Freescale Semiconductor, Inc.

This technical summary provides an overview of the MPC184 security processor, including a brief development history, target applications, key features, typical system architecture, device architectural overview, and a performance summary.

1 Development History

The MPC184 belongs to the Smart Networks platform's S1 family of security processors developed for the commercial networking market. This product family is derived from security technologies Motorola has developed over the last 30 years, primarily for government applications. The fifth-generation execution units (EU) have been proven in Motorola semi-custom ICs and in other members of the S1 family, including the MPC180, MPC190, and MPC185.

2 Typical Applications

The MPC184 is suited for applications such as the following:

- SOHO VPN routers
- Customer Premise Equipment
- Wireless Access Points
- eCommerce servers
- Dedicated Encryption Modules

3 Features

The MPC184 is a flexible and powerful addition to any networking or computing system using the Motorola PowerQUICCline of integrated communications processors, or any system supporting 32-bit PCI. The MPC184 is designed to off load computationally intensive security functions, such as key generation and exchange, authentication, and bulk encryption from the host processor.

The MPC184 is optimized to process all the algorithms associated with IPSec, IKE, WTLS/WAP, SSL/TLS, DOCSIS BPI+, 802.16, and 802.11(WEP). In addition, the Motorola family of security co-processors are the only devices on the market capable of executing

Elliptic Curve Cryptography which is especially important for secure wireless communications.

MPC184 features include the following:

- Public key execution units (PKEU) that supports the following:
  - RSA and Diffie-Hellman
    - Programmable field size up to 2048-bits
  - Elliptic curve cryptography
    - F_2m and $F(p)$ modes
    - Programmable field size up to 511-bits
- Data Encryption Standard execution units (DEU)
  - DES, 3DES
  - Two key (K1, K2, K1) or Three Key (K1, K2, K3)
  - ECB and CBC modes for both DES and 3DES
- Advanced Encryption Standard unit (AESU)
  - Implements the Rinjdael symmetric key cipher
  - Key lengths of 128, 192, and 256 bits.Two key
  - ECB, CBC, and Counter modes
- ARC Four execution unit (AFEU)
  - Implements a stream cipher compatible with the RC4 algorithm
  - 40- to 128-bit programmable key
- Message digest execution units (MDEU)
  - SHA with 160-bit or 256-bit message digest
  - MD5 with 128-bit message digest
  - HMAC with either algorithm
- Random number generator (RNG)
- 8xx compliant external bus interface, with master/slave logic.
  - 32-bit address/32 -bit data
  - up to 66MHz operation
- Optional PCI 2.2 compliant external bus interface with master/slave logic.
  - 32-bit address/data
  - up to 66MHz operation
- 4 Crypto-channels, each supporting multi-command descriptor chains
  - Static and/or dynamic assignment of crypto-execution units via an integrated controller
  - Buffer size of 512 Bytes for each execution unit, with flow control for large data sizes
- 8KB of internal scratchpad memory for key, IV and context storage
- 1.5V supply, 3.3V I/O
- 252MAP BGA, 21x 21mm package body size
- 1.0W power dissipation

4 Typical System Architecture

The MPC184 is designed to integrate easily into any system using the 8xx or PCI bus protocol. The MPC184 is ideal in any system using a Motorola PowerQUICC communications processor (as shown in

Figure 4-1) or any system using PCI. The ability of the MPC184 to be a master on the 8xx bus allows the co-processor to offload the data movement bottleneck normally associated with slave devices. The external processor accesses the MPC184 through its device drivers using system memory for data storage. The MPC184 resides in the memory map of the processor, therefore when an application requires cryptographic functions, it simply creates descriptors for the MPC184 which define the cryptographic function to be performed and the location of the data. The MPC184's mastering capability permits the host processor to set up a crypto-channel with a few short register writes, leaving the MPC184 to perform reads and writes on system memory to complete the required task.
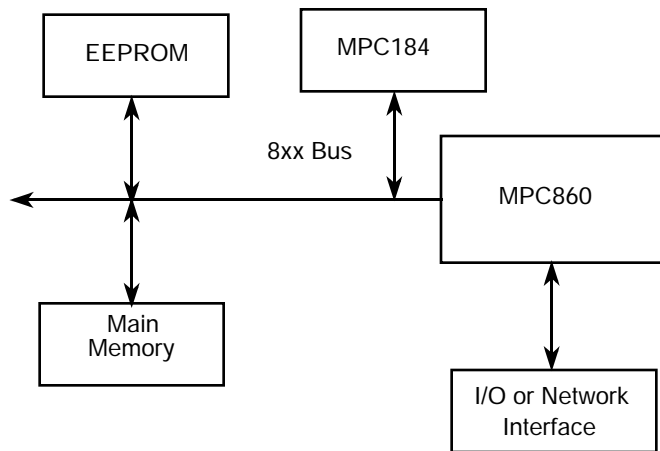

Figure 4-1. MPC184 Connected to PowerQuicc 8xx Bus

Figure 4-2 shows a configuration with the MPC184 communicating with an MPC824x host processor via a integrated PCI bus.
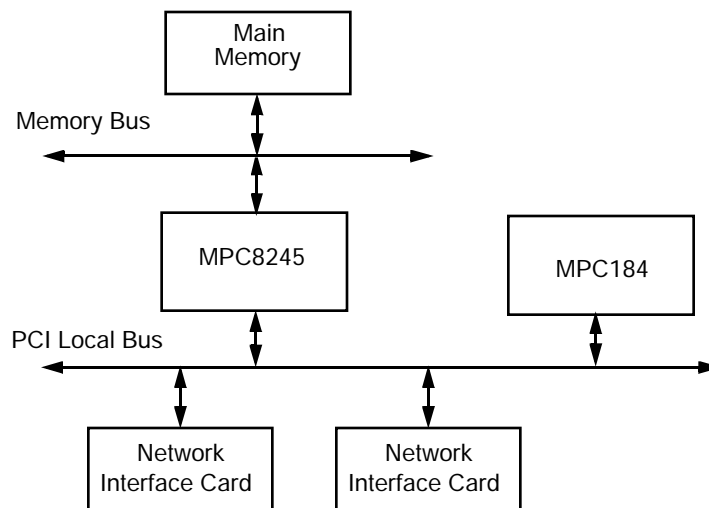

Figure 4-2. MPC184 Connected to host CPU via PCI bus

5 Architectural Overview

A block diagram of the MPC184 internal architecture is shown in Figure 5-3. The mode selectable 8xx/PCI bus interface module is designed to transfer 32-bit words between the external bus and any register inside the MPC184. An operation begins with a write of a pointer to a crypto-channel fetch register which points to a data packet descriptor. The channel then requests the descriptor and decodes the operation to be performed. The channel then makes requests of the controller to assign crypto execution units and fetch the keys, IV's and data needed to perform the given operation. The controller satisfies the requests by assigning execution units to the channel and by making requests to the master interface per the programmable priority scheme. As data is processed, it is written to the individual execution units output buffer and then back to system memory via the bus interface module.
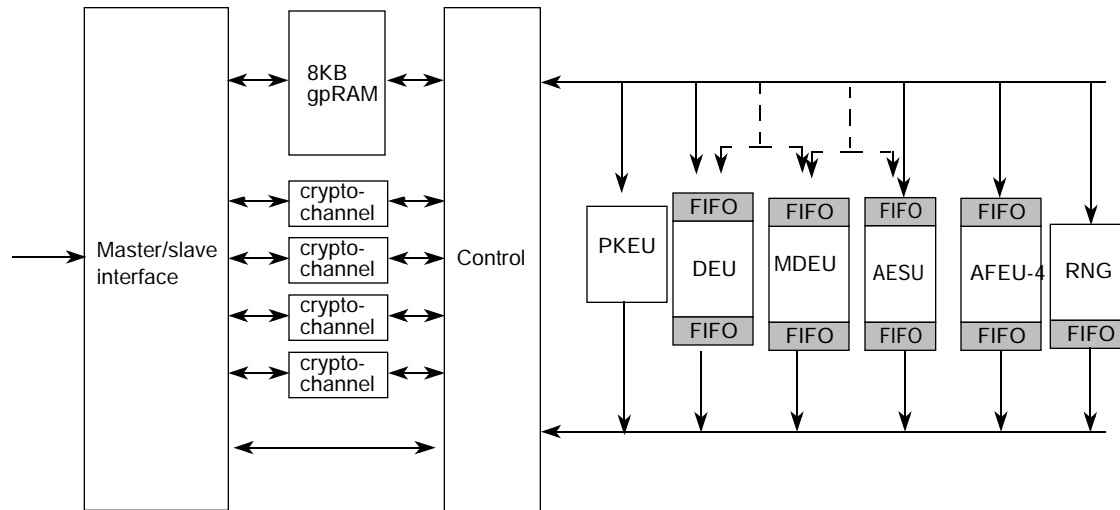

Figure 5-3. MPC184 Functional Blocks

6   Data Packet Descriptors

As an IPSec accelerator, the MPC184's controller has been designed for easy use and integration with existing systems and software. All cryptographic functions are accessible through data packet descriptors, some of which have been defined as multifunction to facilitate IPSec applications. A data packet descriptor is diagrammed in Table 6-1.

Table 6-1. Example Data Packet Descriptor

Field Name	Value/Type	Description
DPD_DES_CTX_CRYPT	tbd	Representative header for DES using Context to Encrypt
LEN_CTXIN PTR_CTXIN	length pointer	Number of bytes to be written Pointer to Context (IV) to be written into DES engine
LEN_KEY PTR_KEY	length pointer	Number of bytes in key Pointer to block cipher key
LEN_DATAIN PTR_DATAIN	length pointer	Number of bytes of data to be ciphered Pointer to data to perform cipher upon
LEN_DATAOUT PTR_DATAOUT	length pointer	Number of bytes of data after ciphering Pointer to location where cipher output is to be written

Table 6-1. Example Data Packet Descriptor

Field Name	Value/Type	Description
LEN_CTXOUT PTR_CTXOUT	length pointer	Length of output Context (IV) Pointer to location where altered Context is to be written
nul length nul pointer	length pointer	Zeroes for fixed length descriptor filter Zeroes for fixed length descriptor filter
nul length nul pointer	length pointer	Zeroes for fixed length descriptor filter Zeroes for fixed length descriptor filter
PTR_NEXT	pointer	Pointer to next data packet descriptor

Each data packet descriptor contains the following:

- Header—The header describes the required services and encodes information that indicates which EUs to use and which modes to set.
- Seven data length/data pointer pairs—The data length indicates the number of contiguous bytes of data to be transferred. The data pointer indicates the starting address of the data, key, or context in system memory.
- Next descriptor pointer

A data packet descriptor ends with a pointer to the next data packet descriptor. Therefore, once a descriptor is processed and if the value of this pointer is non-zero, it is used to request a burst read of the next descriptor.

Processing of the next descriptor (and whether or not a done signal is generated) is determined by the programming of crypto-channel's configuration register. Two modes of operation are supported:

- Signal done at end of descriptor
- Signal done at end of descriptor chain

The crypto-channel can signal done via an interrupt or by a write-back of the descriptor header after processing a data packet descriptor. The value written back is identical to that of the header, with the exception that a DONE field is set.

Occasionally, a descriptor field may not be applicable to the requested service. For example, if using DES in ECB mode, the contents of the IV field do not affect the result of the DES computation. Therefore, when processing data packet descriptors, the crypto-channel skips any pointer that has an associated length of zero.

6.1 External Bus Interface

The External Bus Interface (EBI) manages communication between the MPC184's internal execution units and the external bus. The interface is mode selectable between the 8xx bus protocols, used by the PowerQuicc family of integrated communications processors, and the PCI 2.2 bus protocol. The MPC184 is unique in its ability to act as a bus master on the 8xx bus. All on-chip resources are memory mapped, and the target accesses and initiator writes from the MPC184 must be addressed on word boundaries. The MPC184 will perform initiator reads on byte boundaries and will adjust the data to place on word boundaries as appropriate. The 8xx bus mastering interface allows the MPC184 to off-load both crypto processing and data movement from the PowerQuicc processor, freeing the CPU for other networking system functions, allowing the chipset to achieve best in class performance levels .

6.2 The MPC184 Controller

The MPC184 controller manages on-chip resources, including individual execution units (EUs), FIFOs, the EBI, and the internal buses that connect all the various modules. The controller receives service requests from the EBI and various crypto-channels, and schedules the required activities. The controller can configure each of the on-chip resources in three modes:

- Host-controlled mode—The host is directly responsible for all data movement into and out of the resource.
- Static mode—The user can reserve a specific execution unit to a specific crypto-channel.
- Dynamic mode—A crypto channel can request a particular service from any available execution unit.

6.3 Host-Managed Register Access

All EUs can be used entirely through register read/write access. It is strongly recommended that read/write access only be performed on a EU that is statically assigned to an idle crypto-channel. Such an assignment is the only method for the host to inform the controller that a particular EU is in use.

6.4 Static EU Access

The Controller can be configured to reserve one or more EUs to a particular crypto-channel. Doing so permits locking the EU to a particular context. When in this mode, the crypto-channel can be used by multiple descriptors representing the same context without unloading and reloading the context at the end of each descriptor. This mode presents considerable performance improvement over dynamic access, but only when the MPC184 is supporting a single context (or a single session is being streamed.)

6.5 Dynamic EU Access

Processing begins when a data packet descriptor pointer is written to the next descriptor pointer register of one of the crypto-channels. Prior to fetching the data referred to by the descriptor and based on the services requested by the descriptor header in the descriptor buffer, the controller dynamically reserves usage of an EU to the crypto-channel. If all appropriate EUs are already dynamically reserved by other crypto-channels, the crypto-channel stalls and waits to fetch data until the appropriate EU is available.

If multiple crypto-channels simultaneously request the same EU, the EU is assigned on a round-robin basis. Once the required EU has been reserved, the crypto-channel fetches and loads the appropriate data packets, operates the EU, unloads data to system memory, and releases the EU for use by another crypto-channel. If a crypto-channel attempts to reserve a statically-assigned EU (and no appropriate EUs are available for dynamic assignment), an interrupt is generated and status indicates illegal access. When dynamic assignment is used, each encryption/decryption packet must contain context that is particular to the context being supported.

6.6 Crypto-Channels

The MPC184 includes four crypto-channels that manage data and EU function. Each crypto-channel consists of the following:

- Control registers containing information about the transaction in process
- A status register containing an indication of the last unfulfilled bus request
- A pointer register indicating the location of a new descriptor to fetch
- Buffer memory used to store the active data packet descriptor

Crypto-channels analyze the data packet descriptor header and request from the controller the first required cryptographic service. The controller implements a programmable prioritization scheme that allows the user to dictate the order in which the four crypto-channels are serviced. After the controller grants access to the required EU, the crypto-channel and the controller perform the following steps:

1. Set the appropriate Mode bits available in the EU for the required service.
2. Fetch context and other parameters as indicated in the data packet descriptor buffer and use these to program the EU.
3. Fetch data as indicated and place in either the EU's input FIFO or the EU itself (as appropriate).
4. Wait for EU to complete processing.
5. Upon completion, unload results and context and write them to external memory as indicated by the data packet descriptor buffer.
6. If multiple services requested, go back to step 2.
7. Reset the appropriate EU if it is dynamically assigned. Note that if statically assigned, a EU is reset only upon direct command written to the MPC184.
8. Perform descriptor completion notification as appropriate. This notification comes in one of two forms—interrupt or header writeback modification—and can occur either at the end of every descriptor or at the end of a descriptor chain.

7 Execution Units (EUs)

"Execution unit" is the generic term for a functional block that performs the mathematical permutations required by protocols used in cryptographic processing. The EUs are compatible with IPsec, WAP/WTLS, IKE, SSL/TLS and 802.11i processing, and can work together to perform high level cryptographic tasks. The MPC184's execution units are as follows:

- PKEU for computing asymmetric key mathematics, including Modular Exponentiation (and other Modular Arithmetic functions) or ECC Point Arithmetic
- DEU for performing block symmetric cryptography using DES and 3DES
- AFEU for performing RC-4 compatible stream symmetric cryptography
- AESU for performing the Advanced Encryption Standard algorithm
- MDEU for hashing data
- RNG for random number generation

7.1 Public Key Execution Unit (PKEU)

The PKEU is capable of performing many advanced mathematical functions to support both RSA and ECC public key cryptographic algorithms. ECC is supported in both F(2)m (polynomial-basis) and F(p) modes. This EU supports all levels of functions to assist the host microprocessor to perform its desired cryptographic function. For example, at the highest level, the accelerator performs modular exponentiations to support RSA and performs point multiplies to support ECC. At the lower levels, the PKEU can perform simple operations such as modular multiplies.

7.1.1 Elliptic Curve Operations

The PKEU has its own data and control units, including a general-purpose register file in the programmable-size arithmetic unit. The field or modulus size can be programmed to any value between 160 bits and 512 bits in programmable increments of 8, with each programmable value i supporting all actual field sizes from $i*8$ -7 to $i*8$. The result is hardware supporting a wide range of cryptographic security. Larger field / modulus sizes result in greater security but lower performance; processing time is determined

by field or modulus size. For example, a field size of 160 is roughly equivalent to the security provided by 1024 bit RSA. A field size set to 208 roughly equates to 2048 bits of RSA security.

The PKEU contains routines implementing the atomic functions for elliptic curve processing—point arithmetic and finite field arithmetic. The point operations (multiplication, addition and doubling) involve one or more finite field operations which are addition, multiplication, inverse, and squaring. Point add and double each use of all four finite field operations. Similarly, point multiplication uses all EC point operations as well as the finite field operations. All these functions are supported both in modular arithmetic as well as polynomial basis finite fields.

7.1.2 Modular Exponentiation Operations

The PKEU is also capable of performing ordinary integer modulo arithmetic. This arithmetic is an integral part of the RSA public key algorithm; however, it can also play a role in the generation of ECC digital signatures and Diffie-Hellman key exchanges.

Modular arithmetic functions supported by the MPC184's PKEU include the following:

- $R^2 \bmod N$
- $A'^E \bmod N$
- $(A \times B) R^{-1} \bmod N$
- $(A \times B) R^{-2} \bmod N$
- $(A+B) \bmod N$
- $(A-B) \bmod N$

Where the following variable definitions: $A' = AR \bmod N$, N is the modulus vector, A and B are input vectors, E is the exponent vector, R is 2^s, where s is the bit length of the N vector rounded up to the nearest multiple of 32.

The PKEU can perform modular arithmetic on operands up to 2048 bits in length. The modulus must be larger than or equal to 129 bits. The PKEU uses the Montgomery modular multiplication algorithm to perform core functions. The addition and subtraction functions exist to help support known methods of the Chinese Remainder Theorem (CRT) for efficient exponentiation.

7.2 Data Encryption Standard Execution Unit (DEU)

The DES execution unit (DEU) performs bulk data encryption/decryption, in compliance with the Data Encryption Standard algorithm (ANSI x3.92). The DEU can also compute 3DES and extension of the DES algorithm in which each 64-bit input block is processed three times. The MPC184 supports 2 key (K1=K3) or 3 key 3DES.

The DEU operates by permuting 64-bit data blocks with a shared 56-bit key and an initialization vector (IV). The MPC184 supports two modes of IV operation: ECB (Electronic Code Book) and CBC (Cipher Block Chaining).

7.3 Arc Four Execution Unit (AFEU)

The AFEU accelerates a bulk encryption algorithm compatible with the RC4 stream cipher from RSA Security, Inc. The algorithm is byte-oriented, meaning a byte of plain text is encrypted with a key to produce a byte of ciphertext. The key is variable length and the AFEU supports key lengths from 40 to 128 bits (in byte increments), providing a wide range of security strengths. RC4 is a symmetric algorithm, meaning each of the two communicating parties share the same key.

7.4 Advanced Encryption Standard Execution Unit (AESU)

The AESU is used to accelerate bulk data encryption/decryption in compliance with the Advanced Encryption Standard algorith Rinjdael. The AESU executes on 128 bit blocks with a choice of key sizes: 128, 192, or 256 bits.

AESA is a symmetric key algorithm, the sender and receiver use the same key for both encryption and decryption. The session key and IV(CBC mode) are supplied to the AESU module prior to encryption. The processor supplies data to the module that is processed as 128 bit input. The AESU operates in ECB, CBC, and counter modes.

7.5 Message Digest Execution Unit (MDEU) Module

The MDEU computes a single message digest (or hash or integrity check) value of all the data presented on the input bus, using either the MD5, SHA-1 or SHA-256 algorithms for bulk data hashing. With any hash algorithm, the larger message is mapped onto a smaller output space, therefore collisions are potential, albeit not probable. The 160-bit hash value is a sufficiently large space such that collisions are extremely rare. The security of the hash function is based on the difficulty of locating collisions. That is, it is computationally infeasible to construct two distinct but similar messages that produce the same hash output.

- SHA-1 is a 160 bit hash function, specified by the ANSI X9.30-2 and FIPS 180-1 standards.
- The MD5 generates a 128 bit hash, and the algorithm is specified in RFC 1321.
- The MDEU also supports HMAC computations, as specified in RFC 2104.
- SHA-256 is a 256-bit hash function that provides 256 bits of security against collision attacks.

7.6 Random Number Generator (RNG)

The RNG is a digital integrated circuit capable of generating 32-bit random numbers. It is designed to comply with FIPS 140-1 standards for randomness and non-determinism.

Because many cryptographic algorithms use random numbers as a source for generating a secret value (a nonce), it is desirable to have a private RNG for use by the MPC184. The anonymity of each random number must be maintained, as well as the unpredictability of the next random number. The FIPS-140 compliant private RNG allows the system to develop random challenges or random secret keys. The secret key can thus remain hidden from even the high-level application code, providing an added measure of physical security.

7.7 8KB General Purpose RAM (gpRAM)

The MPC184 contains 8KB of internal general purpose RAM that can be used to store keys, IV's and data. The internal scratchpad allows the user to store frequently used context on chip which increases system performance by minimizing setup time. This feature is especially important when dealing with small packets and in systems where bus bandwidth is limited.

8 Performance Estimates

Bulk encryption/authentication performance estimates shown in Table 8-1. include data/key/context reads (from memory to MPC184), security processing (internal to MPC184), and writes of completed data/context to memory by MPC184, using typical bus overhead.

Table 8-1. Estimated Bulk Data Encryption Performance (Mbps)

	DES CBC	3DES CBC	AES 128	AES 256	ARC4	MD5	SHA-1	3DES/ HMAC-SHA-1(Rx)
64 byte	43	36	38	32	43	38	34	29
128 byte	75	55	60	51	75	66	59	50
256 byte	119	76	83	70	118	100	87	74
512 byte	173	95	104	88	171	135	114	97
1024 byte	223	109	118	100	221	163	136	115
1536 byte	247	114	124	105	252	176	144	123

The MPC184 supports single pass processing of encryption/message authentication. All performance measurements assume descriptor generation and bus availability (66Mhz, 32bit 8xx bus with typical SDRAM read/write latency) are not constraints.

9 Document Revision History

Table 9-1 summarizes the revision history of this document.

Table 9-1. Revision History

Revision No.	Substantive Change(s)
0–0.1	Initial release.
1	Added a Counter mode. Added revision history..
1.1	Updated with new template

THIS PAGE INTENTIONALLY LEFT BLANK

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc. *(vertical text, left margin)*

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140
(800) 441-2447

JAPAN:

Motorola Japan Ltd.
SPS, Technical Information Center
3-20-1, Minami-Azabu Minato-ku
Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.
Silicon Harbour Centre, 2 Dai King Street
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

(800) 521-6274

HOME PAGE:

www.motorola.com/semiconductors

MPC184TS/D

**For More Information On This Product,
Go to: www.freescale.com**