

The Memory Channel

By

James R. (Bob) Davis

Summit Computer Systems, Inc.

Scope of the New Memory Channel Definition

This design provides a flexible, scalable, secure data interface to transfer data to and from storage. This protocol is technology independent, to be supported by current communications links, and removes size and distance limitations on data retrieval, with redundancy and coherency methods.

Purpose of the New Memory Channel Definition

The development of a memory data transport protocol capable of supporting data growth and the necessary data security requirements in the changing microprocessor environment. Memory is data stored in many forms and locations. This Memory Channel protocol is independent of any link technology. This protocol is capable of transparent, secure, access to large, local and remote memory systems, employing coherent and redundant storage methods.

Why specify a new Memory Channel?

The Memory Channel addresses the location and transfer of data to and from a data storage location to another. This new Memory Channel contains no specific information on the structure of physical storage mechanism. This channel does not address data structures within the target storage environment. Only a common understanding of a memory unit address is needed. This concept will expand, significantly, the size and possible location of usable memory and storage in general.

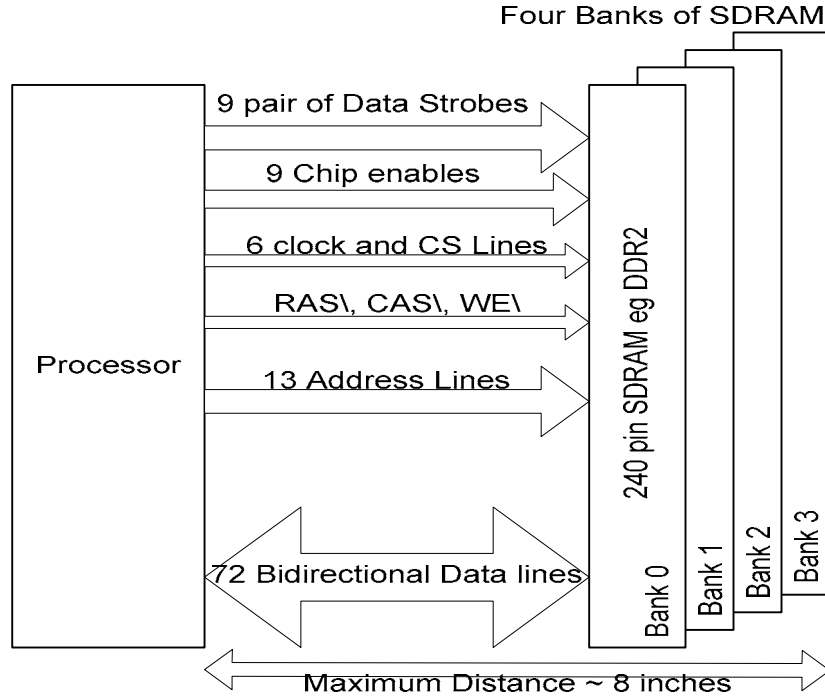
Previous Memory Channels

Different memory access methods have been used in different parts of the system, ranging from RAS and CAS strobes with different numbers of address pins based on current technology to 64 bit addresses on the PCI bus in a single memory space limited to the current addressing.

There has been the segregation of main memory, mostly on multiplexed RAS/CAS basis for addresses and parallel data transfer back to the processor. This presents a very limited range of memory types and sizes to be available.

The Memory Channel

Figure 1. **Typical Current Memory Channel**



Typical Current Processor to/from Memory Configuration

The memory modules are connected to the mother board and processor through about 120 signal lines running at a memory speed of 200 to 400 megabits per second per line.

I/O Channels

While originally separated into a different access space, I/O Channels have been transformed to be simply a part of the memory address space. All the rules that apply to a Memory Channel will continue to apply when a specific I/O port is accessed. The message structure removes any specific differences collapsing the I/O into the general Memory Channel. Mapping in the controller determine the path to the I/O.

Non-Goals of the Memory Channel

Make use of all current and developing physical technologies.

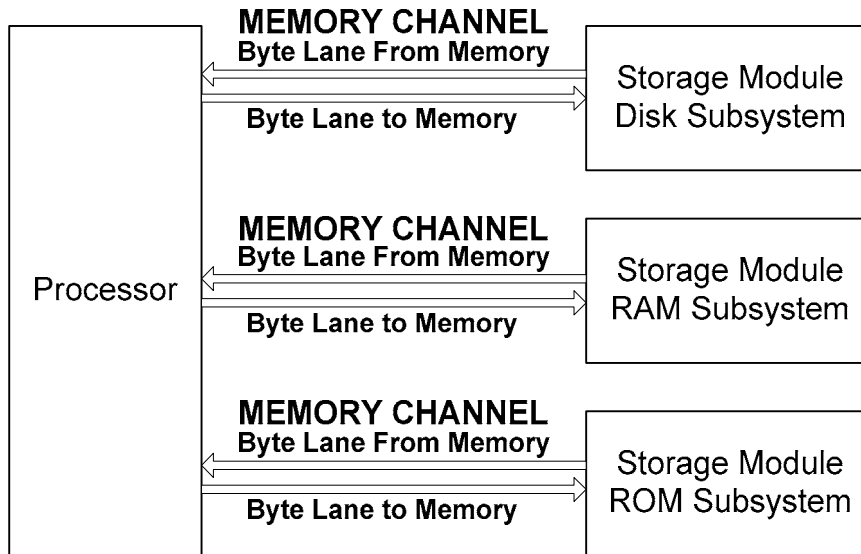
Data transfer technology is changing on a daily basis, the Memory Channel plans to make use of new technology as it is available. There is no redefinition of any communications protocols. The following assumptions are also made: MTU (Maximum Transfer Unit) is a problem of the carrier. Segmentation and Reassembly to adapt the Memory Channel to the MTU is a function of each carrier

Goals for the new Memory Channel

The following goals were considered in the design of the new channel.

The Memory Channel

Figure 2. **New Memory Channel**



Byte Lanes may be a Single wire pair or more.

The MEMORY CHANNEL is implement as a bidirectional connection from processor to/from Memory (Storage)

This Memory Channel relies on smart agents at each end of the channel, it makes no assumptions about the length or width of the channel.

Memory is Memory is Memory

Data storage takes many forms from CPU registers to Archive Tape in a vault. The difference between the various forms of storage is, physical media and blocking factors, the access time and bandwidth and permanency of the data.

Figure 3. **Memory Hierarchy**

<u>Level</u>	<u>Descriptor</u>	<u>Size</u>	<u>Access Time</u>
0	Processor Registers – local 1 clock access	<256 B	XXX ps
1	First Level Cache – I and D caches	~64KB	XXX ps
2	Second Level Cache – unified cache - growing	.5-4 MB	X ns
3	Third Level Cache – Unified possibly multicore	2-16 MB	X ns
4	Main Memory (Now) – current DIMM style memory	.5GB – 1TB	XX-XXX ns
5	Virtual Memory – external smart cache – Ram Disk	XXTB	X us
6	Fast Disk – with cache - journaling	100GB	X ms
7	Active Storage – Main Storage DASD, NFS, CIFS	2TB – 500TB	XX ms
8	Near Storage Backup, Snap shots, LRU Dump	>50TB	XXX ms
9	"ColdStor" slowstore – Low Cost, Slow, Big, Secure	.5 PB – 10PB	<30 s
10	Remote Geo Distributed Storage – Multiple Access	Unlimited	Seconds
11	Redundant Recovery – Business Continuance	Unlimited	Sec - Mins
12	Deep Permanent Storage (Tape etc) Iron Mountian	Unlimited	Sec - Days

The Memory Channel

All different levels of memory store the same information with different access times. The byte retrieved or written is the same.

Support Large Memory

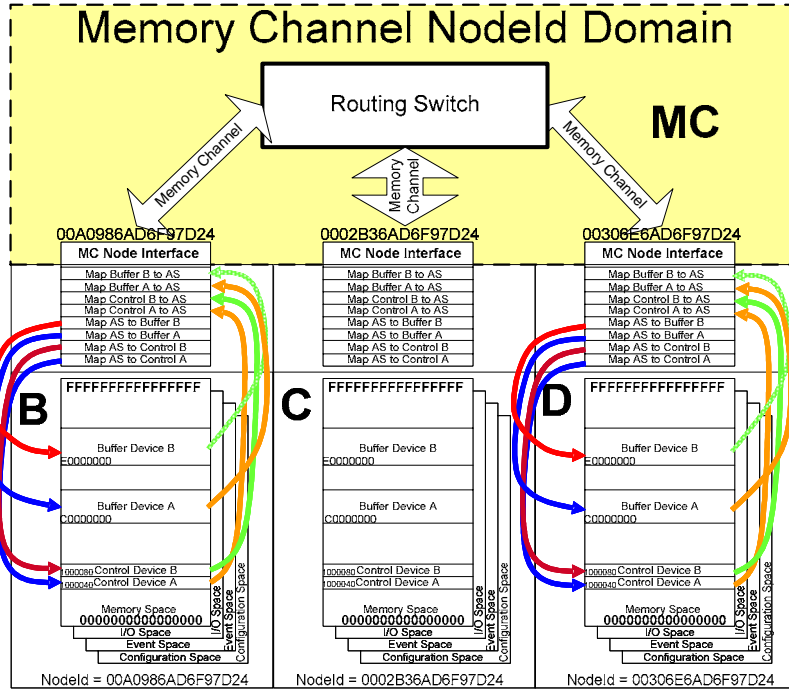
Current memory channels segments have had upper limits of 4GB based on the availability of 32 bit addressing. The Memory Channel should be planned with a minimum of 10 year design life expectancy. If the history of the ISA (Industry Standard Adapter) and PCI (Personal Computer Interface) architectures are any kind of a predictor, it will probably still be viable in 15 years. Consider normal trends, with history, as indicators of future developments. Interconnect speeds still double every 18 months. Typical and Max Memory sizes increase, requiring a additional bit of address every year to 18 months (12 bits in 1974 + 30 years = 42 bits now). The Sun SunFire 15K is offering 512 GB of RAM per domain and currently up to 18 domains now (44 bits) and IBM is just behind at 256 GB of RAM. This would lead to 48-60 bits of memory address space per user requiring the capability of 64 bit addressing.

Alternate forms of storage will become available that will be significantly denser and less expensive over the life of this specification. While the current technology allows for devices in the 200 million devices per die, predictions of this growth by 2014 expand this by at least 2 orders of magnitude. These new devices will both need, support, and provide these larger memory spaces. Processing elements will continue to grow, based on the industry long term technology roadmap. These vendor roadmaps also show technology advancing to 35nm IC technology by 2009, with smaller than 25nm technology within the planning horizon of this Memory Channel. With the smaller geometries, the greater probability of geometry induced faults will increase the need for better error detection in all circuit elements. Planning for the future versions of the specification will help grow to places we have not thought about yet.

The Memory Channel operates between domains defined by the NodeId as the upper level of addressing. In the figure below a transaction is taking place between the BNode and the D Node through the switch in the MC (Memory Channel) domain of NodeId address. This example shows the replication of addresses in all the individual node entities and the necessary translation buffers in each of the interfaces to provide protection of the various domains.

Figure 4. **Switch Controlled Access through Multiple Similar Domains**

The Memory Channel

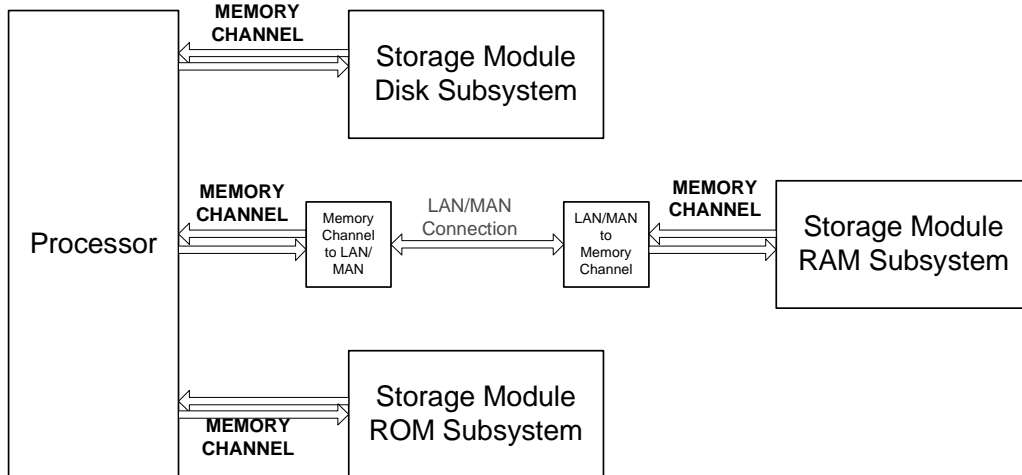


Inter Node Memory Transaction through Buffer Control Registers

Support Remote Memory

Remote memory is defined as memory that is external to local Node physical environment. Once the decision, possibly based on address, to go beyond the local node, another level of addressing is needed to uniformly define location of the target and uniquely identify the target. For example, to transfer data from one PC/unix environment to another PC/unix environment, even if sitting next to each other physically, requires a transition from one physical memory domain to another physical domain in the respective computers, and an addressing capability that distinguishes those domains.

Figure 5. **Remote Memory**



The MEMORY CHANNEL is implement with a LAN/MAN connection carrying the Memory Channel Protocol.

Memory Technology Independent

Memory technology, in all its forms, is progressing rapidly and independent of the processor technology and instruction set technology.

Many of the current processors are designed to support a given memory technology such as DDRx. The complexities of these memories are designed into the processor which also makes the processor slave to that technology. This was much more important prior to the local caching of data when all accesses went to processor external memory. As shown in the hierarchy of memory above the access of die is at level 3 or 4 and moving to higher levels to support the increasing execution speeds of these same processors. It is now time to divorce the actual memory technology from the memory request operation.

Message Based – not bit optimized

The Memory Channel is a byte serial protocol, and may be optimized to use as many transfer lanes as desired for the required performance.

Most accesses to storage are now block oriented with a number of bytes (octets), quadlets (four bytes), or octlets (eight bytes) transferred for each memory access, such as a cache line. This is effectively a message being transferred to the storage from the requesting element for action on a block of data. The Memory Channel recognizes this reality and builds the protocol to support a general purpose storage access and transfer message.

I/O operations have also become memory operations. The difference in delays and the small number of I/O address has long since transitioned to the only difference between I/O data transfers and Memory data transfers is the resolved address..

Common Protocol Layer

The heart of this work is the building of a common protocol that has sufficient capabilities to be used for all memory transfer operations and have the longevity to survive technology of 10 or more years. This is simply a protocol for the handling of the general purpose storage transactions.

Physical Layer Independent

Transportation of the Memory Channel protocol is independent of the protocol. Data link development is an ongoing effort with extensive work in every possible area of development. The Memory Channel will gladly follow and make use of this work and be conveyed over it.

I anticipate that multiple different physical layer technologies will be used in each access that extends more than several millimeters from the processor element. Clearly inter-chassis data transfer requires different technology than 20 cm access methods.

Memory Distance Independent

The demand for memory speed is always present and mediated only by the speed of light and the technology of the devices. Caching has been the most prevalent answer to this problem. This has also reduced the need, within this context, for the same access time from all memory elements. Please see again, the hierarchy of memory above. With time as a relative and controllable quantity, the location of the memory can be extended to whatever length needed to support the use of that memory.

Very large memories can not be physically located next to the processing element due to size constraints. The system designer will now have the ability to tradeoff response time, cost of memory, size of memory, protection of memory as independent variables.

Source and Destination Addressing

Extending the memory domain with global storage requires that the memory channel message has both a destination and source identifiers. This is required for authentication of the source of the message, the return path for the return data, if any, and for security of the transfer.

The Memory Channel

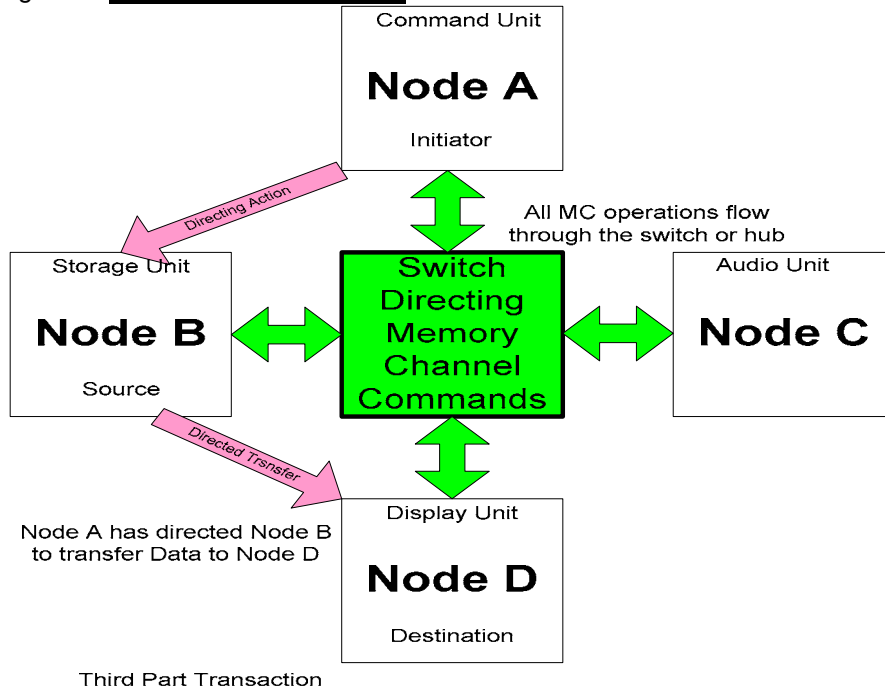
This Memory Channel Protocol specifies the destination address as a combination of destination node and memory offset within that node. Node addressing is derived from a globally unique identifier. This is a 64 bit identifier in the format of, either an EUI (Extended Unique Identifier) or a WWN (World Wide Name). An EUI consist of the IEEE Registration Authority Committee assigned 24 bit OUI (Organizational Unique Identifier) plus the organization guaranteed 40 bit unique number. The WWN consists of 3 fields within the 64 bits consisting of a 4 bit NAA (Naming Authority) mostly fixed at the value 5h which is the IEEE RAC, the 24 bits of the IEEE assigned OUI and 36 bits of organization guaranteed unique number. This forms a 128 bit address that is globally unique.

Third Party Transfers

Third part operations are designed into the Memory Channel. For example, a command module may initiate a transfer of data from a source, such as a disk, to a video or audio processor. The mechanism used is the linked list pointer set with a streaming command.

The Memory Channel

Figure 6. **Third Party Transfers**



Security – Authentication, Validation, Encryption

Any channel used to carry customer data must support Authentication of the user(s), validation of the operation, and hiding of the data with encryption. Authentication identifies the source of the request and the access privileges of the requestor for the transaction proposed. Validation assures the request is complete as received with all information correct. Encryption hides the nature of the transfer to all, assumed, snooping eyes.

Effective RDMA and SRDMA

This Memory Channel supports Remote Direct Memory Access and the Secure Remote Direct Memory Access methods for direct access to the large memory spaces. In the Memory Channel this is anytime other than when the shortest address form is used in a closed system.

Support Redundant Memory Operation

Data reliability now requires redundant storage of all forms. This will necessarily include RAM, DRAM, DISK and any other form of storage for data envisioned. Redundant storage will need to cover both: 1) The reliability, or unreliability, of the storage mechanism and 2) Disaster Recovery of the data from outside the identified threat zone. Mechanism covered by the method will include RAID (Redundant Array of Inexpensive Disks), RAIMM (Redundant Array of Inexpensive Memory Modules), and one or more remote asynchronous, and possibly synchronous, backups and mirrors of the live data.

Each of these mechanisms will have a maintenance plan associated with it. The mechanism and maintenance plan will be dictated by the criticality of the particular data set and will vary from one data set to another and is NOT part of this standard.

This redundant operation is orthogonal to the coherency requirement but can probably use a similar mechanism. While the coherency mechanism is tracking active copies of the live data base this portion of the Directory based, linked list approach to identifying the existing copies, will maintain the coherency of the stored data in the distributed domain.

Support Cache Coherent Operations

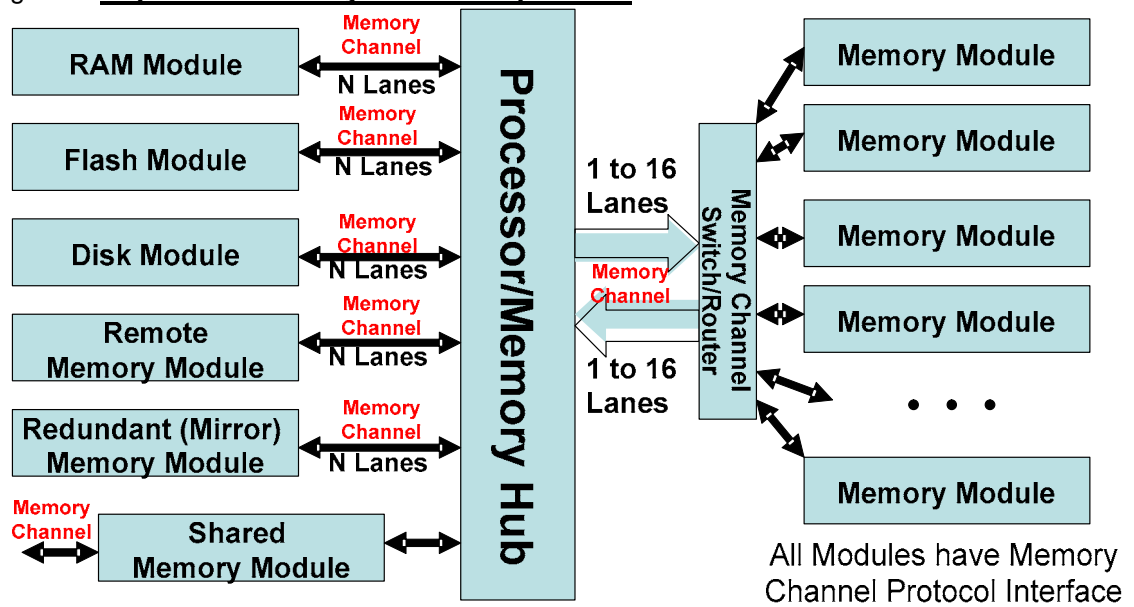
Almost all systems today use one form of caching or another to maintain acceptable performance. With all the caching taking place, a mechanism is required to maintain the coherency of these caches to prevent stale data at the user site.

The best current method for maintaining coherence over a distributed caching system is the SCI protocols, IEEE Std 1596-1992 for a Directory Based Cache Coherency schema. Use of these protocols will require some modification to embrace the nature of the communications protocols and the size of the memory landscape. SCI directory based cache coherency system was designed around 64K nodes with a 48 bit offset into each node and deals with cache lines of data. This system will expand that to 2^{64} potential, although sparsely populated, nodes with a 64 bit offset into node address space and will deal with much larger scalable objects and files over a much larger set of domains. SCI deals only with 64 byte lines in the coherency protocol. This extension work is part of the project and standard.

Scalable Design

Every aspect of this Memory Channel is aimed at a fully scalable system and data design.

Figure 7. **Expanded Memory Channel Operation**



A Complex Multi-use Memory Channel based processor system

Multiple Implementation levels

There is one protocol level for the Memory Channel, but various elements and features of the Memory Channel can be selectively disabled where they are not needed.

The level of implementation required to respond to a request is discernable with the header for each transaction. If the destination of a message is not able to comply with the expected performance, a negative acknowledgement is returned with the error code.

Plug and Play

With a common interface, all memory devices should be able to work together. An example is USB flash memory system.

Cost Effective

With a common interface, all memory devices should be able to work together. An example is USB flash memory system.

Create Vendor Business Opportunities

With a common Memory Channel interface, vendors can create memory of various performance and cost parameters for general use or at a targeted market segment. A model for this business expansion is the Expansion of memory devices enabled by the USB memory devices.

Support Vendor Differentiation

The common interface allow vendors to differentiate their products to improve performance, reliability or cost. Examples are additional levels of caching or different vendor created caching algorithms to improve performance.

Separately defined Encapsulations

This document discusses the protocol for the message and does not define the interface with the transport mechanism.

Among the different structures that can carry the Memory Channel protocol are: Infiniband, PCI Express, USB, Firewire, Ethernet, TCP/IP and the other variant of IP traffic defined through the IETF.

Beneficiaries of the New Memory Channel

The benefits of the new Memory Channel are distributed throughout the supply channel from processor and memory vendors to the end user base.

Processor Vendors

Processor designer and suppliers will have a common target implementation of memory for several generations of processor designs. All constraints of specific knowledge of the operation of the memory are removed and transferred to the vendor of the memory. Operations with respect to decisions respecting the anticipated delay are already being made by the processor and the execution path modified while a memory access is being performed.

Requirements for the refreshing of dynamic RAM and other memory specific function, such as pre-charge cycles become the function of the memory greatly simplifying the design of the processor or memory controller.

Memory Vendors

Memory vendors gain by being able to optimize the overall memory product rather than just meeting the "DIMM" standard. Innovation by the memory designer can improve the performance of a block of memory using statistical techniques similar to those used in the cache on the processor. This is a value added for the vendor allowing for specialization into the different areas in which memory is utilized in the system.

The physical constraints of the packaging are removed and allow the optimization of form factors to meet design parameters.

This Memory Channel interface may be added to new memory die or to a local controller similar to the currently being designed AMB (Advanced Memory Buffer) device to supply FBDIMM (Fully Buffered Dual Inline Memory Module) interface. This provides a far more general operation on the response to a memory request than the AMB design, as currently envisioned although it may be slower in the larger forms.

The larger vision of the interface being implemented at chassis level with multiple technology memory devices providing a uniform face to the requesting process is also valid.

System Builders

The benefit to system builders is a greater variety of memory products available for solving the system objectives. A generalized memory channel interface removes specific design constraints with respect to system layout and the ability to upgrade the memory with changing product needs. Individual memory system design can optimize the design of the memory subsystem signal integrity needs. The tracking of specific memory cell requirements are removed to the maker of those memory cells.

Different speed memory subsystems will work together and perform better from their individual design. Adding faster or slower memory may affect overall systems performance but will always work!

Faster time to market is realized with a general purpose memory interface that is insensitive to the type of memory involved.

Memory constraints of the application are removed as the number of DIMM slots is no long a limit on the overall size of memory.

Memory Module Vendors

Memory Module Vendors can compete for business with more than the best cost for a given configuration. This standard opens the path for memory innovation to meet a set of specification defined by the application and the user. Combinations of memory technologies in one unit should provide a far better solution to specific design goals and computational situations.

Design innovation can provide opportunities for specific design wins that better meet the system builders and end users goals.

The care and feeding of this particular module is the property of the module. RAM with higher refresh requirements can be used on modules with that specific module spending more refresh cycles to take care of that memory. This adjustment is transparent to any other memory in the system.

End Customer, Users

The end user wins by having the ability to upgrade the memory to meet the changing needs. Plug and Play is automatically enabled with each memory sub system taking care of itself. The EUI makes each module and its memory space unique.

Compatibility with existing memory in the system is not a concern when adding new memory, added memory takes care of itself.

Memory modules can be designed to meet special needs, such as graphic, or portable application, while maintain transparency with the other memory systems.

Definitions

The following definitions are used in this document.

Memory – Identifiable data.

Any fully identifiable Addressable location that contains information for use.

Byte – A collection of 8 binary digits

A collection of 8 binary bits, usually described with most significant bit to the left. A Octet is the internationally recognized equivalent of the byte.

Octlet – Eight octets (bytes)

A group of eight Octets (bytes) treated together. A 64 Bit quantity designated as x(63:0).

OUI – Organizationally Unique Identifier

This is a 24bit identifier issued by the IEEE Registration Authority that is guaranteed to be unique to that organization.

EUI – Extended Unique Identifier

The IEEE Extended Unique Identifier is a 64 bit identifier consisting of the OUI issued by the IEEE Registration Authority and 40 bits of identifier specified by the organization and guaranteed to be unique within the organization. This creates a Globally Unique Identifier.

GUI – GUID - Globally Unique Identifier

A globally unique identifier, possibly from another NAA naming authority that has the property of being unique on a Global Basis. This could include the cellular phone pin system or other such system from a recognized authority.

WWN – World Wide Name

World Wide Name (WWN) is a variation on the EUI concept used by INCITS committees T10, T11 and T13 to define the unique serial number located on disk drives and other storage device and propagated through SAS, FibreChannel, and ATAPI7 protocols. As shown below this consists of 3 elements, a NAA field that set to 5h to identify the IEEE RAC as the Naming Authority followed by the IEEE RAC issued 24 bit OUI (Organizationally Unique Identifier) followed by 36 bit of organization generated unique number for a total of 64 bits.

Third Party

A party to a transaction not included in the source or destination.

RDMA – Remote Direct Memory Access

A means of Direct Memory Access to a remote memory facility. This work necessarily deals with the address translation and security methods needed to protect the remote and local memory domains. For more information, see the RDDP/RDMA work being done in the IETF.

SRDMA – Secure Remote Direct Memory Access

This is a derivative of the RDMA protocol with an encryption layer built in to obscure the data being transferred.

Byte Lane – A byte serial bit stream

A byte lane is a serial stream of bytes, shipped most significant bit first in the order described. A byte lane may be duplicated to improve performance by placing different bytes on different links. If the connection is 4 lanes wide, every fourth byte would travel on the same lane.

Nodeld – A Unique 64 Bit address in the Nodeld space

This identifier is made up of either the EUI or the WWN. A separate tag bit is defined to prevent conflict between these two elements.

Nonce

The Nonce field is 32 bits. As the name implies, the nonce is a single use value. It MUST be assigned at the beginning of the security association. The nonce value need not be secret, but it MUST NOT be predictable prior to the beginning of the security association.

MESI States

Simplified MESI coherence states (modified, exclusive, shared, and invalid) are assumed for a particular cached-block copy of home memory:

M (modified). There is one copy of this cache block, whose content is believed to differ from the home memory data.

E (exclusive). There are zero or one copies of this cache block, whose content is the same as the home memory data. This cache-block copy can be changed without informing memory, although its cache-local state changes from exclusive-to-modified.

S (shared). There are zero or more copies of this cache block, whose content is the same as the home memory data. This cache-block copy is read-only.

I (invalid). There is no valid cached-block copy.

Elements in a Linked List of related data structures

Each linked list can be considered to have elements in one of three positions. The Head is the start of the list of participating element and the only element in an unshared list. The Tail is the end of the list and the last element added. All other positions in the linked list are Mid elements.

Head of Linked List

The first element of a linked list. If there is only one member of the linked list this will point back to the Home identifier. All three of it's pointer would contain the same information.

Tail of Linked List

The is the last element in a linked list. In a double linked list such as used here this will also point back to the Home identifier.

Mid of Linked List

All other elements in a linked list contains pointers to the Left Nodeld's and Right Nodeld's. All pointer sets in this document also contain a pointer back to the Home Nodeld.

Design Objectives

The following objectives are assumed.

Generic Memory Interconnect

This protocol will support all types of memory and is independent of technology, speed, distance, and physical implementation.

Remove any memory size and location constraints

Scalability in memory capacity, size and physical location are a requirement of the design.

Extensible Design – Room for Options and Future Developments

Specific room is added to the design to allow modification in the future with features not conceived in the current implementation.

Memory Channel Protocol separated from Link Technology

A specific design objective is to support link technology as it develops over the next generation. The design must not be sensitive to the link implementation employed. A working assumption is that three or more different links may be involved in each path from requestor to target memory.

Does not change with each new Processor

Memory Channel design is independent of the processor type and is constant over several generations of processor. This is the primary economic benefit to the processor and memory vendors.

Plug'n'Play

When a new memory module is discovered by the processor, its availability can be accessed. Discovery methods may be beyond the scope of this project. The EUI makes each memory module unique and prevents mis-addressing of the information.

Capable of RAIMM Operation (RAID with and without rotation)

The Memory Channel supports Redundant Arrays of Inexpensive Memory Modules (RAIMM) to maintain the availability of a specific memory in the event of failure of one component of the array. This is similar to the methods applied to disks in a RAID environment.

Extensive Data Protection in addition to link level protection

The header and the data packet are validated separately from the validation of the link layer.

Data security required

All data must be capable of being secured as it travels over the links. This security requires authentication, validation, and encryption of the message being transferred.

Smart Memory Module

All memory modules are required to provide for their own maintenance.

Memory Caching Write and Read

As memory caching will occur, the cache coherency protocol is needed to assure the management of the data.

Notation used

Memory Channel transfers are concerned with movement of data from one endpoint to one or more other endpoints. Each of the endpoints may chose to treat the data in a different manner. Endianness is in the endpoints point of view and any required conversions should be done at the endpoint. .

The Memory Channel

Octlet based description is helpful in current and future 64 bit register operation. An octlet is 8 octets, (bytes) wide. Transfers on the Memory channel will be aligned to increments of octlets (64 bits, 8 bytes).

Data Notations in the following figures follows IETF rfc1700/STD2

The convention in this document is to express numbers in decimal and to picture data in "big-endian" order. That is, fields are described left to right, with the most significant octet on the left and the least significant octet on the right.

The order of transmission of the header and data described in this document is resolved to the octet level. Whenever a diagram shows a group of octets, the order of transmission of those octets is the normal order in which they are read in English. For example, in the following diagram the octets are transmitted in the order they are numbered.

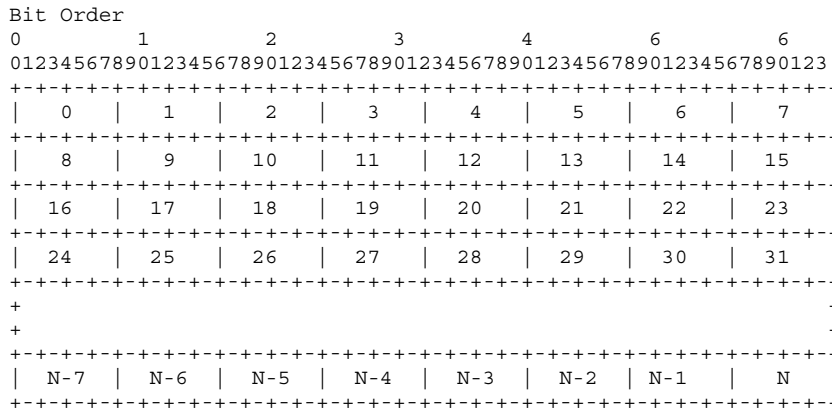
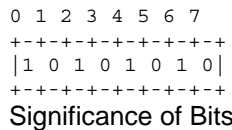


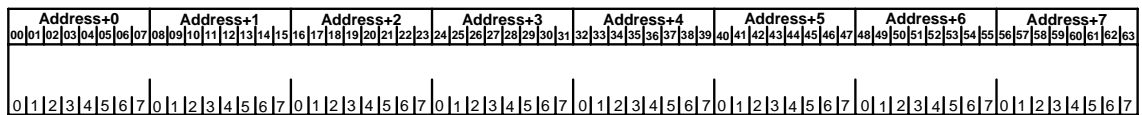
Figure 8. **Byte Order For Transmission**

Whenever an octet represents a numeric quantity the left most bit in the diagram is the high order or most significant bit. That is, the bit labeled 0 is the most significant bit. For example, the following diagram represents the value 170 (decimal).



Similarly, whenever a multi-octet field represents a numeric quantity the left most bit of the whole field is the most significant bit. When a multi-octet quantity is transmitted the most significant octet is transmitted first.

Within this document the octlet will be described with the following layout following the IETF rfc1700 ordering expanded to the 64bit octlet format as shown below.



Elements of the Memory Channel

Overall Message Structure

Header Line

Figure 9. **Memory Channel Header Octlet**

The Memory Channel

Address+0 00 01 02 03 04 05 06 07	Address+1 08 09 10 11 12 13 14 15	Address+2 16 17 18 19 20 21 22 23	Address+3 24 25 26 27 28 29 30 31	Address+4 32 33 34 35 36 37 38 39	Address+5 40 41 42 43 44 45 46 47	Address+6 48 49 50 51 52 53 54 55	Address+7 56 57 58 59 60 61 62 63
Organizational Unique Identifier from IEEE OUI(23:0)			40 bits assigned by Vendor and guaranteed to be unique within Vendors OUI SerialNumber(39:0)				
0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7

The WWN format consist of 3 parts with the NAA (Name Address Authority) that is fixed at 5h (0101b), the IEEE RAC issued OUI and the 36 bit value determined by the OUI holder and guaranteed to be unique.

Figure 11. **WWN NodeId Components**

Address+0 00 01 02 03 04 05 06 07	Address+1 08 09 10 11 12 13 14 15	Address+2 16 17 18 19 20 21 22 23	Address+3 24 25 26 27 28 29 30 31	Address+4 32 33 34 35 36 37 38 39	Address+5 40 41 42 43 44 45 46 47	Address+6 48 49 50 51 52 53 54 55	Address+7 56 57 58 59 60 61 62 63
NAA(5h)	Organizational Unique Identifier from IEEE OUI(23:0)		36 bits assigned by Vendor, guaranteed to be unique within Vendors OUI SerialNumber(35:0)				
0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7

Encoding

Encoding for the address is contained in 4 bits, three bit of selector and an enable for the Tagged Buffer Address.

Figure 29. **Memory Channel Address Structure Encoding**

Address Structure	Address Composition
000	Node Address offset only with an implied NodeId of 0!
001	Reserved
010	EUI based NodeId + Node Address offset
011	WWN based NodeId + Node Address offset
100	Other NAA GUI based NodeId + Node Address offset
101	Other NAA GUI based NodeId Only with no Node Address Offset
110	WWN NodeId only with no Node Address offset
111	EUI NodeId only with no Node Address offset

The Memory Channel

Figure 12. **Address Encoding Fields in Header**

Address+0 00 01 02 03 04 05 06 07				Address+1 08 09 10 11 12 13 14 15				Address+2 16 17 18 19 20 21 22 23				Address+3 24 25 26 27 28 29 30 31				Address+4 32 33 34 35 36 37 38 39				Address+5 40 41 42 43 44 45 46 47				Address+6 48 49 50 51 52 53 54 55				Address+7 56 57 58 59 60 61 62 63						
Protocol Id = "MC" = 4D43h								MC Version	Req	DAS	DAT	SAS	SAT LLHead	M	DataG				MCS	TS	SK	Secure	DROP				Reserved (11:0)				Action (7:0)			

Figure 13. **Address Encoding Fields in Linked List Header**

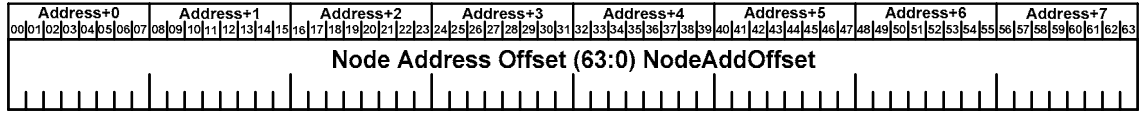
Address+0 00 01 02 03 04 05 06 07				Address+1 08 09 10 11 12 13 14 15				Address+2 16 17 18 19 20 21 22 23				Address+3 24 25 26 27 28 29 30 31				Address+4 32 33 34 35 36 37 38 39				Address+5 40 41 42 43 44 45 46 47				Address+6 48 49 50 51 52 53 54 55				Address+7 56 57 58 59 60 61 62 63			
Left AS	LBTen	Right AS	RBTen	Home AS	HBTen	LeftState (15:0) LeftS				RightState (15:0) RightS				HomeState (15:0) HomeS																	

Offset Address

The offset address is a 64bit value of the starting byte in the memory domain of the node.

The Memory Channel

Figure 14. **Offset Address**



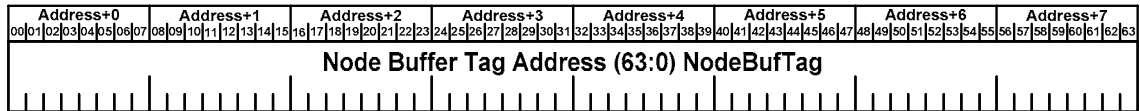
Tagged buffer address

The Tagged Buffer Address, if used is a 64 bit number that point to a buffer address in the respective source or destination.

0b = Buffer Tag address not included

1b = Buffer Tag address follows nodeid and offset.

Figure 15. **Tagged Buffer Address**



Unicast and Multicast

A single M bit is added to specify the destination address is to be interpreted as a Multicast address. (Nodeid or Offset?)

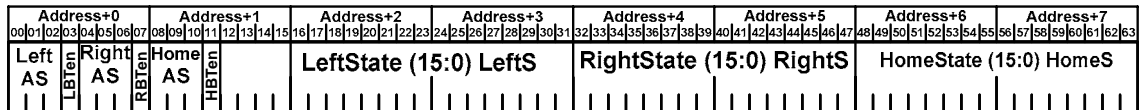
0b = Unicast

1b = Multicast

Linked List Header

The Linked List header describes up to 3 data locations. It describes the Left pointer, the Right Pointer and the Home pointer. This structure is used in both the Cache Coherency and Redundant Data commands.

Figure 16. **Linked List Header**



Elements of the Linked list header are:

Left Address Structure

LBTE - Left Buffer Tag Enable

Right Address Structure

RBTE - Right Buffer Tag Enable

Home Address Structure

HBTE – Home Buffer Tag Enable

Reserved for future use bits

LeftState (15:0) for left pointer information

RightState (15:0) for right pointer information

HomeState 15:0) for home pointer state information

Data Size and Granularity

Memory comes in all sizes and chunk sizes. A chunk is a group of bytes that are treated as a unit such as a “word” or cache line, and will vary in size depending on the application. To accommodate this three fields are needed, the Data Granularity field and a Memory Chunk Size field for odd sizes of any kind, and the Data Size field in unit of the granularity.

The Memory Channel

Figure 17. **Data Granularity and MCS enable locations**

Address+0							Address+1							Address+2							Address+3							Address+4							Address+5							Address+6							Address+7																																																														
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63																																																
Protocol Id = "MC" = 4D43h														MC Version							DAS							DAL							SAS							SAL							LLP							DataG							MCS							IS							SK							Secure							DROP							Reserved (12:0)							Action (7:0)						

Data Granularity, DataG, is a 6 bit field with two encodings based on the most significant bit. If the most significant bit is 0b the remaining bits define the size of the data chunk in the power of 2 of the data increment in bytes. 000000b = single byte element size, 001010b = 1024byte blocks through 19h = 2³¹ byte chunks. Common expected include 1 byte, 16 byte lines, 64 byte lines, 512 bytes sectors, 1024 bytes, 4096 bytes blocks etc. If the most significant bit is 1 then a table is used to determine the size of the chunk in bytes. That table follows:

Figure 30. **Data Granularity Sizes**

DataG Field	Data Chunk Size
0yyyyy	Power of 2 raised to yyyyyb = 2 ^{yyyyyb} Bytes
100000	3 Bytes for audio sample size
100001	5 Bytes for Video Pel descriptions
100010	6 Bytes for Enhanced Video Pel description
100011	Reserved
100100	520 Bytes for Disk Sectors + check bits
100101	528 Bytes for Disk Sectors + check bits
101000	XGA Frame specification
101001	SDTV Frame
101010	HDTV Frame
101011	4Kx2K Motion Pictuer frame
1xxxxx	To be Defined other than above

If the MCS bit is set the chunk size is defined by the addition of the MCS register carrying the byte count of the chunk size up to the limit of 2⁶⁴ bytes.

Figure 18. **Memory Chunk Size Definition**

Address+0							Address+1							Address+2							Address+3							Address+4							Address+5							Address+6							Address+7																														
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63																
Memory Chunk Size (63:0) MCS																																																																															
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7

Data Size field is a 64bit field and is measured in the chunks described here.

Time Stamp Specification

The time specification is used to indicate the time of transmission and to specify the estimated return time. In each case the time shall be specified in the format defined in IETF rfc2030 in seconds and fractions of seconds.

Since NTP (Network Time Protocol) timestamps are critical data and a special timestamp format has been established. NTP timestamps are represented as a 64-bit unsigned fixed-point number, in seconds relative to 0h on 1 January 1900. The integer part is in the first 32 bits and the fraction part in the last 32 bits. In the fraction part, the non-significant low order can be set to 0.

It is advisable to fill the non-significant low order bits of the timestamp with a random, unbiased bitstring, both to avoid systematic roundoff errors and as a means of loop detection and replay detection (see below). One way of doing this is to generate a random bitstring in a 64-bit word, then perform an arithmetic right shift a number of bits

The Memory Channel

return a packet to the origination node with the acknowledgement of the death of the packet, its id, and data.

Cache Coherency Protocol

The coherency of the data structure is needed across multiple instances of caches of various data elements. It is expected that any high performance system will be using caching and read ahead caching of the expected next data read to improve the locality of the data.

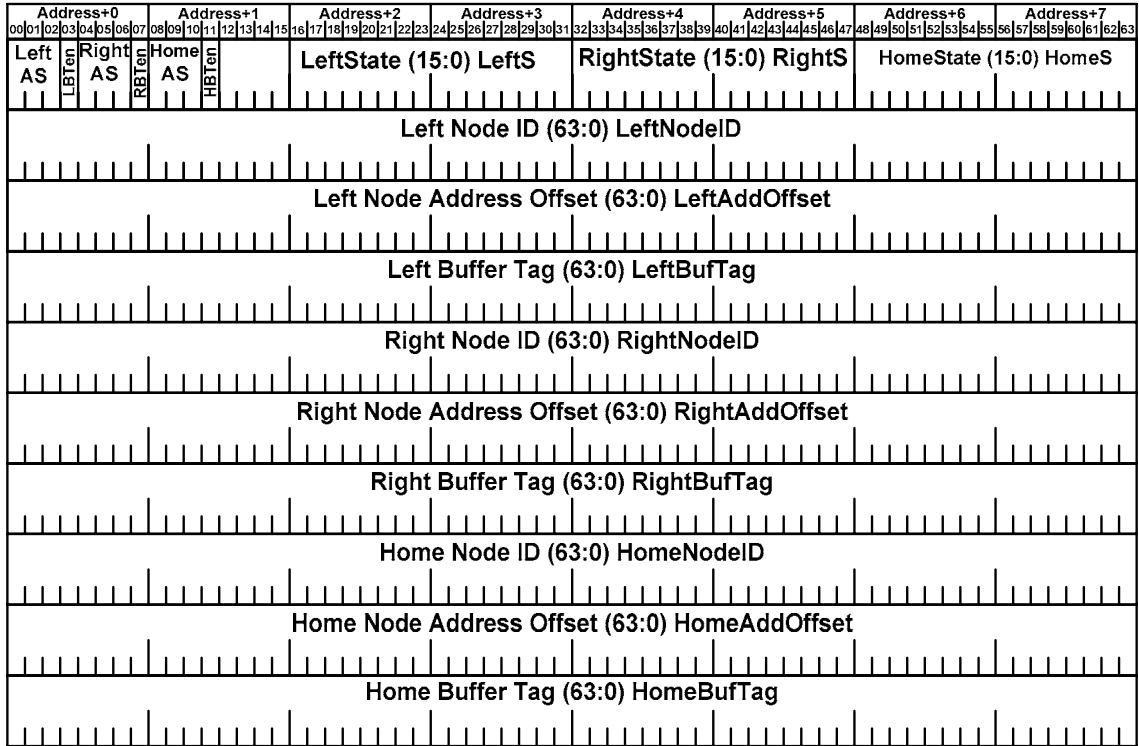
Much work has been done in this area lead by the pioneering work in the IEEE Std 1596-1992. This standard is known as the SCI (Scalable Coherent Interface) Standard.

The SCI standard dealt with cache lines in memory and worked over the 64 bit address space that was broken into a 16bit nodeid and a 48 bit offset into each node. While this seemed extravagant at that time, time has overcome the document with the strong need to extend the node space to the 64 bit EUI/WWN Nodeid space with offsets of 64 bits into each node. Another major change is the size of the coherent chunks, SCI dealt with cache lines while this protocol must extend that to include larger objects and allow up to 2^{64} bytes in size. The addition of the Buffer Tag addresses is included to be compatible with some OS and IETF definitions.

This Memory Channel will follow the transition diagrams for state of the cached elements as described in the IEEE 1596-1992 SCI document obtainable from the IEEE.

The SCI document uses the term Forward Pointer and Backward Pointer in the transition and pointing elements. This document will use Left Pointer in place of the Forward Pointer and Right Pointer in place of the Backward Pointer.

Figure 21. **Complete Linked List Pointer Format**



Directory based linked list approach

This naming convention differs from the SCI (IEEE 1596-1992) with RightId being used instead of the ForwardId and LeftId being used in place of the BackwardId use in the SCI Documents. Each of the NodeId's is 64 bits in length replacing the 16 bit NodeId's used in IEEE1596-1992. This change supports the linked list structure for both the Coherent operations and for the Storage redundancy operations. For the Coherency operations the Buffer Tag fields are not used. For a complete definition of the cState and mState operations see the SCI document.

Memory State information

Memory State information is contained in a 4 bit field consisting of:
 Bits 0 and 1 are mState information.
 00b = Home
 01b = Gone
 10b = Fresh
 11b = Wash

Full descriptions of this memory coherence states are available in the SCI documents.

Cache State Information

The state information is contained in a 4 bit field consisting of 2 bits of directory position information and 2 bits of state information.

Bits 0 and 1
 00b = Memory State
 01b = Directory Head element
 10b = Directory Middle element
 11b = Directory Tail element

The Memory Channel

Bits 2 and 3:

00b = Modified

01b = Shared

10b = Exclusive

11b = Invalid

The 7 bit cState transition states are listed in the SCI document C code.

Commands

The Action field of the first line conveys commands for general data movement, coherent data movement and Redundant Data movement.. The basic command structure supports Read, Write, Move, Stream, Lock, and Event transactions.

The Memory Channel

Request Actions

For Requests Bits 0,1 – define operation category – Normal, Coherent, Redundant, Stream.

For Requests Bits 2,3,4 – define operation – Read Write, Move, Lock, Stream, Message, Event,

For Requests Bits 5,6,7 – define sub operations within the categories.

The first four lines are the different operational domains, this is followed by the expansion into the next level of operation followed by the expansion of that mode with detailed operations.

Operation Space Bits 0,1	Major Command Bits 2,3,4	Expansion Command Bits 5,6,7	Function
00	xxx	xxx	Non-Coherent Memory operation
01	xxx	xxx	Coherent Memory operation, Cache operations
10	xxx	xxx	Redundant Storage Operation
11	xxx	xxx	Streaming Data Transfer Operation
xx	000	000	Read from location
xx	000	000 - 111	Read Reserved
xx	001	000	Write Data to location
xx	001	000 - 111	Write Reserved
xx	010	000	Move – Source to Destination
xx	010	001	Move – Destination to Source
xx	010	010	Move – Destination to Left NodeId
xx	010	011	Move – Destination to Right NodeId
xx	010	100	Move – Left NodeId to Right NodeId
xx	010	101-111	Move Reserved
xx	011	000	Lock request Lock status
xx	011	001	Release Lock Status
xx	011	010	Test Lock Status
xx	011	011-111	Lock Reserved
xx	100	000	Stream – Source to Destination
xx	100	001	Stream – Destination to Source
xx	100	010	Stream – Destination to Left NodeId
xx	100	011	Stream – Destination to Right NodeId
xx	100	100	Stream – Left NodeId to Right NodeId
xx	100	101-111	Stream Reserved
xx	101	000	Send Message from Source to Destination
xx	101	001-111	Message Reserved
xx	110	000	Event – Time Sync
xx	110	001	Event – Send Time Stamp TS
xx	110	010	Event – Return Time Stamp TS

The Memory Channel

xx	110	011	Event – Set Time= TS
xx	110	100-101	Event Reserved
xx	110	110	Event Sync Event
xx	110	111	Event - Interrupt
11	111	001	Storage – Read Storage Control Block (Inode) file information
11	111	010	Storage – Write/update Storage Control Block (Inode) file information
11	111	011	Storage – Return MD5 Signature for file pointed to by Control Block (Inode)
11	111	100-111	Storage Reserved

Fig 22. **Request Action Table**

Respond Action Definitions

For Reply operations with the Request/Reply bit set to 1:

ACK/NAK 0 = Ack 1 = Nak	Action expansion bits 1,2,3	Response Action
0	000	Acknowledgement of Label = Label(31:0)
0	001	Acknowledgement through Label = Label(31:0)
0	010	Data Attached for Label = Label(31:0)
0	011	Operation Complete
0	100	Delayed response Expect Data for Label(31:0) at TS
0	101	Throttle Wait for resume
0	110	My Time Stamp is TS
0	111	Returning your Time Stamp TS = received TS
1	000	Corrupted Data
1	001	Busy – Try Later
1	010	Unable to perform Requested Operation
1	011	Unable to comply – bad address
1	100	Buffer Full – Wait for resume
1	101	Reserved
1	110	Buffer Overflow
1	111	Unknown Failure

Bits 4 through 7 not currently defined and shall be set to 0000b.

Figure 23. **Memory Channel Response Codes**

Redundant Storage Protocol

A double link list is created to keep track of the copies of a memory segment or file. The lists point to the equivalent of the inode used in current file systems expanded with the addressing structure provided by the Memory Channel.

Storage Location pointers

Three pointers are located at each storage location, the Leftward Pointer, the Rightward Pointer and the Home Pointer.

The Rightward Pointer points to the Next Storage location, if there is not a next (rightward) storage location this points back to the primary (Home) location.

The Memory Channel

The Leftward Pointer points to the previous location in the linked list of storage, if there is no previous pointer location, this element is at the Head of the list, and this pointer also points back to the Home pointer.

Home Pointer is the initial storage location. If there is only one storage location, all three pointers contain the same value.

Storage Location Pointer details

Each of these pointers contains the information on the address and data structure of the storage environment

Implementation format

The format for the Doubly Linked List is the same format as used for the Coherency Protocol shown above.

Encryption

The security of the data requires the use of encryption. The specified mode for encryption in this protocol is the AES-CTR mode defined in National Institute of Standards and Technology (NIST) published Advanced Encryption Standard (AES) using the Counter Mode.

Initialization Vector

The Initialization Vector (IV) is generated for each packet and may not be used more than once. This Initialization Vector consists of three elements, a nonce, an modified Time Stamp and a 32 bit counter. The nonce is 32 bit number provided by the IKE key generator or other method that meets the NIST requirements, the IV is the current Time Stamp TS with a random number inserted into the low order bit positions that are not driven by the clocking circuitry, normally set to 0, and the counter is a 32 bit value starting at 00000001h and incrementing with each successive encrypted block.

The generation and maintenance of the AES 256 bit key is described in the Internet Key Exchange (IKE) Protocol. IETF rfc-2409 and rfc-3526.

The third element of the Initialization Vector is a 64 bit quantity that will not be reused in the life of the Key. The Memory Channel uses a TimeStamp for that purpose.

Encryption Key

The selection and distribution of the 128 to 256 bit encryption key for the encoding and decoding of the transported data is well beyond the scope of this document and the IETF IKE (Internet Key Exchange) documents should be consulted.

The selection of the key from a previously agreed upon list of keys is accomplished in the Memory Channel protocol by hashing the completed Source Address (128 bits), the Source Address TAG address (64 bits), the Destination Address (128 bits), the Destination Address TAG address (64 bits) and the Security Key (64 bits) and using that as the index into keys to be used. Any unused parts of the addressing structure are set to 0h for the hash algorithm. Total hash is over 56 bytes. The length of the hash results will be dependent on the Key List structure.

KeyListIndex = Hash(SrcNodeID,SrcBufTag,DestNodeID,DestBufTag,SK)

Security Key

The Security Key is a 64 bit key used to authenticate the source of the data.

Figure 24. **Memory Channel Security Key**

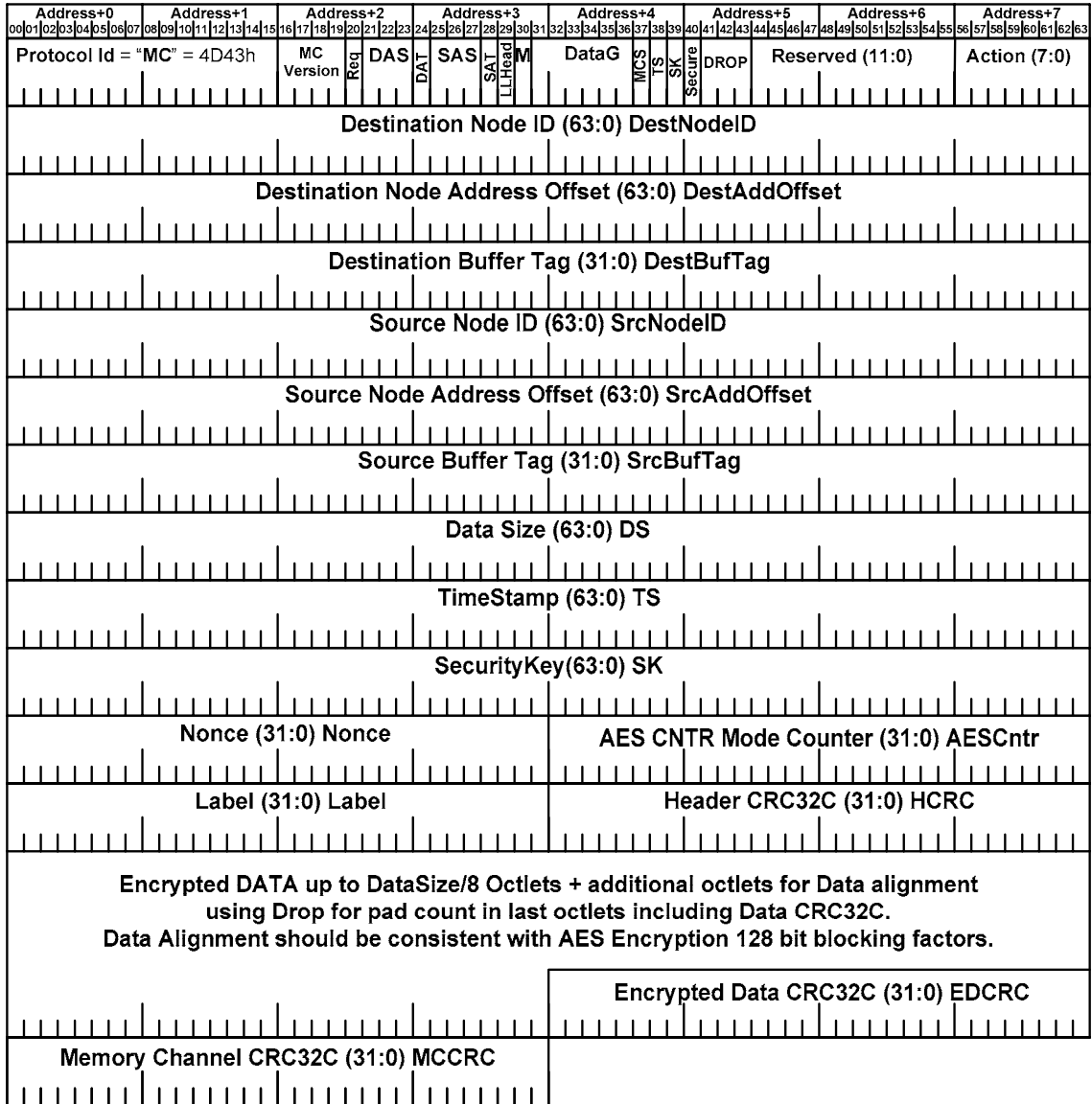
The Memory Channel

Address+0								Address+1								Address+2								Address+3								Address+4								Address+5								Address+6								Address+7							
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
SecurityKey(63:0) SK																																																															
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7

Completed Encrypted Data Packet

The format of a secure packet is demonstrated as follow:

Figure 25. **Memory Channel Encrypted Data Format**



Integrity Checking

Three CRC32C (Cyclical Redundancy Check) checks are in the transmission protocol. All CRC's used are CRC32C a 32 bit CRC with a generator of 0x11edc6f41. This is the CRC used by iSCSI (internet Small Computer Systems Interface) and has better fault coverage than the standard ITU-CRC32 or IEEE-CRC32 (the Ethernet CRC). For additional information see IETF rfc-3385.

Header CRC

This CRC32C follows the header information and is used to validate the header prior to processing of the data section.

Normal and Encrypted Data CRC

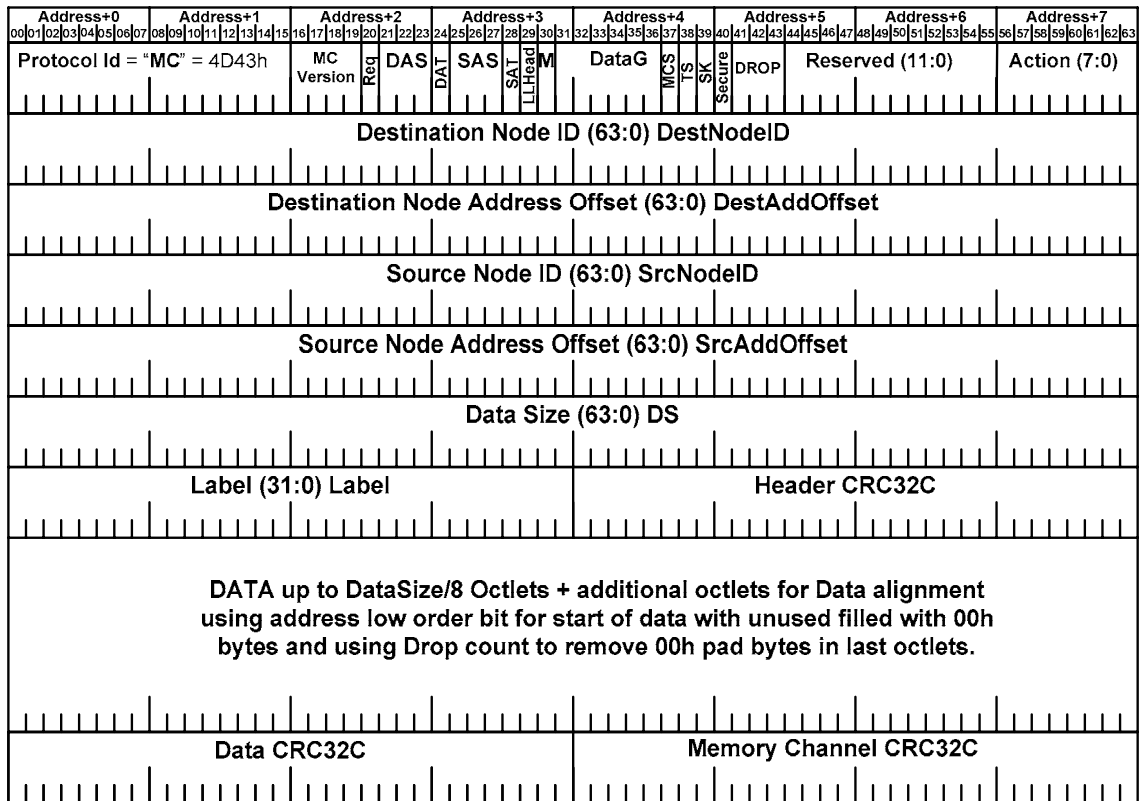
This CRC32C covers the data within the encrypted portion of the message. This is used to validate the decrypted data.

End to End packet CRC

This CRC32C trails validates the entire packet and is plaintext for validation of the message including any encrypted portion.

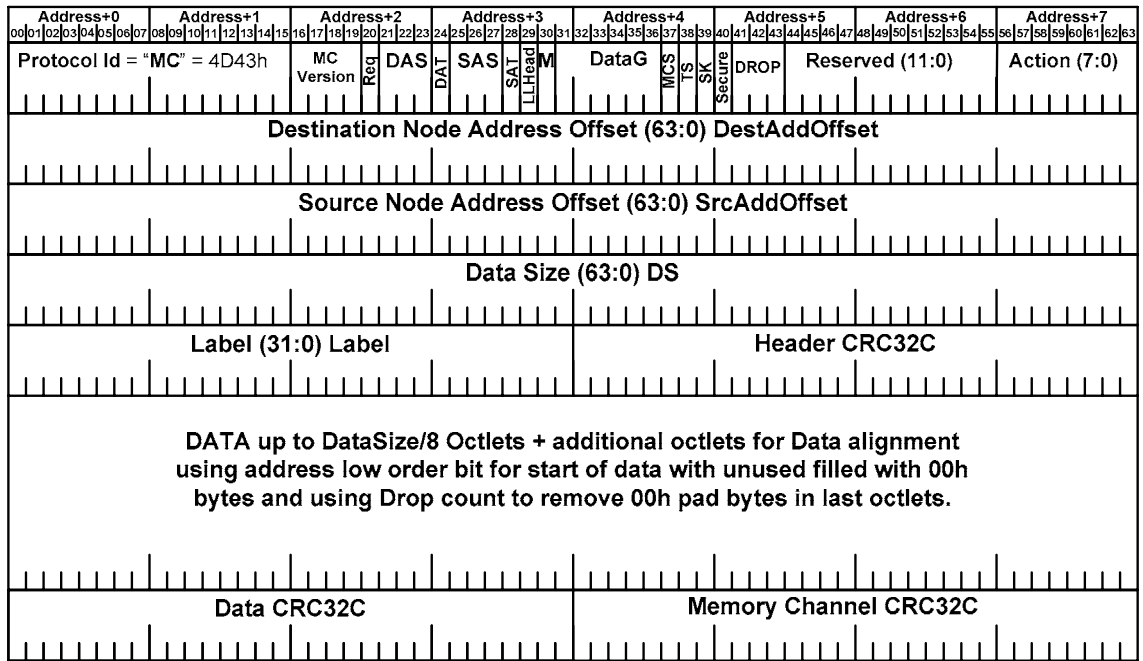
Memory Channel Packet Examples

Figure 26. Typical **Memory Channel Message**



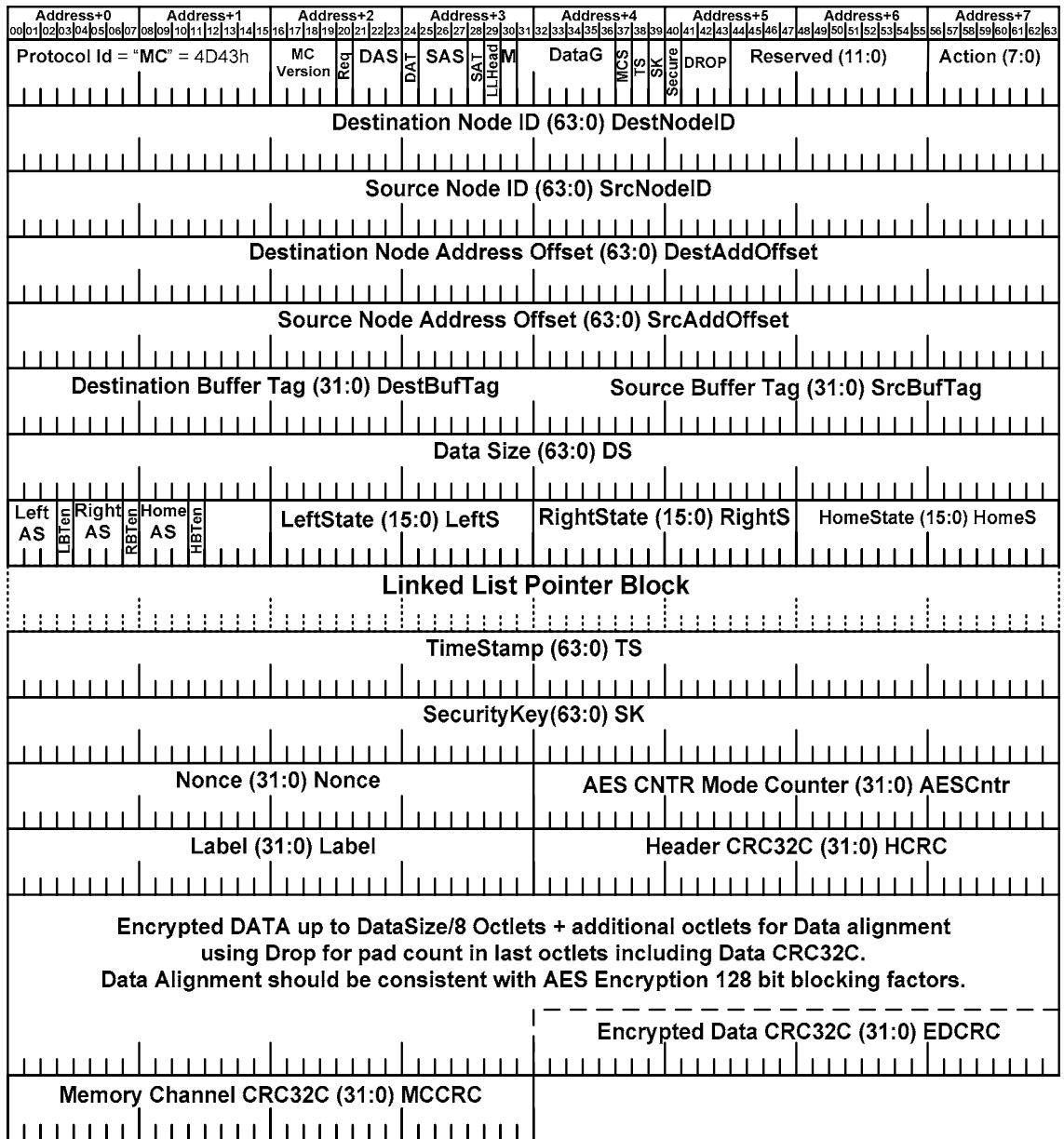
The Memory Channel

Figure 27. **Short Memory Channel Message**



The Memory Channel

Figure 28. **Complete Memory Channel Message with options**



END.