MCS1000

Security Processor

Features

- ARM926EJ-S (Version 3), 32-bit RISC core @ 200MHz
- 32KB instruction cache / 32KB data cache
- 32KB internal Boot ROM
- Operates in Stand-Alone or Peripheral Modes
- Three 10/100Mbps Ethernet interfaces with integrated PHY's
- Auto MDIX on all three Ethernet interfaces
- Optional MII interface for Port-C
- 32-bit, 33MHz PCI support
- Local Bus for system expansion
- Memory Controller supports SDRAM / Flash
- 16 GPIO's
- Serial Port for debug terminal
- Serial EEPROM Interface
- JTAG for Boundary-Scan / Debug
- High integration level for cost effective designs
- Hardware Cryptography Block supporting
  - Lookup Unit for search acceleration
  - 16KB Packet Cache
  - Standards: SHA-1, SHA-256, MD5, DES, 3DES and AES
  - Real-Time encryption / decryption
  - Transport and Tunnel modes
  - Random Number Generator based on physical events

Applications

- VPN Firewall Appliance
- Secure Network Interface Card: Performs all IPSec functionality, and offloads math intensive IPSec cryptographic processing from the host processor avoiding degradation of system performance.

Evaluation Boards

- MCS1000-EVB-6port VPN
- MCS1000-EVB-3 Port VPN
- MCS1000-EVB-3 Port NIC
- MCS1000-EVB-1 Port NIC

Application Notes

- MCS1000-AN-VPN
- MCS1000-AN-Secure-NIC

Ordering Information		
Commercial Grade (0 °C to +70 °C)		
MCS1000CB	272-BGA	RoHS

General Description

The MosChip Security Processor™ (MCS1000) is a highly integrated System-On-a-Chip with three 10/100 Ethernet interfaces. The MCS1000 consists of three functional groups: a CPU Subsystem, Hardware IPSec Module, and General Interface Blocks.

The CPU Subsystem contains an ARM926EJ-S, 200MHz, 32-bit RISC microprocessor. The processor handles all housekeeping functions on the MCS1000 and it is tightly integrated to the Hardware IPSec Module (cryptographic co-processor).

The Hardware IPSec Module handles all IPSec functions. It has been designed around the IPSec specification and implements DES, 3DES, AES, MD5, and SHA algorithms.

The General Interface Blocks consists of a Memory Controller, PCI Controller, Ethernet Interface, and General Purpose I/O interface.

MosChip Semiconductor ♦ 3335 Kifer Rd, Santa Clara, CA 95051 ♦ Tel (408) 737-7141 ♦ Fax (408) 737-7708

The Memory Controller can interface to standard SDRAM and Flash memory. The SDRAM controller supports up to 64 Mega Bytes of SDRAM. The non-volatile memory controller can interface to 8-, 16- or 32-bit wide devices and has 26 address lines for up to 256 Mega Bytes of storage. The Memory Controller has a Local Bus interface that supports 8-, 16- and 32-bit devices.

The PCI interface is a 32-bit 33MHz interface and it complies with the PCI 2.0 standard. The internal PCI controller acts as a slave device and can master the bus, but it does not contain a PCI arbiter so it cannot be a host PCI device in a system.

There are three (3) 10/100 Ethernet controllers integrated into the MCS1000. Each controller consists of an independent Media Access Controller (MAC) and Physical Layer (PHY). The Ethernet interfaces can be connected directly to external magnetics, LEDs and connectors.

The General I/O interface has 16 bits of GPIO, a high-speed UART and a JTAG interface. The GPIO is controlled by the core processor and can be used for any general bit manipulation and control.

The serial interface is typically configured to act as the System Console, but it can also be used as a standard RS-232 port supporting BAUD rates up to 115200 bps.

The JTAG port is reserved for test and debug only.

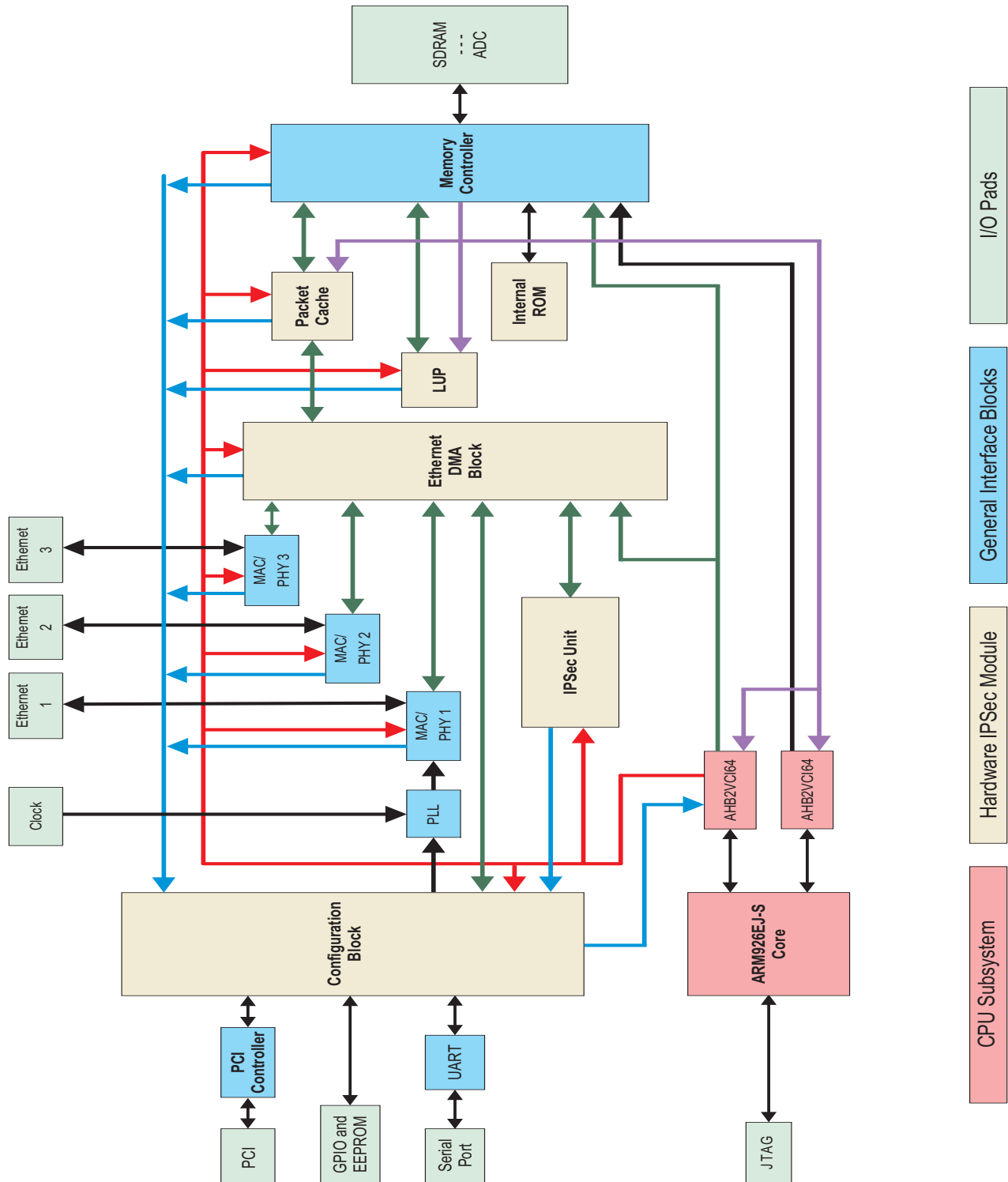**Software and Operating System Support**
The MCS1000 is based on an ARM926EJ-S processor core, and supports any OS that will run on this CPU platform. This processor core has a Memory Management Unit (MMU) and supports operating systems that require this unit in addition to most popular OS's / RTOS's such as Windows-CE, VxWorks from Wind River, Linux, etc.

MosChip Evaluation Board Configurations
MosChip has designed evaluation boards to demonstrate both Stand-Alone and Peripheral Mode configurations.

ARM is a registered trademark of ARM Limited.
All other brands or product names are the property of their respective holders.

MOSCHIP
SEMICONDUCTOR

Block Diagram

SDRAM
- - -
ADC

Memory Controller

Packet Cache

Internal ROM

LUP

Ethernet DMA Block

MAC/ PHY 3

MAC/ PHY 2

MAC/ PHY 1

IPSec Unit

PLL

AHB2VCI64

AHB2VCI64

Ethernet 3

Ethernet 2

Ethernet 1

Clock

Configuration Block

ARM926EJ-S Core

PCI Controller

GPIO and EEPROM

UART

PCI

Serial Port

JTAG

I/O Pads

General Interface Blocks

Hardware IPSec Module

CPU Subsystem

MCS1000

Security Processor

Pin List by Group

Pin Naming Convention
- "N" suffix is an active low signal
- "[x]" where x is the bit in a series of 2 or more bits

GPIO Pin List			
Pin Name / *Alt. Func.*	**Location**	**I/O**	**Description**
GPIO[0]	M17	I/O	General Purpose IO line 0
GPIO[1]	N20	I/O	General Purpose IO line 1
GPIO[2]	N19	I/O	General Purpose IO line 2
GPIO[3]	N18	I/O	General Purpose IO line 3
GPIO[4]	P20	I/O	General Purpose IO line 4
GPIO[5]	P19	I/O	General Purpose IO line 5
GPIO[6]	R20	I/O	General Purpose IO line 6
GPIO[7]	R19	I/O	General Purpose IO line 7
GPIO[8] / *MII_RX_ERR*	P17	I/O	General Purpose IO line 8
GPIO[9] / *MII_RX_DATA[0]*	R18	I/O	General Purpose IO line 9
GPIO[10] / *MII_RX_DATA[1]*	T20	I/O	General Purpose IO line 10
GPIO[11] / *MII_RX_DATA[2]*	T19	I/O	General Purpose IO line 11
GPIO[12] / *MII_RX_DATA[3]*	T18	I/O	General Purpose IO line 12
GPIO[13] / *MII_COL*	U20	I/O	General Purpose IO line 13
GPIO[14] / *MII_CRS*	V20	I/O	General Purpose IO line 14
GPIO[15] / *MII_MDIO*	T17	I/O	General Purpose IO line 15

Note: Alt. Func. Is the Alternate Ethernet PHY function in MII-Mode.

Clock Interface			
Pin Name	**Location**	**I/O**	**Description**
ETH_X1	M4	I	25MHz oscillator input / crystal input. If a clock oscillator is used, then this pin is used.
ETH_X2	N1	O	25MHz crystal output. If a clock oscillator is used, then this pin is unconnected.

Serial EEPROM Interface			
Pin Name	**Location**	**I/O**	**Description**
EEPIO[0]	R2	I/O	EEPROM I/O, programmable, SK
EEPIO[1]	T1	I/O	EEPROM I/O, programmable, SI
EEPIO[2]	P4	I/O	EEPROM I/O, programmable, SO
EEPIO[3]	R3	I/O	EEPROM I/O, programmable, CS#

JTAG and Reset Interface

Pin Name	Location	I/O	Description
POR_N	M19	I	System Reset. Minimum duration is 10 mS.
RST_OUT_N	M18	O	Reset out for other components on board. Same as System Reset re-driven. Delay from POR_N is <1 nS.
TCK	V4	I	JTAG chain clock
TDI	W4	I	JTAG chain input
TDO	Y2	O	JTAG chain output
TMS	U5	I	JTAG chain TMS

Serial Port Interface

Pin Name / *Alt. Func.*	Location	I/O	Description
CTS_N *MII_TX_CLK*	W1	I	Clear To Send. The remote device is ready to receive input.
DCD_N *MII_RX_CLK*	W3	I	Data Carrier Detect
DSR_N	W2	I	Data Set Ready. The modem is ready to answer or originate a call.
DTR_N *MII_TX_ERR*	Y1	O	Data Terminal Ready. The MCS1000 uses this to enable the modem: the modem is allowed to go on-line or answer a call.
RI_N *MII_RX_VAL*	U3	I	Ring Indicator.
RTS_N	V3	O	Request To Send. The MCS1000 is ready to exchange data.
RXD	T4	I	Receive data line.
TXD	T2	O	Transmit data line.

Note: Alt. Func. Is the Alternate Ethernet PHY function in MII-Mode.

Configuration Pin List

Pin Name	Location	I/O	Description
CONFIG_EN	U1	I	Enable alternate configuration
CONFIG_MOD[0]	T3	I	Configuration Mode bit-0
CONFIG_MOD[1]	U2	I	Configuration Mode bit-1
CONFIG_MOD[2]	V1	I	Configuration Mode bit-2

No Connect Pin List

Pin Name	Location	I/O	Description
NC	C3	NA	No Connect
NC	A20	NA	No Connect
NC	Y20	NA	No Connect

PHY Interface			
Pin Name / *Alt. Func.*	**Location**	**I/O**	**Description**
ETH_A_5K	M3	I	Band-gap reference voltage Must be tied to ground with 4.99K +- 1% resistor.
ETH_A_LED[0] *NC*	A2	O	Ethernet A LED TX/RX
ETH_A_LED[1] *NC*	B2	O	Ethernet A LED LINK 100
ETH_A_LED[2] *NC*	B3	O	Ethernet A LED LINK 10
ETH_A_RXN	F1	I	10/100 receive data channel A Negative (-).
ETH_A_RXP	F2	I	10/100 receive data channel A Positive (+).
ETH_A_SDN	B1	I	Signal Detect Negative - FX mode only
ETH_A_SDP	E4	I	Signal Detect Positive - FX mode only
ETH_A_TXN	E1	O	10/100 transmit data channel A Negative (-)
ETH_A_TXP	E2	O	10/100 transmit data channel A Positive (+)
ETH_B_LED[0]	G2	O	Ethernet B LED TX/RX
ETH_B_LED[1]	G1	O	Ethernet B LED LINK 100
ETH_B_LED[2]	H3	O	Ethernet B LED LINK10
ETH_B_RXN	L2	I	10/100 receive data channel B Negative (-).
ETH_B_RXP	L1	I	10/100 receive data channel B Positive (+).
ETH_B_TXN	K2	O	10/100 transmit data channel B Negative (-)
ETH_B_TXP	J1	O	10/100 transmit data channel B Positive (+)
ETH_C_LED[0]	J3	O	Ethernet C LED TX/RX
ETH_C_LED[1]	J4	O	Ethernet C LED LINK 100
ETH_C_LED[2]	H1	O	Ethernet C LED LINK 10
ETH_C_RXN	N2	I	10/100 receive data channel C Negative (-).
ETH_C_RXP	N3	I	10/100 receive data channel C Positive (+).
ETH_C_TXN	R1	O	10/100 transmit data channel C Negative (-)
ETH_C_TXP	P3	O	10/100 transmit data channel C Positive (+)

Note: Alt. Func. Is the Alternate Ethernet PHY function in MII-Mode.

Memory Controller: SDRAM, Flash and Local			
Pin Name	**Location**	**I/O**	**Description**
ADC_CLK	L19	O	Local Bus Clock
ADC_CS_N[0]	J18	O	Local Bus chip select 0, Active low.
ADC_CS_N[1]	J19	O	Local Bus chip select 1, Active low.
ADC_CS_N[2]	J20	O	Local Bus chip select 2, Active low.
ADC_CS_N[3]	K19	O	Local Bus chip select 3, Active low.
ADC_RD_N	L20	O	Local Bus Read
ADC_RDY	L18	I	Local Bus I/O Ready
ADC_WR_N	K20	O	Local Bus Write, Active low
SDRAM_ADDR[0]	A18	O	Address bit 0 for SDRAM interface and Local Bus
SDRAM_ADDR[1]	E20	O	Address bit 1 for SDRAM interface and Local Bus
SDRAM_ADDR[2]	G17	O	Address bit 2 for SDRAM interface and Local Bus
SDRAM_ADDR[3]	D20	O	Address bit 3 for SDRAM interface and Local Bus
SDRAM_ADDR[4]	E18	O	Address bit 4 for SDRAM interface and Local Bus
SDRAM_ADDR[5]	E19	O	Address bit 5 for SDRAM interface and Local Bus
SDRAM_ADDR[6]	D19	O	Address bit 6 for SDRAM interface and Local Bus
SDRAM_ADDR[7]	C20	O	Address bit 7 for SDRAM interface and Local Bus
SDRAM_ADDR[8]	D18	O	Address bit 8 for SDRAM interface and Local Bus
SDRAM_ADDR[9]	C19	O	Address bit 9 for SDRAM interface and Local Bus
SDRAM_ADDR[10]	B18	O	Address bit 10 for SDRAM interface and Local Bus
SDRAM_ADDR[11]	C18	O	Address bit 11 for SDRAM interface and Local Bus
SDRAM_ADDR[12]	C17	O	Address bit 12 for SDRAM interface and Local Bus
SDRAM_ADDR[13]	D16	O	Address bit 13 for SDRAM interface and Local Bus
SDRAM_ADDR[14]	B17	O	Address bit 14 for SDRAM interface and Local Bus
SDRAM_ADDR[15]	F18	O	Address bit 15 for SDRAM interface and Local Bus
SDRAM_ADDR[16]	E17	O	Address bit 16 for SDRAM interface and Local Bus
SDRAM_ADDR[17]	B20	O	Address bit 24 for SDRAM interface and Local Bus
SDRAM_ADDR[18]	F19	O	Address bit 18 for SDRAM interface and Local Bus
SDRAM_ADDR[19]	G18	O	Address bit 19 for SDRAM interface and Local Bus
SDRAM_ADDR[20]	G19	O	Address bit 20 for SDRAM interface and Local Bus
SDRAM_ADDR[21]	G20	O	Address bit 21 for SDRAM interface and Local Bus
SDRAM_ADDR[22]	H18	O	Address bit 22 for SDRAM interface and Local Bus
SDRAM_ADDR[23]	H19	O	Address bit 23 for SDRAM interface and Local Bus
SDRAM_ADDR[24]	H20	O	Address bit 24 for SDRAM interface and Local Bus
SDRAM_ADDR[25]	J17	O	Address bit 25 for SDRAM interface and Local Bus
SDRAM_CAS_N	B15	O	CAS for SDRAM interface
SDRAM_CK	A14	O	SDRAM clock
SDRAM_CS_N[0]	A16	O	Device select signal 0 for SDRAM memory chips
SDRAM_CS_N[1]	B16	O	Device select signal 1 for SDRAM memory chips
SDRAM_CS_N[2]	A17	O	Device select signal 2 for SDRAM memory chips
SDRAM_CS_N[3]	C16	O	Device select signal 3 for SDRAM memory chips
SDRAM_DATA[0]	C4	I/O	Data bit 0 for SDRAM interface and Local Bus.
SDRAM_DATA[1]	D5	I/O	Data bit 1 for SDRAM interface and Local Bus.

Memory Controller: SDRAM, Flash and Local			
Pin Name	**Location**	**I/O**	**Description**
SDRAM_DATA[2]	A3	I/O	Data bit 2 for SDRAM interface and Local Bus.
SDRAM_DATA[3]	C6	I/O	Data bit 3 for SDRAM interface and Local Bus.
SDRAM_DATA[4]	C7	I/O	Data bit 4 for SDRAM interface and Local Bus.
SDRAM_DATA[5]	B6	I/O	Data bit 5 for SDRAM interface and Local Bus.
SDRAM_DATA[6]	A6	I/O	Data bit 6 for SDRAM interface and Local Bus.
SDRAM_DATA[7]	C8	I/O	Data bit 7 for SDRAM interface and Local Bus.
SDRAM_DATA[8]	B8	I/O	Data bit 8 for SDRAM interface and Local Bus.
SDRAM_DATA[9]	B7	I/O	Data bit 9 for SDRAM interface and Local Bus.
SDRAM_DATA[10]	A5	I/O	Data bit 10 for SDRAM interface and Local Bus.
SDRAM_DATA[11]	D7	I/O	Data bit 11 for SDRAM interface and Local Bus.
SDRAM_DATA[12]	B5	I/O	Data bit 12 for SDRAM interface and Local Bus.
SDRAM_DATA[13]	A4	I/O	Data bit 13 for SDRAM interface and Local Bus.
SDRAM_DATA[14]	C5	I/O	Data bit 14 for SDRAM interface and Local Bus.
SDRAM_DATA[15]	B4	I/O	Data bit 15 for SDRAM interface and Local Bus.
SDRAM_DATA[16]	A8	I/O	Data bit 16 for SDRAM interface and Local Bus.
SDRAM_DATA[17]	C9	I/O	Data bit 17 for SDRAM interface and Local Bus.
SDRAM_DATA[18]	A9	I/O	Data bit 18 for SDRAM interface and Local Bus.
SDRAM_DATA[19]	B11	I/O	Data bit 19 for SDRAM interface and Local Bus.
SDRAM_DATA[20]	A11	I/O	Data bit 20 for SDRAM interface and Local Bus.
SDRAM_DATA[21]	D12	I/O	Data bit 21 for SDRAM interface and Local Bus.
SDRAM_DATA[22]	B12	I/O	Data bit 22 for SDRAM interface and Local Bus.
SDRAM_DATA[23]	C13	I/O	Data bit 23 for SDRAM interface and Local Bus.
SDRAM_DATA[24]	A13	I/O	Data bit 24 for SDRAM interface and Local Bus.
SDRAM_DATA[25]	C11	I/O	Data bit 25 for SDRAM interface and Local Bus.
SDRAM_DATA[26]	B13	I/O	Data bit 26 for SDRAM interface and Local Bus.
SDRAM_DATA[27]	A12	I/O	Data bit 27 for SDRAM interface and Local Bus.
SDRAM_DATA[28]	C12	I/O	Data bit 28 for SDRAM interface and Local Bus.
SDRAM_DATA[29]	B10	I/O	Data bit 29 for SDRAM interface and Local Bus.
SDRAM_DATA[30]	A10	I/O	Data bit 30 for SDRAM interface and Local Bus.
SDRAM_DATA[31]	D9	I/O	Data bit 31 for SDRAM interface and Local Bus.
SDRAM_DQM_N[0]	D10	O	Memory Byte enable signal 0 – DQM for SDRAM
SDRAM_DQM_N[1]	C10	O	Memory Byte enable signal 1 – DQM for SDRAM
SDRAM_DQM_N[2]	D14	O	Memory Byte enable signal 2 – DQM for SDRAM
SDRAM_DQM_N[3]	C15	O	Memory Byte enable signal 3 – DQM for SDRAM
SDRAM_RAS_N	C14	O	RAS for SDRAM interface
SDRAM_WE_N	B14	O	Write Enable for SDRAM Interface

PCI Interface			
Pin Name	Location	I/O	Description
PCI_AD[0]	U18	I/O	PCI Address/Data bit 0
PCI_AD[1]	U19	I/O	PCI Address/Data bit 1
PCI_AD[2]	V19	I/O	PCI Address/Data bit 2
PCI_AD[3]	W20	I/O	PCI Address/Data bit 3
PCI_AD[4]	W19	I/O	PCI Address/Data bit 4
PCI_AD[5]	V18	I/O	PCI Address/Data bit 5
PCI_AD[6]	Y19	I/O	PCI Address/Data bit 6
PCI_AD[7]	W18	I/O	PCI Address/Data bit 7
PCI_AD[8]	V17	I/O	PCI Address/Data bit 8
PCI_AD[9]	U16	I/O	PCI Address/Data bit 9
PCI_AD[10]	W17	I/O	PCI Address/Data bit 10
PCI_AD[11]	V16	I/O	PCI Address/Data bit 11
PCI_AD[12]	Y17	I/O	PCI Address/Data bit 12
PCI_AD[13]	W16	I/O	PCI Address/Data bit 13
PCI_AD[14]	V15	I/O	PCI Address/Data bit 14
PCI_AD[15]	Y16	I/O	PCI Address/Data bit 15
PCI_AD[16]	V11	I/O	PCI Address/Data bit 16
PCI_AD[17]	W11	I/O	PCI Address/Data bit 17
PCI_AD[18]	Y11	I/O	PCI Address/Data bit 18
PCI_AD[19]	V10	I/O	PCI Address/Data bit 19
PCI_AD[20]	W10	I/O	PCI Address/Data bit 20
PCI_AD[21]	Y9	I/O	PCI Address/Data bit 21
PCI_AD[22]	W9	I/O	PCI Address/Data bit 22
PCI_AD[23]	V9	I/O	PCI Address/Data bit 23
PCI_AD[24]	Y8	I/O	PCI Address/Data bit 24
PCI_AD[25]	W8	I/O	PCI Address/Data bit 25
PCI_AD[26]	Y7	I/O	PCI Address/Data bit 26
PCI_AD[27]	W7	I/O	PCI Address/Data bit 27
PCI_AD[28]	Y6	I/O	PCI Address/Data bit 28
PCI_AD[29]	W6	I/O	PCI Address/Data bit 29
PCI_AD[30]	U7	I/O	PCI Address/Data bit 30
PCI_AD[31]	V6	I/O	PCI Address/Data bit 31
PCI_CBE_N[0]	Y18	I/O	Command/byte enable bit 0
PCI_CBE_N[1]	W15	I/O	Command/byte enable bit 1
PCI_CBE_N[2]	U11	I/O	Command/byte enable bit 2
PCI_CBE_N[3]	Y5	I/O	Command/byte enable bit 3
PCI_CLK	Y3	I	PCI clock
PCI_DEVSEL_N	W13	I	PCI Device Select

MCS1000

Security Processor

PCI Interface			
Pin Name	**Location**	**I/O**	**Description**
PCI_FRAME_N	Y12	I	FRAME_N is asserted to indicate the start and duration of a transaction. It is de-asserted on the final data phase.
PCI_GNT_N	V7	I	PCI Grant
PCI_IDSEL_N	U9	I	PCI configuration block ID select
PCI_INTA_N	W5	O	PCI interrupt A
PCI_IRDY_N	W12	I/O	Initiator read is driven by the master to indicate valid data on a write transaction, or that it is ready to receive data on a read transaction.
PCI_LOCK_N	W14	I/O	PCI Lock
PCI_PAR	V14	I/O	PCI Parity signal
PCI_PERR_N	Y14	I/O	PCI Parity error signal
PCI_REQ_N	V5	I/O	PCI Request
PCI_RST_N	Y4	I	PCI Reset
PCI_SERR_N	V13	I/O	PCI error signal
PCI_STOP_N	Y15	I/O	PCI Stop
PCI_TRDY_N	Y13	I/O	PCI target ready

Rev.   1.1

Power Pin List			
Pin Name	**Location**	**I/O/P**	**Description**
AGND_DIGA	G4	P	Analog Ground
AGND_DIGA	G3	P	Analog Ground
AGND_DIGB	K1	P	Analog Ground
AGND_DIGC	P1	P	Analog Ground
AGND_PLLG	M1	P	Analog Ground
AGND_REFA	F3	P	Analog Ground
AGND_REFB	K3	P	Analog Ground
AGND_REFC	P2	P	Analog Ground
AGND_RXA	D2	P	Analog Ground
AGND_TXA	D1	P	Analog Ground
AGND_TXB	H2	P	Analog Ground
AGND_TXG	J2	P	Analog Ground
AVCC_PHYA	C2	P	Analog Vcc
AVCC_PLLG	L4	P	Analog Vcc
AVCC_RXA	D3	P	Analog Vcc
AVCC_TXA	C1	P	Analog Vcc
AVCC_TXG	M2	P	Analog Vcc
CPU_PLL_GND	K17	P	Ground for CPU PLL
CPU_PLL_VDD	K18	P	Vcc for CPU PLL
Ground	A1	P	Ground for VCC and VCC_CORE
Ground	D4	P	Ground for VCC and VCC_CORE
Ground	D8	P	Ground for VCC and VCC_CORE
Ground	D13	P	Ground for VCC and VCC_CORE
Ground	D17	P	Ground for VCC and VCC_CORE
Ground	H4	P	Ground for VCC and VCC_CORE
Ground	H17	P	Ground for VCC and VCC_CORE
Ground	N4	P	Ground for VCC and VCC_CORE
Ground	N17	P	Ground for VCC and VCC_CORE
Ground	U4	P	Ground for VCC and VCC_CORE
Ground	U8	P	Ground for VCC and VCC_CORE
Ground	U13	P	Ground for VCC and VCC_CORE
Ground	U17	P	Ground for VCC and VCC_CORE
SYS_PLL_GND	V12	P	Ground for System PLL
SYS_PLL_VDD	U12	P	Vcc for System PLL
Thermal Ground	J9	P	Thermal Conductive Ball (Ground)
Thermal Ground	J10	P	Thermal Conductive Ball (Ground)
Thermal Ground	J11	P	Thermal Conductive Ball (Ground)
Thermal Ground	J12	P	Thermal Conductive Ball (Ground)

Power Pin List			
Pin Name	Location	I/O/P	Description
Thermal Ground	K9	P	Thermal Conductive Ball (Ground)
Thermal Ground	K10	P	Thermal Conductive Ball (Ground)
Thermal Ground	K11	P	Thermal Conductive Ball (Ground)
Thermal Ground	K12	P	Thermal Conductive Ball (Ground)
Thermal Ground	L9	P	Thermal Conductive Ball (Ground)
Thermal Ground	L10	P	Thermal Conductive Ball (Ground)
Thermal Ground	L11	P	Thermal Conductive Ball (Ground)
Thermal Ground	L12	P	Thermal Conductive Ball (Ground)
Thermal Ground	M9	P	Thermal Conductive Ball (Ground)
Thermal Ground	M10	P	Thermal Conductive Ball (Ground)
Thermal Ground	M11	P	Thermal Conductive Ball (Ground)
Thermal Ground	M12	P	Thermal Conductive Ball (Ground)
VCC	D6	P	3.3 Volt I/O Power Ball
VCC	D11	P	3.3 Volt I/O Power Ball
VCC	D15	P	3.3 Volt I/O Power Ball
VCC	F4	P	3.3 Volt I/O Power Ball
VCC	F17	P	3.3 Volt I/O Power Ball
VCC	K4	P	3.3 Volt I/O Power Ball
VCC	L17	P	3.3 Volt I/O Power Ball
VCC	R4	P	3.3 Volt I/O Power Ball
VCC	R17	P	3.3 Volt I/O Power Ball
VCC	U6	P	3.3 Volt I/O Power Ball
VCC	U10	P	3.3 Volt I/O Power Ball
VCC	U15	P	3.3 Volt I/O Power Ball
VCC_CORE	E3	P	Core Vcc
VCC_CORE	L3	P	Core Vcc
VCC_CORE	V2	P	Core Vcc
VCC_CORE	V8	P	Core Vcc
VCC_CORE	Y10	P	Core Vcc
VCC_CORE	U14	P	Core Vcc
VCC_CORE	P18	P	Core Vcc
VCC_CORE	M20	P	Core Vcc
VCC_CORE	F20	P	Core Vcc
VCC_CORE	B19	P	Core Vcc
VCC_CORE	A19	P	Core Vcc
VCC_CORE	A15	P	Core Vcc
VCC_CORE	B9	P	Core Vcc
VCC_CORE	A7	P	Core Vcc

Theory of Operation

The MCS1000 has two modes of operation:

Stand-Alone Mode

In Stand-Alone mode the MCS1000 operates autonomously and does not require a host to supervise its operation. The internal ARM processor fetches instructions from the external Flash. In this mode the PCI bus is inactive. This configuration is ideal for devices like low-cost VPN/Firewall appliances.

Peripheral Mode

In Peripheral mode the MCS1000 functions as a secure Fast Ethernet controller. The MCS1000 acts as a device on a PCI bus. The control log is stored at the host level, but the Security Policy Database (SPD) and the Security Association Database (SAD) is stored in the MCS1000's local memory. The control log and Internet Key Exchange are left to the host processor.

Internal Architecture

The MCS1000 consists of a CPU Subsystem, Hardware IPSec Module and General Interface Blocks.

The CPU subsystem consists of the ARM926EJ-S core and two (2) AHB to VCI bridges.

The MCS1000 contains specific hardware for accelerating the IPSec protocol. The Hardware IPSec Module consists of an IPSec Unit that contains the DES, 3DES, and AES cryptography blocks, plus the MD5, SHA1 and SHA256 authentication blocks; a 16K Packet Cache; Look-Up Accelerator (LUP); Random Number Generator; Ethernet DMA Block; Configuration Block and specialized code in an internal ROM.

The rest of the chip's logic is made up from General Interface Blocks. The General Interface Blocks consist of three (3) Ethernet Controllers, PCI Interface, Memory Controller (SDRAM, Flash and Local-Bus), Serial Port, General Purpose Input and Output (GPIO) Interface, and PLL Clocking Circuit.

Description of Blocks:

CPU Subsystem

- **ARM926EJ-S** - ARM9 32-bit RISC CPU
- **AHB2VCI64** - AHB 32-bit Bus to VCI 64-bit Bus (2)

Hardware IPSec Module

- **Configuration Block** - configuration registers and miscellaneous system interfaces
- **Ethernet DMA Block** - DMA channels and access arbiter for MAC, PCI and Cipher Module
- **LUP** - Table lookup engine for search acceleration
- **Random Number Generator** - Random number generator for key generation based on the physical events inside the IC
- **Packet Cache** - Ethernet packet cache (write-back, write-allocate) with large line size (16 double words)
- **IPSec Unit** - Contains the DES, 3DES, and AES cryptographic blocks; MD5 and SHA authentication blocks; and I/O blocks
- **Internal ROM** - Contains BIST and initialization code

General Interface Blocks

- **PCI Controller** - 32-bit 33MHz PCI 2.2 interface (device only)
- **UART** - Standard serial port with LVCMOS logic level input/output
- **PLL** - Phase Locked Loop used to generate internal clocks
- **MAC/PHY** - Ethernet interface to external magnetics
- **Memory Controller** - Memory controller back-end for Flash and SDRAM, arbiter/mux between CPU, LUP and packet cache
- **Bootstrap Register** - The bootstraps control four main aspects of the MCS1000: Debug Mode, Processor Speed, Internal ROM Disable and User Defined functions

MCS1000

Security Processor

MOSCHIP
SEMICONDUCTOR

| CPU Subsystem | Hardware IPSec Module | General Interface Blocks | I/O Pads |
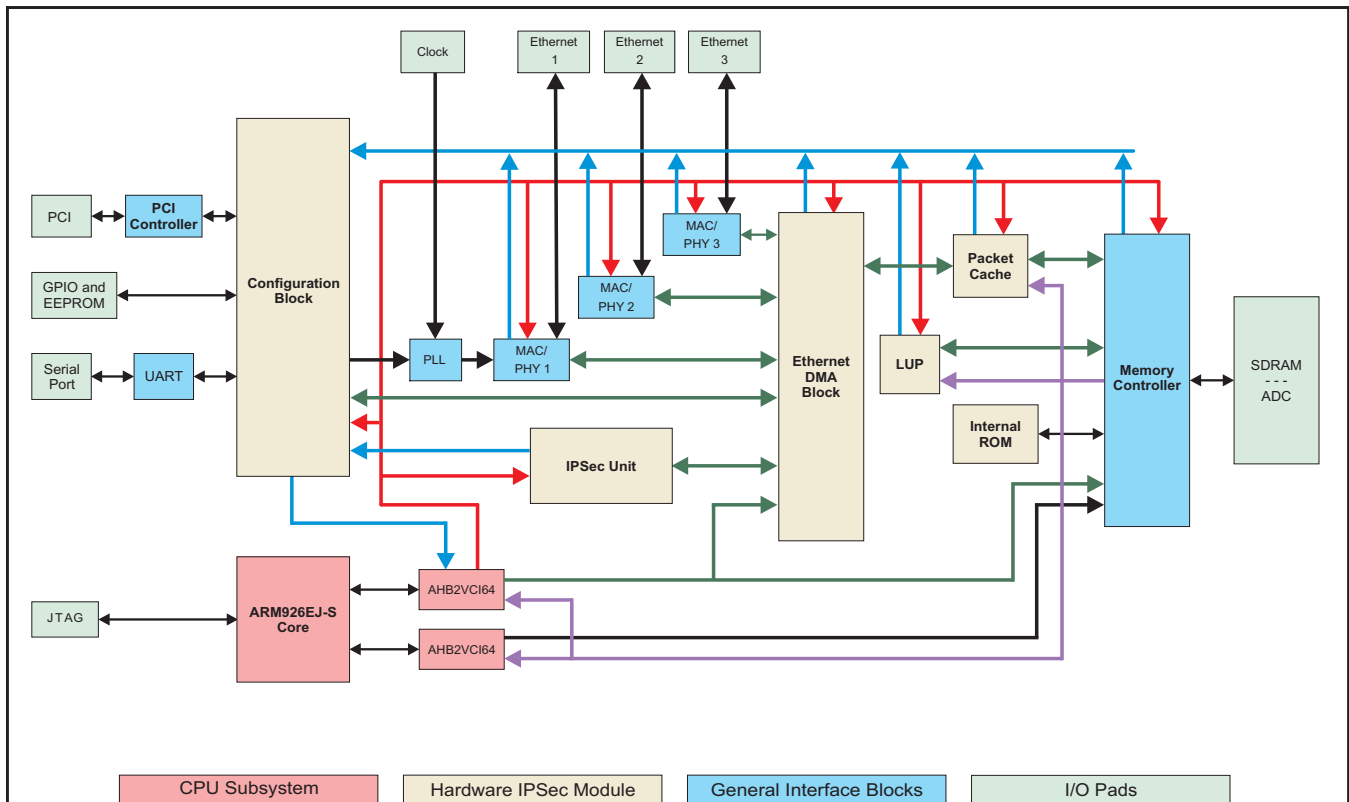
Hardware Description

The red and blue busses are the configuration buses. The write data and address busses (red) go to all devices. The return data bus (blue) is gathered into the Configuration Block where it is multiplexed according to the address. The green and violet are the data busses. The bus (black) going from the lower AHB2VCI64 to the Memory Controller is the Control/Address for the memory controller. All internal data busses are 64-bit@50MHz. The SDRAM back-end is translated from a 64-bit@50MHz to a 32-bit@100MHz interface.

All blocks inside the IPSec Unit (cryptography, authentication and I/O) run at 50MHz except for the four (4) 3DES cryptography blocks which run at 100MHz.

There are two (2) PLLs in the PLL Block. One PLL is for the CPU core and the other is for the Hardware IPSec Module and the remaining blocks. Both of the PLLs are sourced from the 25 MHz clock taken from the PHYs.

CPU Subsystem

The MCS1000 is based upon an ARM926EJ-S processor core. Additional detailed documentation on the ARM926EJ-S core can be found at the ARM Website. There are differences between the 926 implemented in the MCS1000 and the standard ARM926EJ-S.

In addition to the ARM9, the CPU Subsystem includes an ARM9 AMBA (AHB) to VCI Bus Bridge. The AHB to VCI Bridge connects the ARM926EJ-S to the IPSec hardware.

MCS1000 Implementation of the ARM926EJ-S

The MCS1000 includes all functions in the standard ARM926EJ-S except for the Embedded Trace Macrocell (ETM) port. The instruction set (ARMv5TEJ), Memory Management Unit, Caches, Jazelle™ processing engine (Java), and JTAG interface remain unchanged.

- **Memory Management Unit (MMU)** - The ARM926EJ-S has an MMU and the MCS1000 has the MMU implemented.
- **Caches (Instruction and Data)** - The ARM926EJ-S has a 32KByte Instruction cache and a 32KByte Data cache.
- **JTAG Interface** - The ARM926EJ-S has a JTAG interface, but the EXTEST feature of the JTAG is not implemented.
- **AHB2VCI64 (AHB to VCI Bus Bridge)** - The link between the ARM926 core and the additional hardware in the MCS1000 consists of two AHB to VCI Bridges. The bus interface on the ARM926EJ-S is an AMBA (AHB) interface. The IPSec Hardware utilizes a Virtual Component Interface (VCI) bus. The bridge enables communication between the ARM9 core and the rest of the system.

Hardware IPSec Module

The Hardware IPSec Module accelerates the IPSec protocol by using dedicated hardware blocks. It implements Encapsulating Security Payload (ESP) and Authentication Header (AH) IPSec protocols. The encryption and authentication algorithms that the MCS1000 uses are DES, 3DES, AES, SHA-1, SHA-256 and MD5. The MCS1000 performs DES, 3DES and AES in both Cipher Block Chaining (CBC) mode and Electronic Code Book (ECB) mode. The AES algorithm can be performed in 128-, 192-, and 256-bit modes.

- **Configuration Block** - The Configuration Block holds all of the configuration registers and miscellaneous system interfaces.
- **Ethernet DMA Block** - DMA channels and access arbiter for MAC, PCI and Cipher Module
- **Look Up Accelerator (LUP)** - Used to alleviate the CPU. If the Security Parameter is not found in the cache, then the CPU hands off to the LUP the SPI to search the SPD (stored on external SDRAM) for a match.
- **Packet Cache** - Ethernet packet cache (write-back, write-allocate) with large line size (16 double words). The Packet Cache is used to store Ethernet packets on-chip. This on-chip storage increases system performance since the packet does not need to be stored off-chip and frees memory controller for CPU access.
- **Internal ROM** - The internal 32K Byte ROM contains Built-In Self Test (BIST) and initial configuration information for the MCS1000. The ROM is used to test and initialize the ARM core and the Hardware IPSec Module. The ROM enables the CPU to operate in Stand-Alone or Peripheral mode. This ROM is factory masked and is not user programmable.
- **IPSec Unit** - The IPSec Unit contains the Operator Blocks, I/O Blocks and the DMA Channel. The Operator Blocks include the cryptography blocks (DES, 3DES, AES) and the authentication blocks (MD5, SHA-1, and SHA-256). The cryptography blocks are known as the BL operators and the authentication blocks are known at the HM operators. The I/O Blocks are the means of input and output to the IPSec Unit. The DMA Channel "links" the required blocks together in order to perform encryption/decryption and signature/authentication quickly. Collectively the cryptography blocks and authentication blocks are known as operator blocks.
- **Random Number Generator (RNG)** - The RNG is a hardware-based block that generates true random numbers.

IPSec Unit Architecture

The IPSec Unit contains the data I/O channels and the operator blocks. The operator blocks transform the data and the I/O channels serialize and de-serialize the data for the operators. The IPSec Unit communicates with the system memory through the arbiter as described in the previous section. The operators have different block widths, but all of them are multiples of a double-word (32 bits).

The channels and operators are configured via the Configuration Block. This block decouples the reads and writes for the addressed device. Each device or channel has up to 6 double-words of configuration/ status interface. To ease the addressing, the configuration areas are gathered into 8 double-word blocks and the unused areas are marked as reserved. The reserved registers are not decoded – writes are discarded and reads return random data. Detailed descriptions of the register maps for the channels and the operators are found in the data book document. The configuration is entered into the registers using CPU load/store instructions.

The Ethernet DMA Block distributes the packet data-flow through the operators and channels. The DMA router and channel arbiter turn operators on and off from the stream and control the DMA channel bus multiplexers. The input-output channels allow several streams to run through the IPSec Unit in parallel.

The operators have to start at the correct double-word location. The HASH algorithm data is read-only and is not written by the device. The operator blocks must bypass the data and start at the correct location. This is achieved by flushing the output buffers at the beginning of the operation. This also guarantees the correct block alignment. The DMA router and channel arbiter control the flow of information inside the DMA channel and serve two purposes:

- Assemble the required operators into op chains.
- Control the start/end of the operator processing inside the stream.

This simplifies the operator architecture (There are 28 operators, but only one router for all of them) and saves power by disconnecting the operator from the chain in bypass mode. The disconnected operator disconnects from the data transaction as opposed to being simply in bypass mode. The disconnect has to occur at the block aligned address for the operator, otherwise an error is flagged by the DMA controller.

The IPSec Unit within the MCS1000 consists of twenty-eight (28) blocks: four (4) Input channels, four (4) Output channels, two (2) AES encryption blocks, two (2) AES decryption blocks, four (4) DES/3DES blocks, four (4) SHA-1 authentication blocks, four (4) SHA-256 authentication blocks, and four (4) MD5 authentication blocks. The DMA channel performs the arbitration between all 28 devices.

Configuration Access Interface
This unit sends the configuration data to the correct operator or I/O channel. It generates the write pulses that store the data into the configuration registers inside the operator units. This unit converts VCI bus cycles to operator configuration signals. The configuration interface consists of 8 data transfer ports.

Input and Output Channel
These channels form the interface between the DMA channel and the system memory. Memory transfers are burst-oriented. It is possible to set the limit at which the system memory transactions start. The channels monitor the transfer count and hold it if it is less than a full burst; once the count is full the data is transferred. The maximum buffer size can be read from the channel status word and the currently programmed buffer size can be read from the channel control register. This allows for tuning the burst sequence length to the system memory. I/O channels use the basic VCI protocol to connect to memory.

The data path consists of a FIFO and a Byte aligner. The Byte aligner is always near the memory. In VCI I/O operations, the memory data-path is twice as wide as the DMA channel data-path. The data flows into the input channel and out of the output channel. The Byte aligner formats the double-word stream to be correctly aligned in memory. For this the data is rotated and multiplexed with the previous data depending on the Byte start address. The FIFO is a double-word asymmetric FIFO. The I/O channel DMA is the same for both Input and Output channels. The difference is in the start and end conditions. The configuration register file is accessed via the configuration port described above.

MCS1000

Security Processor

Operator Blocks

The operators function on the packet data stream. Two types of operators exist – HM and BL. The HM operators sign and authenticate the data and are read-only. The BL operators encrypt and decrypt the data.

Each operator connects to a DMA channel. The DMA system takes care of the data transfer and routing. The operator queries the DMA channel with rd_req and wr_req. Upon receiving rd_gnt and write_gnt the data is either available from the transmitter or captured by the receiver. Kill terminates the operator and clears the I/O queues. Run starts the operator processing.

The operators handshake for data using rq/gnt pairs. They are unaware from where the data originates or where it ends up. The signature operators do not have a transmitter and resultant signature is stored inside them for later readout by the CPU using the configuration interface.

All operators process data in blocks. The block sizes are 64 bits (2 double-words) for (3)DES, 128 bits (4 double-words) for AES ciphers and 1024 bits (16 double-words) for all signature blocks. The operators do not have start/end counters inside the blocks. The routing is handled by the DMA and the disconnected operators do not receive the rd_gnt or wr_gnt. They are disconnected from the read port on block aligned positions inside the stream.

BL Operator

The BL operator contains the cipher algorithm and the DMA channel controller. It has both outputs and inputs. The operator is only connected to the data-stream for the duration of processing operation. It only works on the data in its range. The range registers are inside the DMA router.

Configuration registers hold the Key, IV, Status and Configuration registers. It is possible to write the IV through the configuration interface or use the first block as an IV. The mode is set by the configuration register. These are accessed by the configuration mechanism described above. The CBC Control and the BL accumulator form the data-path and DMA handshake controller. They also contain the required logic to create a cipher feedback mode and power management. This block is the same for all BL channels. The cipher block size is entered as a parameter to this module.

The ECB cipher module is the encryption engine that differentiates the block operators. This is the common Electronic Code Book mode of the cipher required. This block runs asynchronously and is clock-stopped by the power management system in case it is idle. All the inputs are stable for the duration of the cipher run.

HM Operator

The HM operator contains the signature block and the DMA channel controller. The Channel controller is a subset of the BL operator. The HM operator only reads data leaving the signature inside the block after the transformation is complete. This is done because the channel has no direct access to the stream and therefore cannot install ICV value in the headers that have been bypassed. The signature value is small compared to the data count and does not create a bottleneck. Signature blocks by themselves do not allow for keying and the IPSec standard requires a keyed hash – HMAC. The data preparation requires special handling done inside the operator by the HMAC FSM. The differentiating factor is the hash algorithm used: MD5, SHA-1 and SHA-256. The hash operator can be divided into data preparation and transformation stages. The data preparation is the same for all three of the above-mentioned hash algorithms.

Power management works in the same way as with the BL operator. It is possible to stop all of the clocks to the operator or those to the hash module. See BL operator for additional description.

DMA Channel (Router and Arbiter)

The DMA channel connects the operators and the IO channels into chains. The structure of the chain will depend on the IPSec mode negotiated between the communicating parties and the available operators. It is possible to process the data sequentially with a small number of operators or to "wake up" sleeping operators in case more processing power is required. When disconnected the operators will be idle and consume very little power.

Functionally the operators do not require any information about the data stream origin or its fragmentation. When new data is received the operator processes the block and delivers the next block of data for output. The chain is a FIFO block and therefore, automatically flushes the channel as the data moves through the blocks.

IPSec Unit Operation

The operator blocks within the IPSec Unit perform the data transformations required by the IPSec standard. These are encryption, decryption and signing of the IP packet. The packet is fed into the operators from the Packet Cache by the block I/O channels. There is an arbiter within the Hardware IPSec Module that enables sharing of the system memory between competing devices and uses a round-robin based priority scheme in the arbiter to allow for evenly distributed access between the channels. The arbitration scheme will not cause the device to lock-up because the cipher block is aware of the I/O status, and will not request additional data before output is flushed. The CPU Subsystem has concurrent access to the Packet Cache and can perform packet analysis concurrently.

The operator block works on the stream(s) of data by processing it on the fly. The packet data stream is fetched from memory by Input Channel(s), processed by the operator block(s) and sent back to memory through Output Channel(s). The Input channel serializes the memory data for the operator blocks and the Output channel writes the processed stream back into the memory. Two types of operator blocks process the data stream. The Block Encryption (BL) operator encrypts or decrypts the stream data. The signature operator (HM) calculates the Integrity Check Value (ICV) but since it does not alter the data it only has read capabilities. Several operators can be connected into a chain to allow data processing in a pipeline. This mode can be used for IPSec tunneling and avoids unnecessary data transfers between the memory and cipher block. The stream connection is controlled by the router block inside the connection unit.

Consider for example the common calculation in IPSec: the AH-ESP set. This requires two signatures and one encryption. The first signature must start from the beginning. The second signature and encryption must start at the ESP header. For this to take place the operators are pre-configured with the correct set of keys and the start/end points in the data-stream.

Rev.   1.1   Page 19

MOSCHIP
SEMICONDUCTOR

IP Packet with IPSec Headers
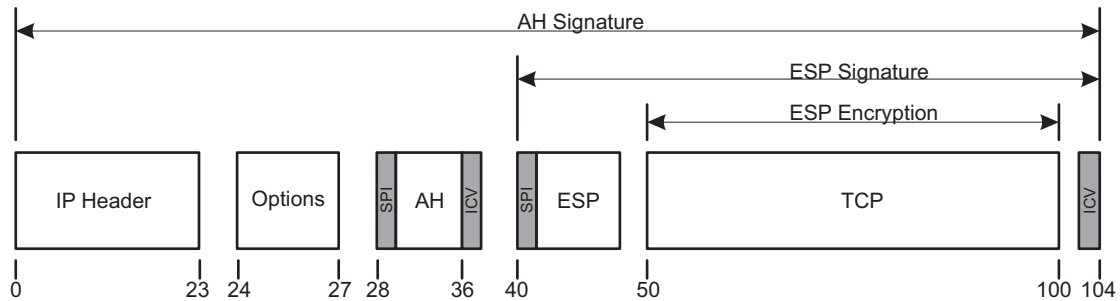
Consider the IP packet shown in the figure above. It contains both IPSec headers and offers full transport protection. The AH header signs the cleartext (open) header section up to the ESP at location 40. The ESP signs and encrypts the packet payload. Although the ESP signature in this case is redundant because it is already covered by the AH it must still be considered during the performance analysis because the IPSec standard supports it. All of the algorithms used for signature and encryption with their respective keys are negotiated a priori by a key exchange protocol (IKE) or entered manually. Both IPSec headers contain a security parameter index (SPI) that points to the description of these settings which are collectively called the security association (SA). When a packet enters the IPSec processing process the first thing to do is to look up the SA from the database, use that information to initialize the operators with their keys and program the start and end double word locations within the packets.

Assuming that the SPI lookup revealed the following SAs for the packet above:

SA1	AH:	HMAC-MD5	with 160 bit ICV		
SA2	ESP:	HMAC-SHA1	with 160 bit ICV	3DES	for encryption, IV included in packet payload.

From the packet analysis we know that the AH header starts from the beginning (0) and ends at the packet end (104). The ESP signature starts at the ESP header (40) and ends at the packet end (104). The ESP encryption starts at the ESP header after SPI (44) and ends before the ICV at the end (100). Thus we have the following ranges for the operators:

	Start	End
HM for AH	0	104
HM for ESP	40	104
BL for ESP	44	100

Rev.   1.1

As the BL in the IPSec must be in cipher feedback mode (CBC) it requires also the IV. For all currently used BL algorithms the IV is the first block in encryption data stream (It need not be so and the possibility must exist to enter the IV with the keys from SA). This information is all that is required to process the packet. The operators are programmed with keys and start/ end info. This information is written using the registers in the Configuration Block. After the operators are configured with the key data they are connected into a pipeline chain by programming the registers in the Configuration Block.

The data-stream always goes through the BL units and is only read by the HM units. The data processing goes through the following transformations:

Location	ESP BL	ESP HM	AH HM
0	Bypass	Suspend	Begin
40	Bypass	Begin	Continue
44	Begin	Continue	Continue
104	End	End	Continue
104	Suspend	Suspend	End

After the packet has passed through the operator chain the data is transformed and signatures are available within the HM units. Although this example used encryption, decryption follows the same principle but both HM units are located in front of the BL encryption block.

Hardware IPSec Module Performance

The Hardware IPSec Module performance is rated in the next section. One of the primary advantages of the MCS1000's architecture is the presence of the Packet Cache. The Packet Cache allows the packet to be inside the cache for the duration of its lifetime through the pre-process (CPU) - en/decrypt (cipher) - post-process (CPU) cycle. This frees the memory controller for an alternate program (CPU) code/data fetch and lookup unit. The IPSec Unit operator chaining feature avoids multiple memory transactions during the cipher run. The packet is transferred in a single pipelined pass and the actual throughput achieved will not be degraded because only one pass is required.

Cryptography and Authentication Block Performance

The five (5) Encapsulating Security Payload (ESP) and Authentication Header (AH) algorithms run at the rates described below:

- **DES** algorithm is 16 cycles running at 100MHz (64-bit block data) or 100/16*64 = 400Mbps with 4 blocks for a total bandwidth of 1.6Gbps.
- **3DES** algorithm is 48 cycles running at 100MHz (64-bit block data) or 100/48*64 = 128Mbps with 4 blocks for a total bandwidth of 512Mbps.
- **AES** algorithm is 50 cycles running at 50MHz (128-bit block data) or 50/50*128 = 128Mbps with 4 blocks for a total bandwidth of 512Mbps.
- **MD5 and SHA(-1)** algorithms run in 80 cycles at 50MHz (64 Bytes) or 50/80*512 = 320Mbps with 4 blocks for a total bandwidth of 1.32Gbps.

There are four (4) DES/3DES blocks, four (4) AES blocks, four (4) MD5 blocks, four (4) SHA blocks, and four (4) SHA-1 blocks.

Internal Bus Performance

The connection between the IPSec Unit and the Ethernet DMA Block is a 64-bit 50MHz interface, yielding a throughput of 3.2Gbps. This is then divided up into eight (8) I/O streams giving the MCS1000 a 400Mbps bandwidth per input or output stream.

General Interface Blocks
The General Interface Blocks consist of three Ethernet controllers, PCI interface, Memory Controller (SDRAM, Flash and Local-bus), serial port, GPIO and Clock Interface. These blocks are described briefly in this section. For a detailed description see the next section.

Ethernet Interface
The MCS1000 has three (3) Fast Ethernet controllers. These controllers consist of a MAC and a PHY. Only the PHY interfaces are brought off-chip. The PHYs can be connected directly to Ethernet magnetics.

The Ethernet interfaces consist of DMA, TLI, MAC, and PHY sub-blocks. The DMA automates memory transfers and frees the CPU from this task. The Transaction Layer Interface (TLI) is a 32-bit wide block designed to provide a bridge between the DMA controller and a 10/100 Ethernet MAC. The TLI uses two (2) 2 KByte FIFOs for receive and transmit buffering.

The MCS1000 has an alternate Ethernet configuration. In this configuration the third Ethernet interface (Ethernet-C) does not use the internal PHY, and its MII is available on the external pins. In this configuration LED signals for Ethernet-A and B, eight (8) GPIO and three (3) UART control signals are all unavailable. This configuration is controlled via pull-ups and pull-downs on the four (4) configuration pins and cannot be entered through software.

Ethernet MAC Features
The MAC core has the following features:
- IEEE 802.3, 802.3u specification compliant
- Supports 10/100 Mbps data transfer rates
- VLAN support
- Full-duplex and half-duplex support
- Supports flow-control frames in full-duplex mode
- Collision detection and auto retransmission on collisions in half-duplex mode
- CSMA/CD protocol support for half-duplex mode
- Preamble generation and removal
- Automatic 32-bit CRC generation and checking
- Options to insert PAD/CRC32 on transmit
- Insertion and stripping of padding Bytes
- Flexible address filtering modes:
  - One 48-bit perfect address
  - 64 hash-filtered multicast addresses
  - Pass all multicast addresses
- Promiscuous mode
- Pass all incoming packets with a status report
- Separate 32-bit status is returned for transmit and receive packets

Ethernet PHY Features
There are three (3) integrated physical layers (PHYs), one connected to each internal MAC. The PHYs are designed to be connected directly to Ethernet magnetics.

- 10/100BASE-TX Ethernet PHY device
- Data is reliable over cable lengths up to 100 meters
- Supports transformers with 1.41:1 turns ratio (E.g.: Tyco 1605011-1)
- Wave-shaping – no external filter required
- Full and half-duplex operation with full-featured Auto-Negotiation function
- LED indicators (3 each): Fully configurable for Link, TX activity, RX activity, Collision, 10 Mbps, 100 Mbps, Full or Half Duplex
- Media Dependent Interface / Media Dependent Interface Crossover MDI/MDIX
- Built-in Loop-Back and Test Modes

The MCS1000 has three (3) Fast Ethernet controllers. These controllers consist of a MAC and a PHY. The PHY interfaces are brought off-chip in normal mode (3 PHY mode). The PHYs can be connected directly to Ethernet magnetics. There are two configurations for the PHYs. The standard configuration brings all three PHYs to the IC's pins. In the alternate configuration two PHYs and one MII are brought out to the IC's pins. See table below for pin-out details.

Ethernet PHY Alternate Configuration Interface Pins				
Pin Name	**I/O**	**Location**	**Alternate Function**	**I/O**
ETH_A_LED[0]	O	A2	MII_TX_MDC	O
ETH_A_LED[1]	O	B2	MII_TX_EN	O
ETH_A_LED[2]	O	B3	MII_TX_DATA[0]	O
ETH_B_LED[0]	O	G2	MII_TX_DATA[1]	O
ETH_B_LED[1]	O	G1	MII_TX_DATA[2]	O
ETH_B_LED[2]	O	H3	MII_TX_DATA[3]	O
GPIO[8]	I/O	P17	MII_RX_ERR	I
GPIO[9]	I/O	R18	MII_RX_DATA[0]	I
GPIO[10]	I/O	T20	MII_RX_DATA[1]	I
GPIO[11]	I/O	T19	MII_RX_DATA[2]	I
GPIO[12]	I/O	T18	MII_RX_DATA[3]	I
GPIO[13]	I/O	U20	MII_COL	I
GPIO[14]	I/O	V20	MII_CRS	I
GPIO[15]	I/O	T17	MII_MDIO	I/O
CTS_N	I	W1	MII_TX_CLK	I
DCD_N	I	W3	MII_RX_CLK	I
DSR	I	W2	MII_TX_ERR	I
RI_N	I	U3	MII_RX_VAL	I
ETH_C_LED[0]	O	J3	No Connect	NA
ETH_C_LED[1]	O	J4	No Connect	NA
ETH_C_LED[2]	O	H1	No Connect	NA

Serial EEPROM Interface

The Serial EEPROM interface can be used to interface to any SPI device. Primarily it is used to interface to a Serial EEPROM that is used to store the MAC addresses of the three (3) Ethernet controllers along with PCI information. The serial device can also store additional user defined information.

PCI Interface

The MCS1000 contains a device only 32-bit 33MHz interface conforming to the PCI Local Bus Specification 2.2. The MCS1000 does not contain an internal PCI arbiter, but it can master the PCI bus as a device.

Memory Controller

There is a Memory Management Unit (MMU) and a Memory Controller (MC). The MMU and the MC sit on opposite sides of the AHB to VCI bridge. The MMU sits between the ARM9 and the AHB BIU and the MC sits on the VCI bus and interfaces with external memory (SDRAM and Flash) and the Local-bus. For information on the MMU see the CPU Subsystem section of this document.

SDRAM Interface

The SDRAM interface is a 100MHz 32-bit interface. The maximum SDRAM capacity is 64 Megabytes.

Flash Interface

The Flash interface can be 8-, 16-, or 32-bits wide. There are 26 address lines that yield a maximum storage of 64Megabytes with an 8-bit bus, 128Megabytes with a 16-bit bus, and 256Megabytes with a 32-bit bus width.

Local Bus Interface

The local bus interface can be 8-, 16-, or 32-bits wide. The local bus interface waveforms are programmable enabling access to different target devices. The bus transaction can be extended by asserting the ready input signal. The polarity of this control input is programmable. The GPIO pins can be programmed to generate interrupts.

Serial Port

The Serial port uses a standard 16550-compatible UART and it is fully pinned with TX, RX, RTS, CTS, DSR, DTR, DCD, and RI. This port is uses standard digital levels. A common RS-232 driver is required in order to achieve standard RS-232 levels. The registers in the UART are compatible with the 16550 and code that interfaces with a standard 16550 should interface with the UART in the MCS1000. BAUD rates can range from 1200 BAUD to 115.2K BAUD.

General Purpose I/O

There are 16 GPIO pins. Each GPIO can be configured independently of all other GPIOs. The GPIOs allow data input and IRQ generation.

Clock Circuit

All clocks throughout the MCS1000 are generated via a single external 25 MHz crystal.

Bootstraps

There are certain attributes of the MCS1000 that need to be set-up before reset becomes inactive. These configuration settings are controlled via a bootstrap register. The bootstrap register is loaded with values based on the logic levels of the ADC address bus pins immediately after reset is inactivated.

The bootstraps control four main aspects of the MCS1000: Debug mode, Processor speed, Internal ROM disable and User defined

Bootstrap Pins

Bootstrap Pins					
Bootstrap	**Name**	**Description**	**Default**	**Level**	**Pin**
BS7	User_Defined	Defined by user	Disabled	Low	SDRAM_AD25
BS6	Speed_Control_2	Processor Speed	150MHz	Low	SDRAM_AD24
BS5	Speed_Control_1	Processor Speed	150MHz	Low	SDRAM_AD23
BS4	Speed_Control_0	Processor Speed	150MHz	Low	SDRAM_AD22
BS3	Boot ROM	Boot from Internal ROM	Enabled	Low	SDRAM_AD21
BS2	Boot-Up Message	Boot-Up messages through UART, only used for internal boot.	Disabled	Low	SDRAM_AD20
BS1	SDRAM_CONFIG	SDRAM Configuration, (must be pulled high)	Disabled	Low	SDRAM_AD19
BS0	User_Defined	Defined by user	Disabled	Low	SDRAM_AD18

Processor Speed Bootstraps

Processor Speed Bootstraps			
Processor Clock	**BS6**	**BS5**	**BS4**
150.0 MHz	Low	Low	Low
162.5 MHz	Low	Low	High
175.0 MHz	Low	High	Low
187.5 MHz	Low	High	High
200.0 MHz	High	Low	Low
137.5 MHz	High	Low	High
125.0 MHz	High	High	Low
112.5 MHz	High	High	High

MCS1000

Security Processor

MOSCHIP
SEMICONDUCTOR

Electrical Specifications

Absolute Maximum Ratings
Stresses beyond those indicated in the following table may cause permanent damage to the MCS1000 device, reduce device reliability and result in premature failure, even when there is no immediately apparent sign of failure. Prolonged exposure to conditions at or near the absolute maximum ratings may also result in reduced device life span and reduced reliability.

Note: The values in the following table are stress ratings only. They do not imply that operation under other conditions is impossible.

Parameter	Min	Max
Operating Case Temperature	0 °C	+70 °C
Storage Temperature	-45 °C	+125 °C
Core Supply Voltage	1.8V - 5%	1.8V + 5%
I/O Supply Voltage	3.3V - 5%	3.3V + 5%
Input clamp Current		10mA
Output clamp current		25mA

Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max
Tc	Operating Case Temperature	0 °C	+25 °C	+70 °C
Vcc_CORE	Core Supply Voltage	1.7V	1.8V	1.89V
Vcc	I/O Supply Voltage	3.14V	3.30V	3.46V

272-Pin BGA Pin-Out

Top View
(Transparent)
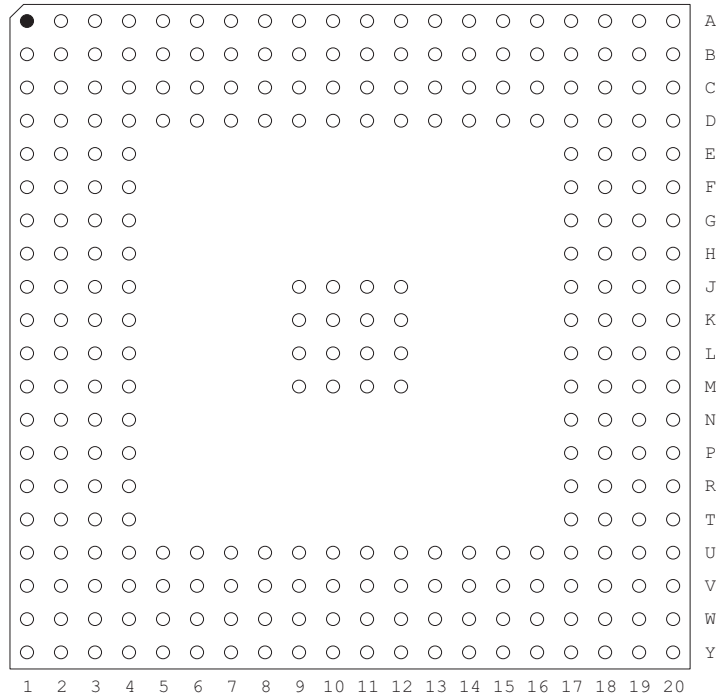
Bottom View
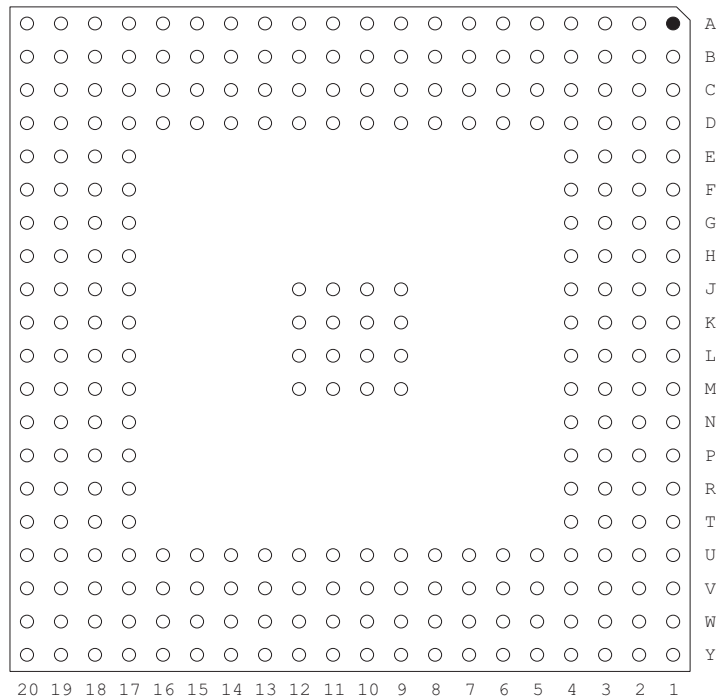
MCS1000

Security Processor

Top View

E

D

Bottom View

E1

D1

Side View

e   b

e/2

B1   B3

B2
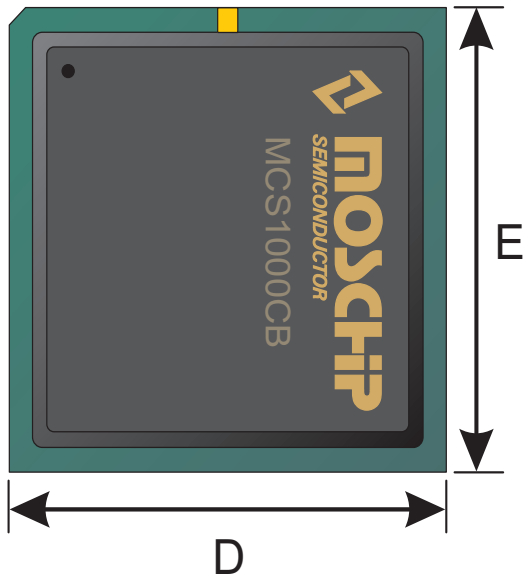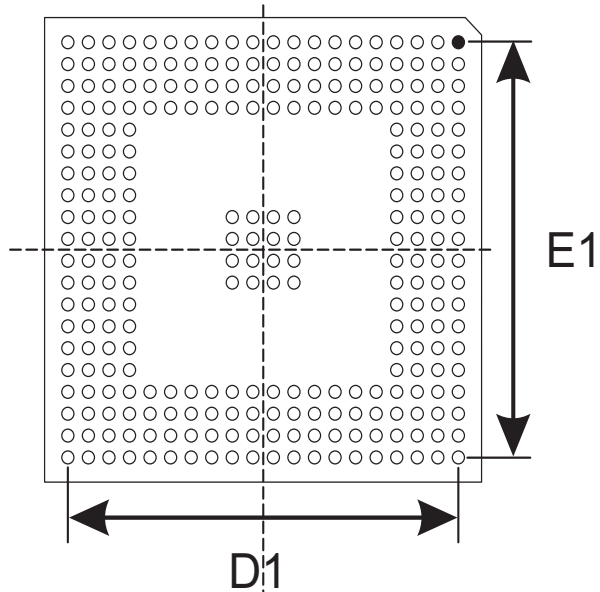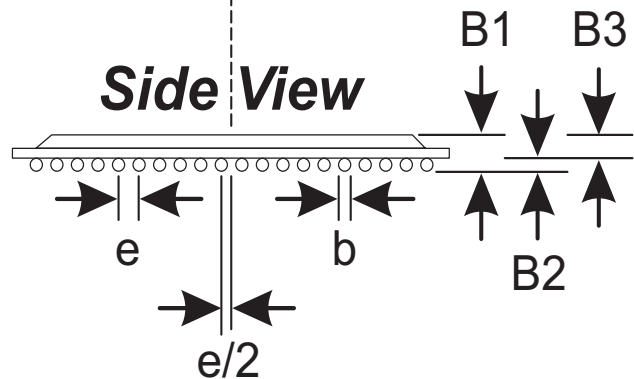
**272-Pin PBGA
Package Dimensions**

Symbol	Millimeters		
	Min	**Typ**	**Max**
B1	1.80	2.15	2.50
B2	0.45	0.60	0.75
B3	1.35	1.55	1.75
b	0.60	0.75	0.90
e		1.27 BSC	
D/E	26.80	27	27.20
D1/E1		24.13 BSC	

Note: "B1" dimension already considers collapse of balls.

Rev.   1.1

IMPORTANT NOTICE

MosChip Semiconductor Technology, LTD products are not authorized for use as critical components in life support devices or systems. Life support devices are applications that may involve potential risks of death, personal injury or severe property or environmental damages. These critical components are semiconductor products whose failure to perform can be reasonably expected to cause the failure of the life support systems or device, or to adversely impact its effectiveness or safety. The use of MosChip Semiconductor Technology LTD's products in such devices or systems is done so fully at the customer risk and liability.

As in all designs and applications it is recommended that the customer apply sufficient safeguards and guard bands in both the design and operating parameters. MosChip Semiconductor Technology LTD assumes no liability for customer's applications assistance or for any customer's product design(s) that use MosChip Semiconductor Technology, LTD's products.

MosChip Semiconductor Technology, LTD warrants the performance of its products to the current specifications in effect at the time of sale per MosChip Semiconductor Technology, LTD standard limited warranty. MosChip Semiconductor Technology, LTD imposes testing and quality control processes that it deems necessary to support this warranty. The customer should be aware that not all parameters are 100% tested for each device. Sufficient testing is done to ensure product reliability in accordance with MosChip Semiconductor Technology LTD's warranty.

MosChip Semiconductor Technology, LTD believes the information in this document to be accurate and reliable but assumes no responsibility for any errors or omissions that may have occurred in its generation or printing. The information contained herein is subject to change without notice and no responsibility is assumed by MosChip Semiconductor Technology, LTD to update or keep current the information contained in this document, nor for its use or for infringement of patent or other rights of third parties. MosChip Semiconductor Technology, LTD does not warrant or represent that any license, either expressed or implied, is granted to the user.

Revision History

Revision	Changes	Date
1.0	Initial Release	14-Mar-2006
1.1	Modified General Description & Block Diagram	6-Apr-2006