Scalable Coherent Interface

Introduction

*Scalable Coherent Interface* (*SCI*) is the modern equivalent of a Processor-Memory-I/O bus and a Local Area Network, combined and made parallel to support distributed multiprocessing with very high bandwidth, very low latency, and a scalable architecture that allows building large systems out of many inexpensive massproduced building blocks.

SCI reduces the delay of interprocessor communication by an enormous factor compared to even the newest and best interconnect technologies that are based on the previous generation of networking and I/O channel protocols (FibreChannel and ATM), because SCI eliminates the need for run-time layers of software protocol-paradigm translation. A remote communication in SCI takes place as just a part of a simple load or store opcode execution in a processor. Typically the remote address results in a cache miss, which causes the cache controller to address remote memory via SCI to get the data, and within on the order of a microsecond the remote data are fetched to cache and the processor continues execution.

To make program porting easy, the old protocols can be layered on top of SCI transparently. Of course, such an implementation only gains SCI's raw speed factor: to get SCI's full potential speedup, applications will need to eliminate the protocol overheads by using global shared memory directly.

The old approach, moving data through I/O-channel or network-style paths, requires:

- assembling an appropriate communication packet in software
- pointing the interface hardware at it
- initiating the I/O operation, usually by calling a subroutine
- when the data arrive at the destination, hardware stores them in a memory buffer
- hardware alerts the processor by an interrupt when a packet is complete or the buffers are full
- software then moves the data to a waiting user buffer (sometimes this move can be avoided, in the latest systems)
- user application examines the packet to find the desired data

Typically this process results in latencies that are tens to thousands of times slower than SCI. These latencies are the main limitation on the performance of Clusters or Networks of Workstations.
Scalable Coherent Interface

Basic Features

The SCI interface standard defines a point to point communication between neighbour nodes reducing the non-ideal transmission line problems of bus-based systems. It is designed to scale well as the number of attached processors increases.

The transfer rate is 1 GByte/s point-to-point. SCI allows up to 64K nodes to be connected to an interconnect. Memory may be shared by all processors. The addressing scheme uses a 64-bit Fixed Addressing Model. It uses 48 bit as a node offset address and 16 bit for the node addressing.

SCI defines an interface standard that enables the use of many different interconnect configurations ranging from simple rings to complex multistage switching networks. The SCI-Protocols define a set of packet based bus transactions. They can very effectively be used in clustered systems because they have been designed for distributed processing and are based on an underlying multiprocessor architecture. The protocol use small packets (16-288 bytes including header) to carry data and commands between nodes with low latency. Packets can be pipelined to achieve high overall transfer rate.

Basic features

• shared memory programming model
• directory based cache coherence protocol
• distributed directory scheme
• 64K addressable nodes; 16 bit node identifier
• Split phase transaction protocol
• support for non-switched and switched topologies
• Packet switching network, ring or crossbar
• physical level definition
• logical level definition
• C language model of protocols
• 1 GByte/s 16bit parallel; 1Gbit/s serial
Scalable Coherent Interface

Implementations

Computers using SCI:

- **Hewlett-Packard/Convex Exemplar SPP-2000**
  (Shared memory, PA/RISC based)
  http://www.top500.org/ORSC/1997/exemplar.html
- **National Supercomputer Centre (NSC) Monolith, Sweden/2002**
  200x Dual XEON 2.2GHz w/t Dolphin/SCALI SCI
  TOP500 11/03 rank: 103
  http://www.nsc.liu.se/systems/monolith/

Implementations by:

- Dolphinics, www.dolphinics.com

References:

- see above
- Hellwanger, Reinefeld: SCI: Scalable Coherent Interface
- Hwang, Xu: Scalable Parallel Computing

![SCI card (3D) by Dolphinics](image)
Scalable Coherent Interface

Transactions and Protocols

Communication among nodes is accommodated by a set of SCI transactions and protocols that include support for:

- data read/write
- cache coherence
- synchronization primitives
- message passing

All transactions are sent as SCI packets between source and destination nodes. Protocols are provided to handle flow control, error recovery, and deadlock prevention. The transaction format definition is independent of the network topology.

SCI uses a distributed directory-based cache coherence protocol. Each shared line of memory is associated with a distributed list of processors sharing that line. All nodes with cached copies participate in the update of this list.

The shared cache lines are linked together by a double linked list. The coherence list pointers are the nodes addresses. The typical cache line size is 64 bytes.

The cache coherence transaction manipulate a linked-list structure used to maintain a coherent memory image.

![Directory structure diagram]
## Scalable Coherent Interface

### Transaction Formats

#### Request
- **Block Read:** Head
- **Block Write:** Head, 64 or 256 bytes
- **Selected Read:** Head
- **Selected Write:** Head, 16 bytes
- **Selected Locks:** Head, 16 bytes
- **Cache Updates:** Head, 16 bytes
- **Block Moves:** Head, 16, 64 or 256 bytes

#### Respond
- **Head, 64 or 256 bytes**
- **Head**
- **Head, 16 bytes**
- **Head**
- **Head, 16 bytes**
- **Head, 0, 64 or 256 bytes**
- **no respond**

### Transaction Formats

#### Requester
- **Request**
- **Echo**

#### Responder
- **Respond**
- **Echo**
Scalable Coherent Interface

Synchronization Primitives

SCI supports two types of lock primitives, cached and non-cached.

- **Cached locks** are implemented by temporarily locking a cache-line in the only_dirty (exclusive modified) state while these instruction sequences are executed.

- **Non-cached lock transactions** (e.g. swap, swap-and-compare) are provided for accessing non-cached shared data.

An important feature of SCI is the ability to interface to other busses. A bus converter can simply translate the SCI commands to native bus cycles because they are very similar. Two cases are handled with special care, **bus locking and cache coherence**. Most backplane busses have a atomic read-modify-write operation to manipulate semaphores and other critical data. Since SCI is a four phase transaction protocol with no guaranteed delivery in order, the lock is defined as a single SCI command.

SCI supports message passing as defined by IEEE Std. 1212. A standard non-coherent write64 transaction is used to send short messages to a specified control and status register (CSR) within the target node. A typical application for such transaction is accessing control registers of an I/O bridge.

Passing of interrupts between CPU nodes and/or I/O bridges is supported by a standard SCI write transaction to a CSR register.
Scalable Coherent Interface

Physical level

Signaling

SCI uses a narrow 16-bit data path (plus clock and one flag bit) at 2ns/word (250MHz clock, both edges active), to control the interface chip pin-count problem and make switch elements more practical. The signals are transferred using differential ECL voltage levels. One unidirectional SCI connection requires 36 signal wires. The complete bidirectional SCI link sums up to 72 signal wires.

Packets

A packet consists of three main sections:
- header section
- address and data section
- error check symbol

The first word of the packet contains the ID code of the final receiving node, the target. By looking at the first word of the packet, a node or switch can quickly determine where to route this packet.

Packet format
Scalable Coherent Interface

Physical level

The control word of the header controls packet flow and network access. Priority arbitration is supported with round robin arbitration on each level.

The command word of the header contains the transaction command and a sequence number. The sequence number is a tag to identify a packet.

A node may send many requests (up to 256; 64 ?) before a response is received. This transaction pipeline can cause responses to returned out-of-order, and therefore a sequence number is needed to identify a response with the corresponding request.

The command field contains the command a receiver must execute.
Scalable Coherent Interface

**Topology**

The ring interconnect is the simplest structure. There, nodes pass packets to their neighbour. In such a structure are no active components except the nodes. The nodes have to control the arbitration, priority and forward progress schemes.

A switched solutions is provides more performance:

Modular Switch for System Area Networks from Dolphin

- Scalable switch (standard = 4 nodes) supports clusters with up to 16 nodes
- 6.4 Gigabit/sec links (bidirectional) provide high data throughput
- Ultra-low port-to-port packet routing latencies
- High availability clusters supported by hot-pluggable ports, port fencing, and redundant links
- ANSI/IEEE 1596-1992 Scalable Coherent Interface (SCI) compliant
- Non-blocking, Cut-through routing
Scalable Coherent Interface

Node interface

An SCI node receives a steady stream of data and transmit another stream of data. These streams consists of SCI packets and IDLE symbols. A node that is granted interconnect access and that has an empty bypass fifo is allowed to transmit a packet. Contention is solved either by buffering in the interconnection network or by filling the bypass fifo of the next node. SCI uses idles, packet headers, and echos to selectively grant interconnect access under heavy system loading.
Scalable Coherent Interface

Address Map

The PCI-SCI card translates a 32-bit PCI address into a 64-bit SCI address and vice versa. The outgoing translation is based on address translation tables residing in PCI memory. Address translation table entries are cached in the PSB address translation cache. For incoming SCI requests, address protection can be used to disable access to the node or certain PCI memory regions.

In order for a processor to directly access far memory through load/store instructions, the far memory must be addressed through an SCI address. A 64-bit SCI address is used to select nodes and address data within a node. The 64-bit SCI address is split into a 16-bit node id that is used to select the target node. The remaining 48 bits are used within a node to address data and CSR registers.
Scalable Coherent Interface

Address Mapper

During system configuration, driver software on each of the nodes will agree upon (using messages) the desired mappings and set up the contents of the page tables. Each node has a page table that allows that nodes map memory from one or more other nodes. Once the page tables have been configured, the processor can access far memory through load/store instructions.

The address translation is implemented using an address translation table (ATT). The page table consists of 8K entries of 32-bits each. The page size is 512 KBytes.

The information contained in a page table entry are:

- **Node ID**
- **Node Offset Adress**
- **Valid**: Shows if the current entry is valid
- **Atomic** (Lock bit): An access will generate an atomic operation (FETCH_ADD+1)
- **Ordering**: Enforces write before read ordering.