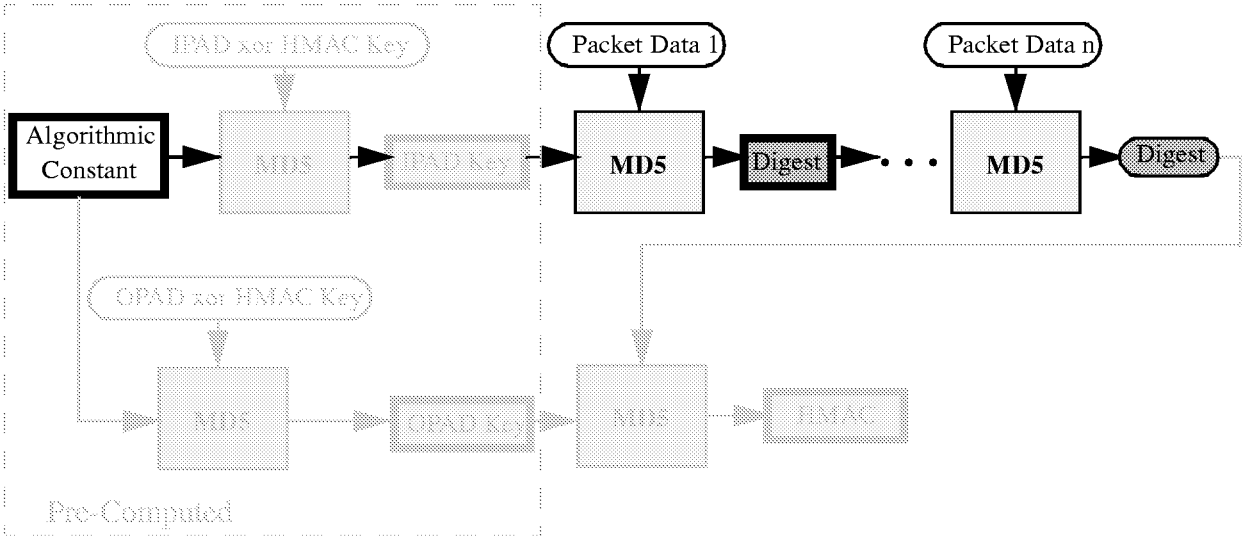
Context Write/Read Data

Context Field	Write Data	Read Data
Length	0x00380000	same
Offset	0x00000000	same
DES IV	0x00000000 0x00000000	same same
IPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
OPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
DES Key 0	0x00000000 0x00000000	same same
DES Key 1	0x00000000 0x00000000	same same
DES Key 2	0x00000000 0x00000000	same same

Packet Information

Packet	Write Data	Read Data
h h h h	64 63 62 61	-- -- -- --
h h h h	65 64 63 62	-- -- -- --
h h h h	66 65 64 63	-- -- -- --
h h h h	67 66 65 64	-- -- -- --
h h h h	68 67 66 65	-- -- -- --
h h h h	69 68 67 66	-- -- -- --
h h h h	6A 69 68 67	-- -- -- --
h h h h	6B 6A 69 68	-- -- -- --
h h h h	6C 6B 6A 69	-- -- -- --
h h h h	6D 6C 6B 6A	-- -- -- --
h h h h	6E 6D 6C 6B	-- -- -- --
h h h h	6F 6E 6D 6C	-- -- -- --
h h h h	70 6F 6E 6D	-- -- -- --
h h h h	71 70 6F 6E	-- -- -- --
. . . .	00 00 00 0A	-- -- -- --
. . . .	00 00 00 00	-- -- -- --

5.3.5.1.6 Test Case 6: Hash - MD5



Context Data

Context Field	Value
Total Byte Length	0x0020 (32)
HASH Byte Length	0x000E (14)
HASH Byte Offset	0x0000 (0)
DES Byte Length	0x0000 (0)
DES Byte Offset	0x0000 (0)
HASH Mode	HASH
HASH Type	MD5
DES Mode	DISABLED-ZERO-LENGTH

Context Write/Read Data

Context Field	Write Data	Read Data
Digest	n/a n/a n/a n/a n/a	0x7D696BF9 0x8D93B77C 0x312F5A52 0xD061F1AA 0xF0E1D2C3
OpCode	0x00200028	same
Length	0x000E0000	same

Context Write/Read Data

Context Field	Write Data	Read Data
Offset	0x00000000	same
DES IV	0x00000000 0x00000000	same same
IPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
OPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
DES Key 0	0x00000000 0x00000000	same same
DES Key 1	0x00000000 0x00000000	same same
DES Key 2	0x00000000 0x00000000	same same

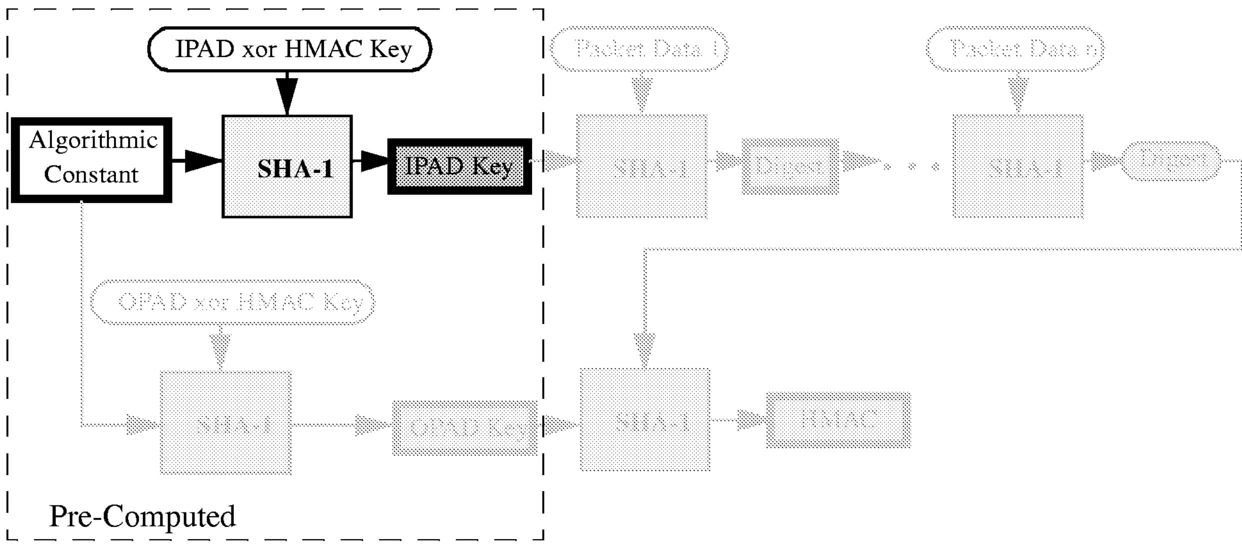
Packet Information

Packet	Write Data	Read Data
h h h h h h h h h h h h . . h h	73 73 65 6D 20 65 67 61 65 67 69 64 00 0A 74 73 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	-- --

5.3.5.1.7 Test Case 7: HMAC - SHA-1 (with pre-compute)

IPAD Computation

IPAD = 0x36363636



HMAC Generation Data

HMAC Key	IPAD xor HMAC Key	OPAD xor HMAC Key
0x0b0b0b0b	0x3d3d3d3d	0x57575757
0x0b0b0b0b	0x3d3d3d3d	0x57575757
0x0b0b0b0b	0x3d3d3d3d	0x57575757
0x0b0b0b0b	0x3d3d3d3d	0x57575757
0x0b0b0b0b	0x3d3d3d3d	0x57575757
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c

Context Data

Context Field	Value
Total Byte Length	0x0040 (64)
HASH Byte Length	0x0040 (64)
HASH Byte Offset	0x0000 (0)

Context Data

Context Field	Value
DES Byte Length	0x0000 (0)
DES Byte Offset	0x0000 (0)
HASH Mode	PRECOMPUTE
HASH Type	SHA-1
DES Mode	DISABLED-ZERO-LENGTH

Context Write/Read Data

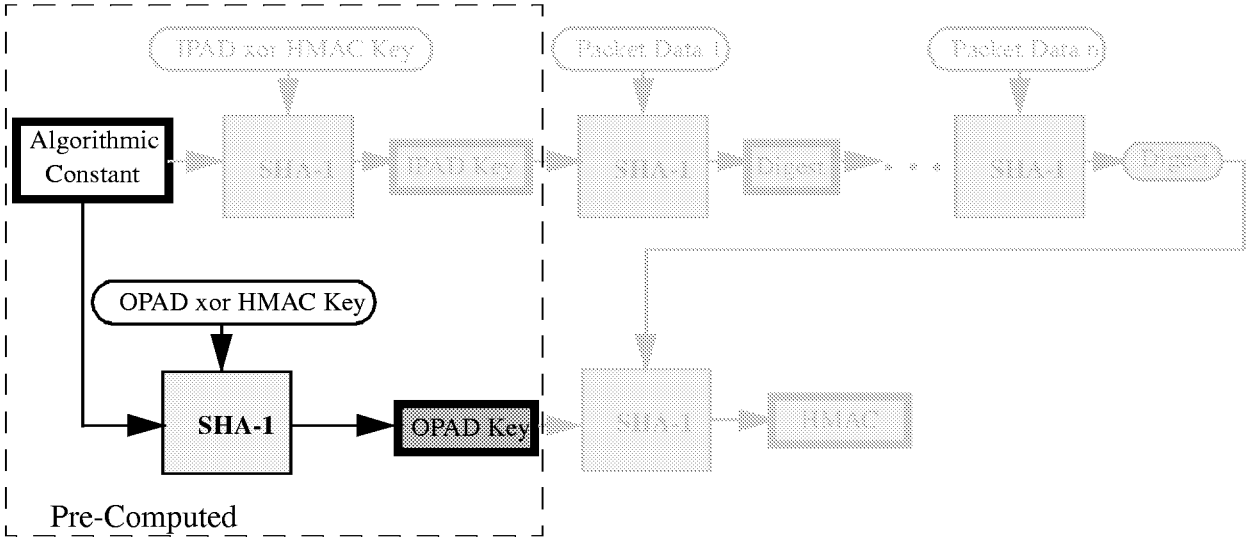
Context Field	Write Data	Read Data
Digest	n/a n/a n/a n/a n/a	0x2A664C06 0xB0D16ECC 0xB09AFA6C 0x7F132F04 0xEB70A5CE
OpCode	0x00400120	same
Length	0x00400000	same
Offset	0x00000000	same
DES IV	0x00000000 0x00000000	same same
IPAD Key	0x01234567 0x89ABCDEF 0xFEDCBA98 0x76543210 0xF0E1D2C3	same same same same same
OPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
DES Key 0	0x00000000 0x00000000	same same
DES Key 1	0x00000000 0x00000000	same same
DES Key 2	0x00000000 0x00000000	same same

Packet Information

Packet	Write Data	Read Data
h h h h	3D 3D 3D 3D	-- -- -- --
h h h h	3D 3D 3D 3D	-- -- -- --
h h h h	3D 3D 3D 3D	-- -- -- --
h h h h	3D 3D 3D 3D	-- -- -- --
h h h h	3D 3D 3D 3D	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --

OPAD Computation

OPAD = 0x5C5C5C5C



Context Data

Context Field	Value
Total Byte Length	0x0040 (64)
HASH Byte Length	0x0040 (64)

Context Data

Context Field	Value
HASH Byte Offset	0x0000 (0)
DES Byte Length	0x0000 (0)
DES Byte Offset	0x0000 (0)
HASH Mode	PRECOMPUTE
HASH Type	SHA-1
DES Mode	DISABLED-ZERO-LENGTH

Context Write/Read Data

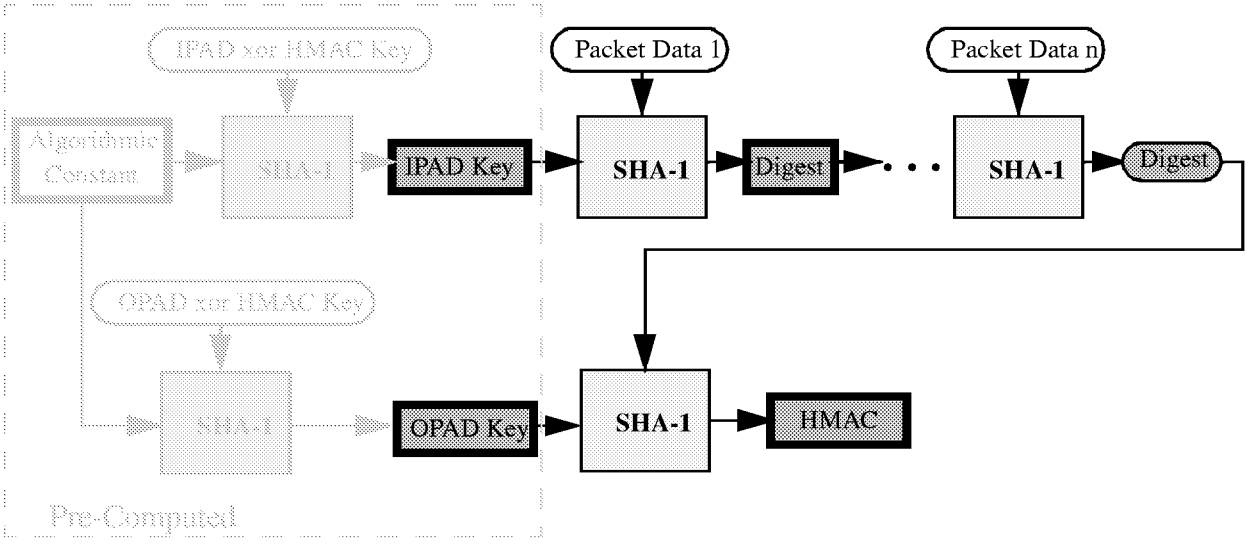
Context Field	Write Data	Read Data
Digest	n/a n/a n/a n/a n/a	0xCCCC2AD6 0xB3986A7E 0x025B01DF 0xE0C385D8 0xCB84D19B
OpCode	0x00400120	same
Length	0x00400000	same
Offset	0x00000000	same
DES IV	0x00000000 0x00000000	same same
IPAD Key	0x01234567 0x89ABCDEF 0xFEDCBA98 0x76543210 0xF0E1D2C3	same same same same same
OPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
DES Key 0	0x00000000 0x00000000	same same
DES Key 1	0x00000000 0x00000000	same same
DES Key 2	0x00000000 0x00000000	same same

Packet Information

Packet	Write Data	Read Data
h h h h	57 57 57 57	-- -- -- --
h h h h	57 57 57 57	-- -- -- --
h h h h	57 57 57 57	-- -- -- --
h h h h	57 57 57 57	-- -- -- --
h h h h	57 57 57 57	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --

echo 'Hi There' >! packet.txt

HMAC Computation



Context Data

Context Field	Value
Total Byte Length	0x0008 (8)
HASH Byte Length	0x0008 (8)

Context Data

Context Field	Value
HASH Byte Offset	0x0000 (0)
DES Byte Length	0x0000 (0)
DES Byte Offset	0x0000 (0)
HASH Mode	HMAC
HASH Type	SHA-1
DES Mode	DISABLED-ZERO-LENGTH

Context Write/Read Data

Context Field	Write Data	Read Data
Digest	n/a n/a n/a n/a n/a	0x863117B6 0x64720555 0xB6C08BE2 0x8E8C37FB 0x00BE46F1
OpCode	0x00080080	same
Length	0x00080000	same
Offset	0x00000000	same
DES IV	0x00000000 0x00000000	same same
IPAD Key	0x2A664C06 0xB0D16ECC 0xB09AFA6C 0x7F132F04 0xEB70A5CE	same same same same same
OPAD Key	0xCCCC2AD6 0xB3986A7E 0x025B01DF 0xE0C385D8 0xCB84D19B	same same same same same
DES Key 0	0x00000000 0x00000000	same same
DES Key 1	0x00000000 0x00000000	same same
DES Key 2	0x00000000 0x00000000	same same

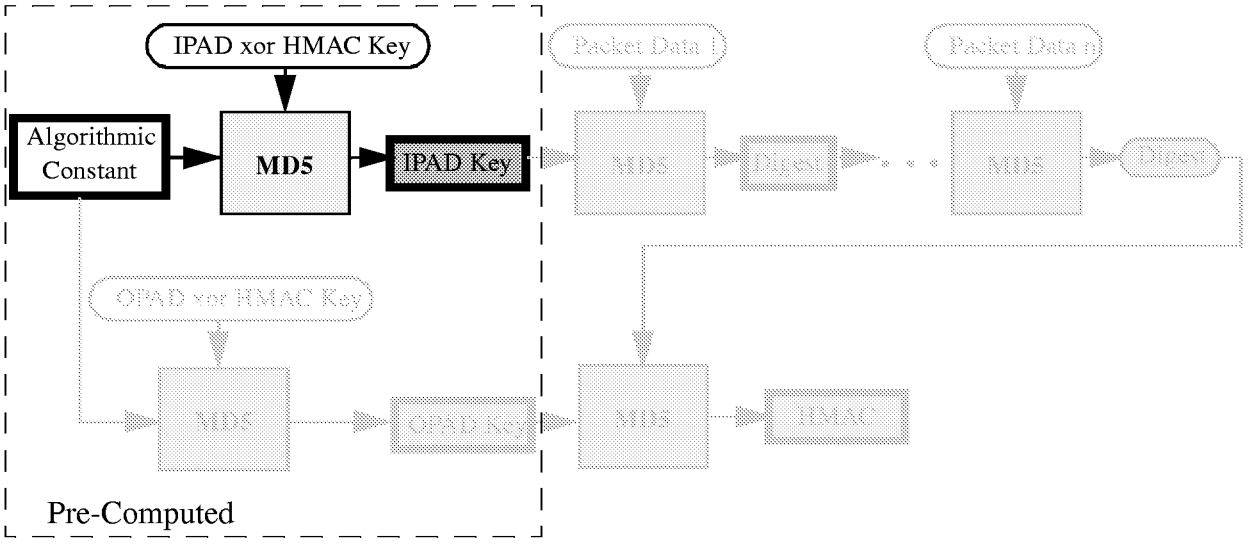
Packet Information

Packet	Write Data	Read Data
h h h h h h h h	54 20 69 48 65 72 65 68	-- -- -- -- -- -- -- --

5.3.5.1.8 Test Case 8: HMAC - MD5 (with pre-compute)

IPAD Computation

IPAD = 0x36363636



HMAC Generation Data

HMAC key	IPAD	OPAD
0x4a656665	0x7c535053	0x16393a39
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c
0x00000000	0x36363636	0x5c5c5c5c

Context Data

Context Field	Value
Total Byte Length	0x0040 (64)
HASH Byte Length	0x0040 (64)
HASH Byte Offset	0x0000 (0)
DES Byte Length	0x0000 (0)
DES Byte Offset	0x0000 (0)
HASH Mode	PRECOMPUTE
HASH Type	MD5
DES Mode	DISABLED-ZERO-LENGTH

Context Write/Read Data

Context	Write	Read
Digest	n/a n/a n/a n/a n/a	0xCB6180E8 0x2FA83EA8 0x43278D6C 0xB9526934 0xF0E1D2C3
OpCode	0x00400128	same

Context Write/Read Data

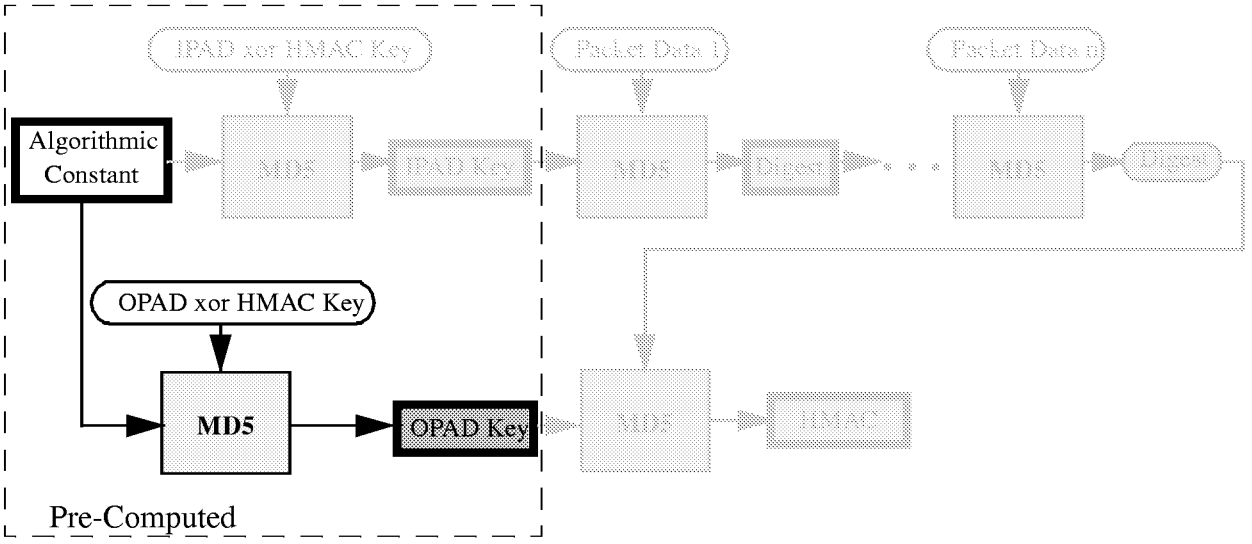
Context	Write	Read
Length	0x00400000	same
Offset	0x00000000	same
DES IV	0x00000000 0x00000000	same same
IPAD Key	0x67452301 0xEFCDAB89 0x98BADCFE 0x10325476 0x00000000	same same same same same
OPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
DES Key 0	0x00000000 0x00000000	same same
DES Key 1	0x00000000 0x00000000	same same
DES Key 2	0x00000000 0x00000000	same same

Packet Information

Packet	Write Data	Read Data
h h h h	53 50 53 7C	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --
h h h h	36 36 36 36	-- -- -- --

OPAD Computation

OPAD = 0x5C5C5C5C



Context Data

Context Field	Value
Total Byte Length	0x0040 (64)
HASH Byte Length	0x0040 (64)
HASH Byte Offset	0x0000 (0)
DES Byte Length	0x0000 (0)
DES Byte Offset	0x0000 (0)
HASH Mode	PRECOMPUTE
HASH Type	MD5
DES Mode	DISABLED-ZERO-LENGTH

Context Write/Read Data

Context	Write	Read
Digest	n/a n/a n/a n/a n/a	0x53F5670C 0x4D4CA5E3 0x0F66FACA 0x781F2AB7 0xF0E1D2C3
OpCode	0x00400128	same

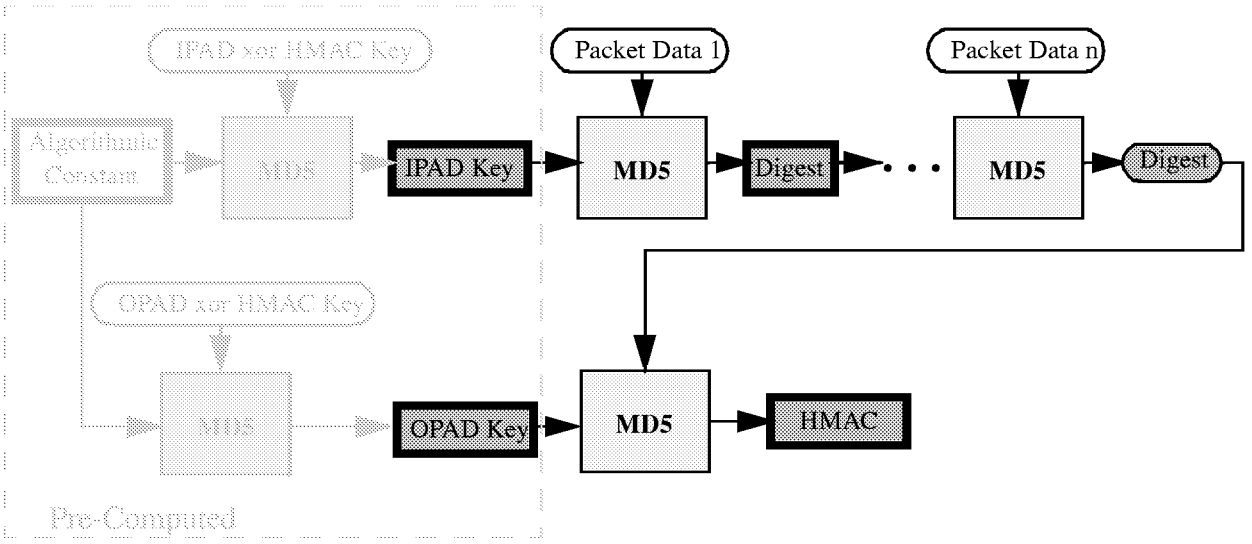
Context Write/Read Data

Context	Write	Read
Length	0x00400000	same
Offset	0x00000000	same
DES IV	0x00000000 0x00000000	same same
IPAD Key	0x67452301 0xEFCDAB89 0x98BADCFE 0x10325476 0x00000000	same same same same same
OPAD Key	0x00000000 0x00000000 0x00000000 0x00000000 0x00000000	same same same same same
DES Key 0	0x00000000 0x00000000	same same
DES Key 1	0x00000000 0x00000000	same same
DES Key 2	0x00000000 0x00000000	same same

Packet Information

Packet	Write Data	Read Data
h h h h	39 3A 39 16	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --
h h h h	5C 5C 5C 5C	-- -- -- --

HMAC Computation



Context Data

Context Field	Value
Total Byte Length	0x001D (29)
HASH Byte Length	0x001C (28)
HASH Byte Offset	0x0001 (1)
DES Byte Length	0x0000 (0)
DES Byte Offset	0x0000 (0)
HASH Mode	HMAC
HASH Type	MD5
DES Mode	DISABLED-ZERO-LENGTH

Context Write/Read Data

Context	Write	Read
Digest	n/a n/a n/a n/a n/a	0x3E780C75 0x03B5B06A 0x316EA8EA 0x38B75D0A 0xF0E1D2C3
OpCode	0x001D0088	same
Length	0x001C0000	same
Offset	0x00010000	same

Context Write/Read Data

Context	Write	Read
DES IV	0x00000000 0x00000000	same same
IPAD Key	0xE88061CB 0xA83EA82F 0x6C8D2743 0x346952B9 0xF0E1D2C3	same same same same same
OPAD Key	0x0C67F553 0xE3A54C4D 0xCAFA660F 0xB72A1F78 0xF0E1D2C3	same same same same same
DES Key 0	0x00000000 0x00000000	same same
DES Key 1	0x00000000 0x00000000	same same
DES Key 2	0x00000000 0x00000000	same same

Packet Information

Packet	Write Data	Read Data
h h h .	61 68 77 25	-- -- -- --
h h h h	6F 64 20 74	-- -- -- --
h h h h	20 61 79 20	-- -- -- --
h h h h	74 6E 61 77	-- -- -- --
h h h h	72 6F 66 20	-- -- -- --
h h h h	74 6F 6E 20	-- -- -- --
h h h h	67 6E 69 68	-- -- -- --
* * * h	** ** ** 3F	** ** ** --

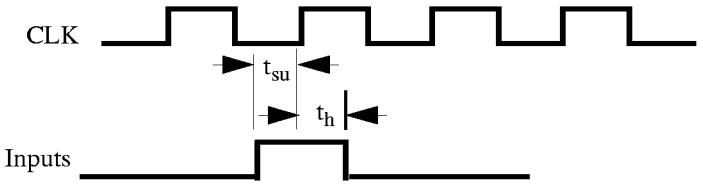
6 AC Parameters and Timing

This section describes the timings for each of the VMS115 interfaces. Propagation delays, setup and hold times are specified for each interface. The following diagrams illustrate the timings relative to CLK and the tables define the timings for each specific interface.

All timings assume 3.3 +/- 5% volts at 0^0 to 125^0 C junction and 0^0 to 70^0 C ambient.

6.1 Input Timing

Setup and Hold Timing Diagrams



Input Timings Table

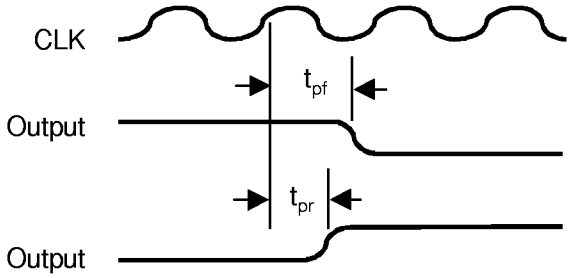
Signal	t_{su} Setup Time (nS)	t_h Hold Time (nS)	Notes
CE_L	2.5	1.5	
CMD	2.5	1.5	
A	2.5	1.5	
DQ	2.5	1.5	

Note: Input signals ZZ and RESET are asynchronous inputs.

6.2 Output Timings

6.2.1 Propagation Delay Timing

Propagation Delay Timing Diagram



t_{pf} - Propagation fall delay (high-to-low transition)

t_{pr} - Propagation rise delay (low-to-high transition)

Timings are measured between 50% of rise and fall points.

20pF Output Interface Propagation Delay Table

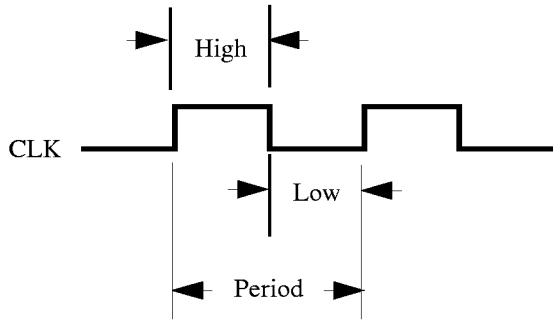
Signal	t_{pr} Propagation Delay		t_{pf} Propagation Delay		Notes
	Min (ns)	Max (ns)	Min (ns)	Max (ns)	
TxRDY	3.38	7.17	3.56	6.87	
RxRDY	3.35	7.15	3.52	6.82	
INT	3.25	6.96	3.30	6.44	
DQ	1.90	6.51	1.98	6.59	Min time results from PAD OEN transitioning from 0 to 1.

0pF Output Interface Propagation Delay Table

Signal	t_{pr} Propagation Delay		t_{pf} Propagation Delay		Notes
	Min (ns)	Max (ns)	Min (ns)	Max (ns)	
TxRDY	2.08	4.16	2.12	4.11	
RxRDY	2.05	4.15	2.07	4.05	
INT	1.96	3.99	1.87	3.72	
DQ	1.75	3.95	1.75	3.93	Min time results from PAD OEN transitioning from 0 to 1.

6.3 Clocks

This section defines the minimum low and minimum high times as well as the minimum clock period for the VMS115 clocks.



Clock Timings

Clock	Minimum Low (ns)	Minimum High (ns)	Minimum Period (ns)
CLK	6.25	6.25	12.5

7 DC Parameters

Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Units	Notes
VIP	Voltage Applied to Any Pin		VDD * 1.5	V	1
TS	Storage Temperature	-40°	+125°	C	1

Recommended DC Operating Conditions - 3.3 volts CMOS

Symbol	Parameter	Min	Max	VDD	Units	Notes
VIH	Input High Voltage	.7 * VDD	VDD + 0.3V	2.70 to 3.63V	V	2
VIL	Input Low Voltage	-0.5V	.3 * VDD	2.70 to 3.63V	V	2
VOH	Output High Voltage	VSS - 0.1V		2.70V	V	
VOL	Output Low Voltage		VSS + 0.1V	2.70V	V	
T	Operating Temperature	0°	+70°		C	

Recommended DC Operating Conditions - TTL Inputs

Symbol	Parameter	Min	Max	VDD	Units
VIH	Input High Voltage	2.0V	VDD + 0.5V	2.97 to 3.63V	V
VIL	Input Low Voltage	-0.5V	0.8V	2.97 to 3.63V	V
T	Operating Temperature	0°	+70° Ambient		C

DC Characteristics

Symbol	Parameter	Min	Max	Units	Notes
IDD	Supply Current		1.0	mA	
ILATCH	DC latch-up current 70° C	+/- 100		mA	
VLATCH	Latch-up voltage 70° C	VDD + 1.5V		V	
IIN	Input Leakage Current		10.0	µA	
CIN	Input Capacitance		3.922	pF	

Notes:

1. These are stress ratings only. Exceeding the absolute maximum ratings may permanently damage the device. Operating the device at absolute ratings for extended periods may affect device reliability, and void the warranty.
2. Voltages measured with respect to VSS.
3. All timings reference VLSI Technology, Inc. 0.35µ, 3.3V-Core, 3.3V I/O, Cell-Based Libraries-Rev. A

8 Power Dissipation

The estimated power dissipation for the VMS115 is defined in the following table. Power figures are included for 80MHz operation only.

VMS115 Power Dissipation

Mode	Power Dissipation	Switching Rate	Frequency	Vdd
Operating	360 mA	10%	80MHz	3.3 volts
Sleep	10 mA	0%	80MHz	3.3 volts

Note: Core is 3.3 volts, pads are 3.3 volts.

9 Pinout

The VMS115 is packaged in a 100 MQFP package.

VMS115 100 MQFP Pinout

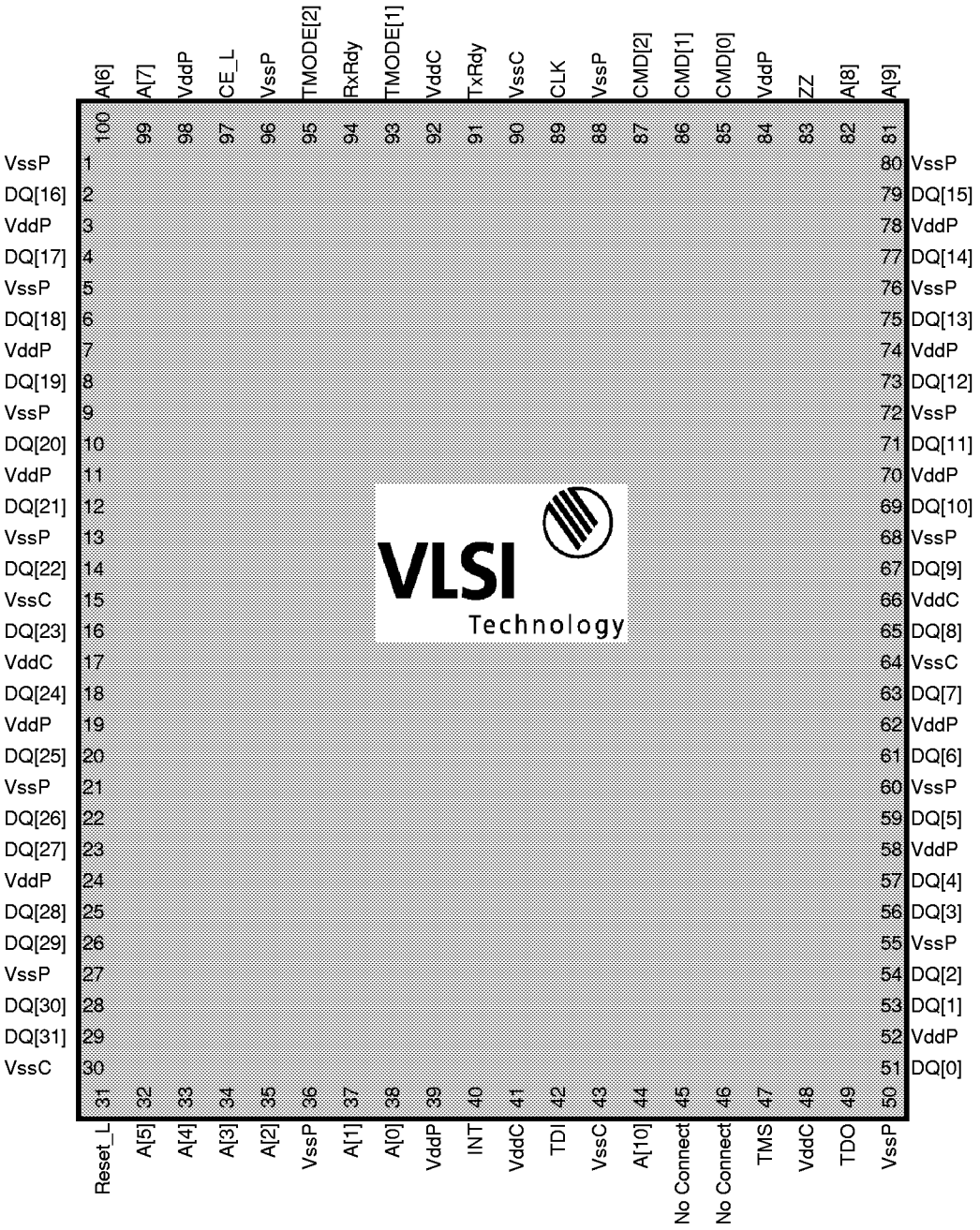
Pin Name	Pin #	Type	Description
System Interface			
CLK	89	IT	System Clock: This input is the master clock input used to derive all internal clock signals.
RESET_L	31	ITS	Reset (active low): This input is the global reset signal that will trigger and entire chip reset. This signal can be asserted asynchronous to CLK. The signal must be asserted for one CLK period to ensure exit of the reset state.
INT	40	OC	Interrupt (active high): Indicates that any enabled interrupt is pending.
ZZ	83	IT	Sleep: This input is used to put the part in a sleep mode. In sleep mode the chip operates in a reduced power consumption mode.
SSRAM Interface			
CE_L	97	IT	Chip Enable (active low): Must be asserted for all data transfers to the VMS115 chip. The VMS115 will ignore all bus activity when CE_L is not asserted.
CMD[2:0]	87,86,85	IT	Command: These inputs are used to communicate the command to the chip. 00x - Read register space 010 - Read Packet Data. 011 - Read context. 10x - Write register space 110 - Write packet data. 111 - Write context
A[10:0]	44,81,82, 99,100, 32,33,34, 35,37,38	IT	Address: Register address of memory location that is written or read.
DQ[31:0]	29,28,26, 25,23,22, 20,18,16, 14,12,10, 8,6,4,2, 79,77,75, 73,71,69, 67,65,63, 61,59,57, 56,54,53, 51	IT/OC	Data In/Out: Bi directional data bus for moving data in/out of the VMS115 chip.

VMS115 100 MQFP Pinout

IPSEC Processor Interface			
ORDY	91	OC	Transmit Ready: This output is used by the VMS115 to notify the external controller that the VMS115 is ready to transmit data.
IRDY	94	OC	Receive Ready: This output is used by the VMS115 to notify the external controller that the VMS115 is ready to receive data.
SCAN Interface			
TDI	42	IT	Test Data In: Test data input used to perform internal scan testing. This input should be driven low when not used.
TDO	49	OC	Test Data Out: Test data output used to perform internal scan testing. This output should be driven low when not used.
TMS	47	IT	Test Mode Select: Test mode select used to perform internal scan testing. This input should be driven low when not used.
TMODE[2:1]	95,93		Test Mode Pins: Reserved
Power/Ground/No Connect Inputs			
VDDC[4:0]	17,41,48, 66,92		Core Power
VSSC[4:0]	15,30,43, 64,90		Core Ground
VDDP[13:0]	3,7,11,19, 24,39,52, 58,62,70, 74,78,84, 98		Pad Power
VSSP[15:0]	1,5,9,13, 21,27,36, 50,55,60, 68,72,76, 80,88,96		Pad Ground
No Connects	45,46		Attach Pins to VSS

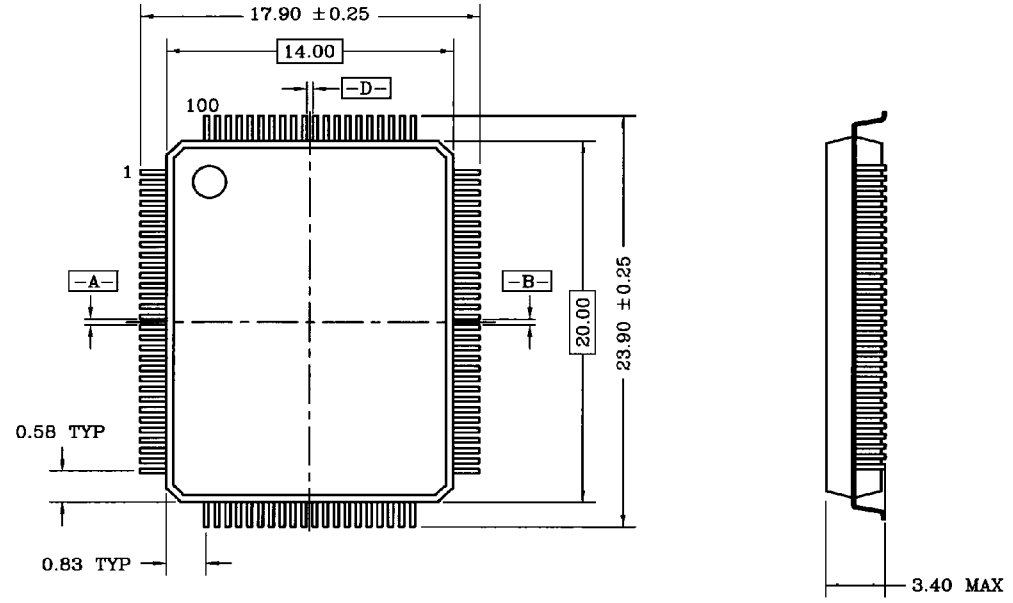
IT = Input TTL
ITS = Input TTL/Schmitt
IC = Input CMOS
IS = Input Schmitt
OC = Output CMOS
IT/OC = Input TLL and Output CMOS

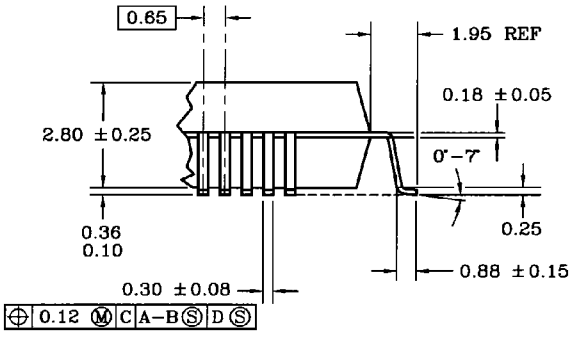
9.1 Physical Pinout Description



10 Mechanical Drawing

VLSI TECHNOLOGY, INC.





DETAIL -A-

NOTE:
1. DIMENSIONS ARE IN MILLIMETERS.
2. LEADFRAME MATERIAL: COPPER.
3. LEAD FINISH: SOLDER PLATE.

100 METRIC QUAD FLAT PACK
DWG: 25-90002 REV: •K