GE Fanuc
Embedded Systems

Real Time Networking with Reflective Memory™

GE FANUC

Designed to fulfill the need for a highly deterministic data communications medium, Reflective Memory™ networks provide a unique benefit to real-time developers. These specialty networks provide the tightly timed performance necessary for all kinds of distributed simulation and industrial control applications. Since their inception, Reflective Memory networks have benefited from advances in general-purpose data networks, but they remain an entirely independent technology, driven by different needs and catering to a different set of users.

A Reflective Memory network is a Real Time Local Area Network that offers unique benefits to the network designer. Reflective Memory has become a de facto standard in demanding applications where determinism, implementation simplicity, and lack of software overhead are key factors. When a designer chooses to use the Reflective Memory architecture, the decision is usually based upon several of the following characteristics:

Performance

- High speed, low latency, data delivery
- Deterministic data transfers for demanding real-time communications

Simplicity

- Easy to implement and transparent to use
- Operating system and processor independent
- It is just memory – Read it and Write it
  - Each networked node has its own local copy of the data
  - Write operations are first performed in local RAM, then automatically broadcast to all other nodes
  - Read operations access the local copy of the data, which always mirrors other local copies on the network

Flexibility

- Drastically reduced software development time and cost (time-to-market)
- Connections to dissimilar computers and bus structures including standard and custom platforms.
- Simple software, low overhead, high noise immunity
- Small to large separation distance between nodes
- Data can be shared regardless of processor type or operating system

This paper discusses distributed multiprocessing scenarios which require demanding off-the-shelf networking solutions for proper performance. Reflective Memory is compared with other communication technologies, and the benefits of the Reflective Memory architecture are discussed in detail.

Part I
Reflective Memory — A Solution for Distributed Computing

Introduction

Networking computers together is a challenging and continually evolving process. Computing processor speeds are doubling every 18 to 24 months, but the common LAN (local area network) technologies that interconnect these machines typically lag behind. This is true for their maximum theoretical bandwidth, and is especially true for their ability to fully utilize this bandwidth.

Clustering

While these limitations are evident in enterprise networks, the limitations of even the most advanced interconnect technologies become acute when dealing with specialized computing systems like clusters of high performance workstations. These clusters may be used to perform distributed computing tasks, utilizing their combined processing power to perform extremely complex computations. With the proper high speed interconnections, a cluster of workstations can equal or exceed the computational performance of very expensive supercomputers; however, historically when an application required a lot of computing power, the answer was just to get a faster computer rather than to wrestle with complex networking difficulties.

Supercomputing

That approach has been carried to an extreme with the design of supercomputers, which operate at clock speeds so high that even the propagation delays through wiring and connectors become limitations. However, the falling cost of consumer-grade 32- and 64-bit microcomputers justifies the economics of multiple processor systems. Distributed processing systems or clustering (up to dozens of microprocessors tightly coupled) can provide supercomputer power at a much lower cost. Unfortunately, when it comes to real-time applications, the response time of distributed systems is often limited by the communication latency of the network that interconnects the microprocessor units. The only way to improve the system response time is to provide a better connection between the microprocessors. By using Reflective Memory systems, designers are able to eliminate most communication latency and realize drastic improvements in resource utilization over traditional LAN technologies.

Strategically Distributed Computer Systems

The use of this distributed computing approach is growing because these systems are economical to build and use, but this approach is more often used because it is practical for other reasons. In supercomputing applications, multiple, similar processors work to solve separate parts of a large problem. Most distributed computing systems differ in that their system applications are broken down into pieces with specialized computers each handling specific independent tasks. This modular approach to the hardware makes even very complicated tasks easier to partition and code. Using multiple distributed computers allows the system designer to partition tasks and to select computers that perform best for certain tasks. The designer is also able to place computers strategically or place them to fit within existing space constraints.

For instance, a rocket engine test stand uses hundreds of transducers to measure various parameters. Operators need a lag-free connection to the testing, but for safety reasons, the instrumentation/viewing center may be located 3,000 meters away. By distributing the implementation, the designer is able to install a computer at the test stand which digitizes and preprocesses the data. Then, instead of hundreds of discrete wires spanning the 3,000 meters, one high speed Reflective Memory network link is all that is required to send the data back to the main computer in the control room. This distant computer then analyzes, archives, formats and displays the data on monitors for viewing by the test operators.

By using a high-speed Reflective Memory link, operators can observe and react to changes as they occur, with minimal delays imposed by the connection. By placing the control staff and core processing computers at a safe distance from the volatile testing, operators are able to minimize risks to personnel and equipment with no degradation of test performance.

Part II
Traditional Methods for Networking Distributed Computers

Of course there are many methods other than Reflective Memory that may be used to transfer messages or large blocks of data between systems, and each method had its own unique capabilities and limitations. Four of the most common methods are examined here.

Overview

The simplest data transfer technique uses Bus Repeaters. These hardware devices actually transfer the CPU read and write signals from one computer to the backplane of another computer. A second technique, Direct Memory Access (DMA), moves data between the global memories of two or more computers. DMA requires backplane control from the local processor to accomplish this. Also considered are message passing via a single shared-global RAM. Finally, standard LANs (like Ethernet and Gigabit Ethernet) will also be examined.

Bus Repeaters

A bus repeater connects the CPU backplane of one computer to the CPU backplane of another computer as shown in Figure 1. This connection allows message passing between CPUs,and also allows each CPU to access resources in the other computer. Since bus transfers may occur at any time and in any direction between computers 1 and 2, a bus arbitration cycle is required on every read or write cycle between the two systems.

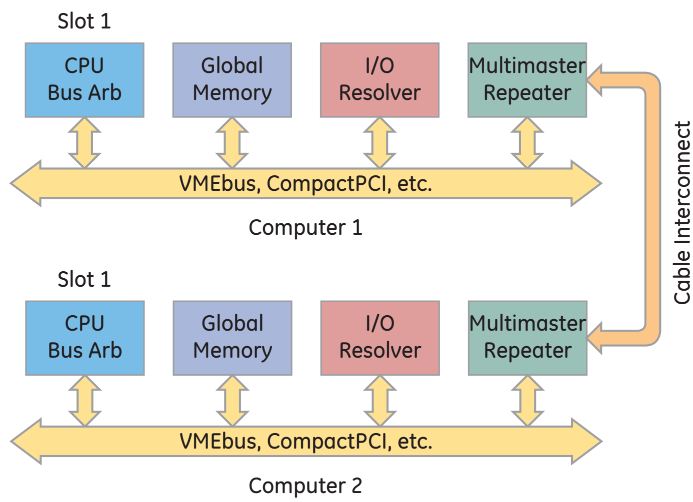

Figure 1: Bus Repeater Connection

The problem with this approach is that each time a CPU wants to access a resource in a remote backplane, it must first request access to the remote backplane, and then wait until that access is granted. Depending on the priority level and type of other bus activity taking place in the remote backplane, this might take anywhere from several microseconds to several milliseconds. This overhead delay not only impedes the data transfer, but it also ties up the requesting backplane, blocking any other activity in this backplane until the remote access is completed. As the systems spend more and more time waiting for each other, the compounded latency delays become prohibitive for real-time applications.

Direct Memory Access (DMA)

Bus repeaters can be very efficient for moving small amounts of data (such as bytes or words) from backplane to backplane. However, in many distributed multiprocessing systems larger amounts of data are exchanged between the various CPUs in the form of parameter blocks. These blocks of data can be moved more efficiently by using DMA controller boards, like the one shown in Figure 2.

In these connections the CPU in each system initializes the address register and the size register on its own DMA controller board. In this process, the address register on the originating DMA controller board indicates where the DMA controller should begin reading the parameter block from global memory. The address register on the destination DMA controller board indicates where the DMA controller should begin storing the parameter block. Once the two CPUs have initialized their respective DMA registers, the transfer is automatic, and the CPUs can direct their attention to other activities.

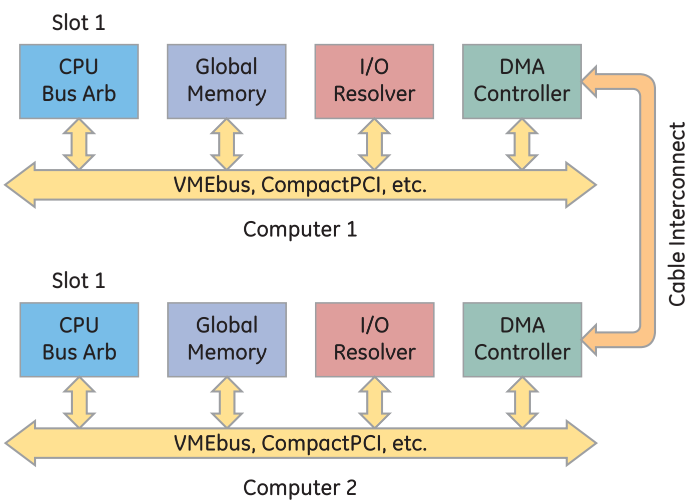

Figure 2: DMA Controller Connection

DMA transfers can occur at a very high rate of speed. That is, once all the above overhead programming and setup has occurred. Both the originating and destination computers must have active involvement in the data transfer. Most importantly, every time a block transfer is completed, both processors must be interrupted so they can reconfigure the DMA controller to prepare for the next transfer. While these DMA transfers are occurring, each local processor must share the available bus bandwidth with its DMA board. This setup can be efficient in certain circumstances, but frequent updates require frequent interrupts which impose latency. Splitting of bus bandwidth between the DMA and the main application can create a data bottleneck for the host application, as well as the DMA process.

Message Passing Via Shared (Global) Memory

A third configuration would be for two or more computers to share a single set of global memory as in Figure 3. A typical shared global memory scenario would be two or more computers residing in the same backplane-based chassis (usually VMEbus or CompactPCI). Each of these computers would have their own local memory where accesses occur at the full speed of the processor. The computers could then communicate and share data with each other via a global memory set resident in the same backplane by utilizing a pre-established message protocol scheme.

In this type of system, the global memory is basically a single-ported memory shared among several computers and, while it may be accessible to all computers residing within the same chassis, access to this resource must be arbitrated. Also, inter-processor communications occur at the speed of the bus-memory card combination, which is typically much slower than accessing local memory. The individual computers end up competing for the one scarce resource that facilitates the sharing of information and even when a processor has free access to the shared memory, it is at a lowered speed.

The communication becomes more cumbersome when externally distributed computers are connected into the single-ported global memory via repeaters, DMAs, or LANs. The total data latency may become compounded as each processor must wait its turn to access the memory (both in writing in new data and in receiving messages from other computers via the global memory). In this scenario data latency (which can be broadly defined as the time it takes before all computers can gain access to new data) can quickly spiral out of control.

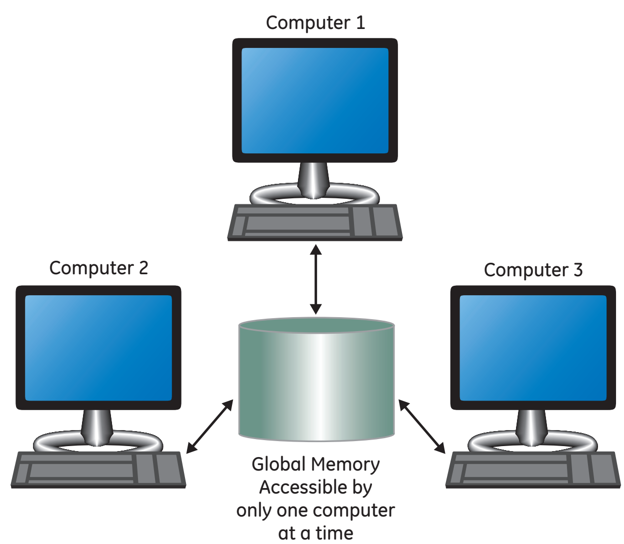

Figure 3: Global Shared Memory Architecture.
Only one node at a time may access the shared memory.

Traditional Local Area Networks (LANs)

The most familiar method of sharing data between computers is to rely on conventional networking standards such as 10/100 or Gigabit Ethernet. With this approach, the computers may be physically separated and connected via a network, and a common database is maintained from data sent through that standard network. This allows for wider connectivity and a more standardized communications approach, but adds considerable overhead for data transmissions. Also, because of Ethernet's arbitration schemes, determinism (the ability to define a time window in which communication will become available at a specific place on the network) is lost.

The communication overhead of a LAN protocol like Ethernet adds another layer of complexity while decreasing the usable data payload. Once a system grows beyond a few nodes, that overhead can outweigh the advantage provided by the shared memory scheme. Like the other examples, this is still a single-ported memory approach and only one node may update the database at any one time. While LAN technologies enable developers to distribute their systems, they do not address the bottleneck of accessing the single-ported memory, which is still essentially an arbitrated resource.

In summary, standard LANs have several shortcomings when real time communication is required:

- Transfer rates are low.
- Data latency is hard to predict, and is typically too large for real time distributed multiprocessing systems.
- Layered protocol software consumes too much valuable processor time.

Part III
Reflective Memory — As a Networking Solution

As stated above, a Reflective Memory network is a special type of shared memory system, designed to enable multiple, separate computers to share a common set of data. Unlike global shared memory systems, in which individual systems vie to access a single, centralized data set, reflective memory networks place an independent copy of the entire shared memory set in each attached system as shown in Figure 4. Each attached system has full, unrestricted rights to access and change this set of local data at the full speed of writing to local memory.

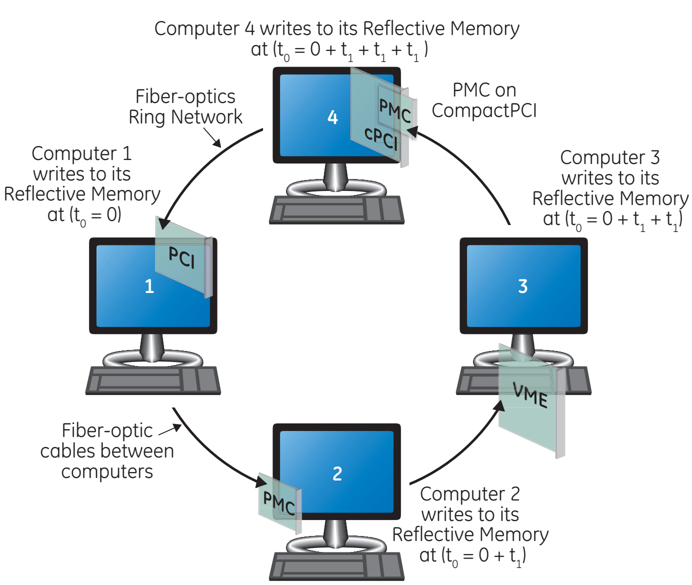

Figure 4: Reflective Memory for Real-Time Networks.
The 5565 series provides very low latency between network nodes.

Each Reflective Memory board (node) has local memory, an embedded interface, and arbitration logic that enables both the host computer and the Reflective Memory network to access it.

When data is written to the local copy of Reflective Memory, high speed logic simultaneously sends it to the next node on the ring network as illustrated in Figure 5. Each subsequent node simultaneously writes this new data to its local copy and sends it on to the next node on the ring. When the message arrives back at the originating node, it is removed from the network and, depending on the product and number of nodes, every computer on the network has the same data at the same address within a few microseconds. Local processors can read this data at any time without a network access. In this scheme each computer always has an up-to-date local copy of the shared memory set.

The Reflective Memory boards may be physically installed or connected to a variety of computer backplanes, including VME, and PCI/PCI-X, as well as CompactPCI or any standardized or proprietary system capable of hosting a PMC site. This allows most popular workstations and single board computers to be connected via Reflective Memory regardless of their interoperability.

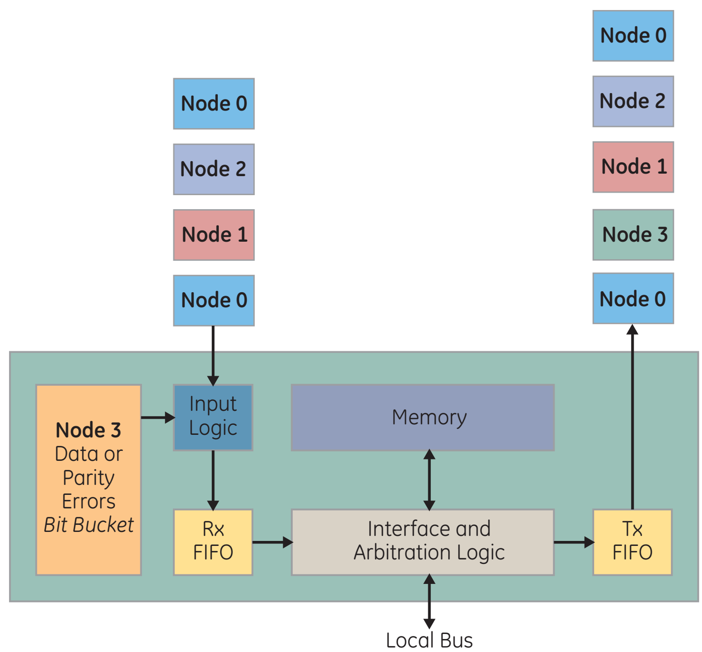

Figure 5: Reflective Memory Data Insertion.
Data from the network is automatically written to local memory and transmitted on to the next network node by embedded logic.

"Where Do You Use Reflective Memory?"

Reflective Memory may be used in any application that uses Ethernet, Fibre Channel, or other serial networks to connect computers or PLCs together but it is not ideal for all applications. Reflective Memory is most relevant in systems where interaction in true real-time is a primary concern. In systems where determinism, low latency, and high-speed communication are necessary, Reflective Memory boards, while typically more expensive than lower performance hardware, provide a huge return in performance with the added benefit of ease of use.

"How Do You Use Reflective Memory?"
Using a Reflective Memory network requires only a few simple steps:

- Plug Reflective Memory into any available backplane slot (VME, PCI, etc.), or connect it to any single board computer or carrier (VME, CompactPCI etc.) with a PMC site and connect cabling.
- Write to memory (Reflective Memory's global memory appears to the computer as standard RAM).
- Read memory (any Reflective Memory board on the network).

The benefit of a low software, high speed,, hardware-driven network like Reflective Memory is extremely low data latency, both overall and between individual network nodes. This low-latency performance is of paramount importance when building real-time systems such as simulators.

Simulation: An Acute Application Example
One example that shows the importance of low-latency performance is an interactive combat flight simulator where multiple participants participate as independent variables in a combined synthetic environment. In this type of simulation each separate computing system may be responsible for generating a display, managing individual participant's inputs into the simulation, generating terrain or other environmental aspects, managing weapons systems, or any of a variety of other functions. Because multiple independent participants are performing highly-dynamic movements at extremely high speeds, it becomes imperative that the system updates frequently enough to represent an accurate and lifelike simulation of reality.

In the GE Fanuc Embedded Systems 5565 series Reflective Memory network, any memory write placed on the Reflective Memory network, is available for the next computer to receive as data in 700 nanoseconds. This is measured from the time it was written into local RAM and transmitted to the next Reflective Memory board. In the 4 node example provided in Figure 4, it takes 2.1 µs for all computers to receive the data that was written to Reflective Memory.

Let us assume that two simulated F-22 aircraft are traveling at their cruising speed of 1,070 mph. If one pilot in the simulation makes a radical maneuver which forces the other pilot to take evasive action, it must be quickly represented in the simulation or the realism of the simulation will be lost.

By minimizing update latency, Reflective Memory ensures the integrity of the simulation. The speed of the network enables the simulation to withstand the scrutiny of the sensory perceptions of the human participants and it meets higher requirements for serving electronic "participants" such as each aircraft's weapons, control and navigation systems.

Part IV
"Why Would A Network Designer Choose Reflective Memory?"

Reflective Memory LANs or Real Time Networks are usually constructed because the designer has needs or problems that are solved by one or more of the following Reflective Memory board characteristics:

- Deterministic data transfers
- High speed performance
- Ease of use
- Operating system and processor independence
- Economics and available time-to-build systems

Deterministic Data Transfers
Reflective Memory is a hardware-based network. All data transfered to a node is stored in local memory and automatically sequenced out to all the other nodes' memory. There are no software delays and minimal hardware delays associated in the data transfer. Any latency is imposed at the hardware level and can be predetermined within a very small window of best-to-worst case latency.

The determinism of Reflective Memory — the guaranteed time in which communication between two or more nodes is completed — allows system designers to build effective real-time LANs that can guarantee data delivery within a tight window of time. This enables guaranteed scheduling of sequential actions and ensures that data is not lost.

High Speed Performance
The demands of real-time performance dictate that Reflective Memory networks must operate at very high speeds. In some cases, this speed and the throughput it generates are compelling reasons to use Reflective Memory. As an example, the 5565 series from GE Fanuc Embedded Systems offers dynamic packet sizing (4 to 64 bytes), DMA capability for minimal CPU overhead, full PCI (64-bit/66 MHz) compliance with PCI Rev. 2.1, up to 128 Mbyte DRAM, 2.12 Gbaud fiber optic connections, and a sustained 170 Mbyte/s transfer rate. This compares favorably for both speed and throughput against most data networks.

Ease of Use
No high performance LAN is as easy to install and operate as Reflective Memory. An ideal network would allow every computer simultaneous access to every computer's memory. Reflective Memory meets this ideal by giving every computer on the network an actual copy of all the other computer's memories and, after network latencies measured in miocroseconds, it is available for immediate access. Reflective Memory allows this to occur in up to 256 distinct computers.

Since this memory is global, accesses to the memory space may be simultaneous by multiple computers. All CPU write accesses to this common memory space are replicated to all nodes in the network. Reflective Memory transparently monitors and replicates this data, so that the application(s) can share the data without software overhead penalties.

Operating System and Processor Independence

Reflective Memory hardware is currently available for VME, PCI/PCI-X, PMC, and various other form factors. This allows dissimilar buses to be connected in a single Reflective Memory network. An embedded single board computer (VME or CompactPCI) with a PMC slot may use a Reflective Memory PMC board to offload traffic on the host backplane. Users are able to configure a high speed network where any desktop workstation, single board computer, or server can share information directly with ANY computer having an available CompactPCI, PCI/PCI-X, VMEbus slot, or a PMC site.

Even when connecting different computer types which use different byte formatting (big- and little-endian types), byte swapping is a non-issue in Reflective Memory systems. PCI-based Reflective Memory boards can include hardware designed specifically to translate the language of byte swapping. This hardware provides quick, efficient and repeatable bidirectional conversion. Again, no protocol overhead or time penalties are incurred for conversion between big- and little-endian types.

Reflective Memory has the same ease of use regardless of the operating system or machine it is used in.

Economics and Available Time-to-Build Systems

System designers are increasingly asked to build more capable, complex systems in a shorter amount of time. In such cases, the cost of hardware may be minimal when compared to software and the time spent integrating hardware/software. This is especially true of one-of-a-kind or small-run systems. In cases where time to market is a critical dimension, the lower investment in software person-hours required to get a system up and running when using Reflective Memory may be an extreme benefit. In these demanding systems, Reflective Memory's simple read/write method of communication substantially improves time to market, as well as improving data throughput on the network.

Comparison to Standard LAN Technologies

Reflective Memory offers a variety of features over standard networks. Features such as global memory, high speed data transfers, and software transparency make Reflective Memory an easy-to-use but powerful solution to multicomputer communication. When compared to the costs associated with additional development time, testing, maintenance, documentation and extra CPU requirements of traditional communication approaches, Reflective Memory provides a cost-effective alternative. Table 1 is a summary of comparison characteristics for Reflective Memory, Ethernet, and Gigabit Ethernet.

Table 1. Comparison of Off-the-Shelf Networking Technologies

Characteristics	Reflective Memory Network (5565 Product Line)	10/100 Ethernet	Gigabit Ethernet
Transmission Speed	2.1 Gbaud/s	10/100 Mbit/s	1000 Mbit/s
Data Transfer Speed	170 MB/s	1/10 MB/s	100 MB/s
Endian Data Conversion	Yes	No	No
Software Transparent	Yes	No	No
Media	Fiber-Optic	Coax, UTP	Fiber-Optic
Topology	Ring	Ring, Hub	Ring
Network Data Transmission/ Reception Is Deterministic?	Yes	No	NO
Network Transfer Scheme	Data Insertion	Carrier Sense Multiple Access/ Collision Detect	Token Passing
Memory Mapped Access to Shared Data?	Yes	No - Messaging Application Must Be Built	No - Messaging Application Must Be Built
Application Must Be Constructed to Share Data?	No	Yes - Messaging Application	Yes - Messaging Application
Application Must Encode/ Decode Messages?	No	Yes	Yes
Application Must Perform Error Check/Handling Retransmits, etc.?	No	Yes	Yes
CPU Overhead to Support Shared Data Functionality?	No	Yes	Yes
CPU Overhead Required at Transmission Hardware Interface?	No	Yes	Yes

Part V
"Who Uses Reflective Memory?"

Reflective Memory has been used in hundreds of applications to network computers together. Any application where the designer desires (or requires) high performance or ease of use (or both), Reflective Memory is ideal for the designer. Reflective Memory is currently operating in the following applications:

- Aircraft simulators
- Automated testing systems
- Ship and submarine simulators
- Aluminum rolling mill control/monitoring
- Power plant simulators
- Engine test stands
- Industrial process control
- High speed data acquisition
- Over-the-horizon radar
- PLC users

The following pages examine two Reflective Memory application examples. The first is an example that highlights Reflective Memory's ease of use. The second example is a real life application that demonstrates the high speed performance associated with a Reflective Memory communication system.

Part VI
Application Examples

Networking Two Computers Together Using Traditional Components (Ethernet or Gigabit Ethernet)

The following example shows the process required to share data between two computers. These steps would hold true for regular Ethernet, as well as Gigabit Ethernet LANs.

In this example, Computer A collects raw data samples from ten different types of sensors. With 20 sensors of each type, there are a total of 200 sensors. This data is stored in computer A's own memory, then transferred to computer B for processing and display via a Graphical User Interface (GUI).

Gigabit Ethernet Example

1. Computer A collects the data for each sensor type at different intervals; therefore, it does not send a fixed format data stream of all the data since this is too inefficient. Instead, Computer A sends the data to Computer B by sensor type. To accomplish this, Computer A must include the sensor type and number (1-20) with the sensor's data so that Computer B knows how to process the incoming data.
2. Between these two computers, there must be an application that encodes and decodes these sensor type/number/data messages. It is clear that Computer A would have to know how to encode ten different types of messages, one for each type of sensor's data, and Computer B would have to know how to decode ten different messages. Computer B would then act on the contents of those messages.
3. Computer A, after constructing a sensor type/number/data message, must transmit that message to Computer B. It does this by relying on the network hardware and the hardware's driver software. Computer A passes these constructed messages to the network adapter. The network adapter then reformats this message into data packets for transmitting through the network. The adapter hardware has to add information such as routing addresses, error checking information, and other networking protocols so that the receiving hardware interface on Computer B gets the information and can check its validity.
4. Upon receiving the information, the network adapter hardware in Computer B reads and interprets the data packets to verify that the packets arrived intact and error free. The hardware adapter then notifies the computer that the transmission data is ready to be placed in memory. Computer B then decodes this type/number/data message constructed by Computer A. Computer B must decode this message to separate the sensor type and sensor number from the actual sensor's data.
5. Computer B determines the sensor type and number through various case or case-like statements to determine which particular sensor type this message contains, and it must also determine which of the 20 sensors the data came from. After this information is extracted, then and only then, can the actual data originating from Computer A be written into the memory of Computer B so that the actual processing of this data may begin.

Same Example with Reflective Memory Real Time Network

The same example implemented with Reflective Memory:

1. Computer A places the raw data from each of the sensors into the memory on its Reflective Memory board. Each sensor has its own unique address within the memory.
2. Reflective Memory automatically replicates this data to Computer B's Reflective Memory board.
3. Computer B now has the data available in its local memory, and may begin processing this data.

Reflective Memory is Used as a PLC Accelerator™ in an Aluminum Rolling Mill

The Reflective Memory real time network (GE Fanuc Embedded Systems 5565 series) was recently used to enhance a PLC-controlled Aluminum Rolling Mill operation. Distributed GE Fanuc Automation Programmable Logic Controllers (PLCs) were used to monitor and control a 3,500 ft/min rolling mill. The PLC Control loop had a resolution response that allowed 2 to 3 feet of aluminum to pass through before actuators could respond.

These actuators were responding to apply/release pressure on the aluminum to vary the thickness.

Using Reflective Memory products from GE Fanuc Embedded Systems, a system of communication between separate VME-based systems was developed to reduce this response time. The 2 to 3 foot resolution was reduced to 4 inches, resulting in tremendous waste reduction and quality improvement in the final product. Data relating to the mill was imported to the PLC which immediately wrote it into Reflective Memory's memory (Figure 6), thereby sending it to a separate VME computer system to transfer complex control algorithms. The system sends the output control data back to the PLC by simple write commands to its Reflective Memory. In this application, data appears in local Reflective Memory in an almost instantaneous fashion on the PLC. The data transfers are fast and the computations by the PLC Accelerator are so fast that the PLC runs its control loop with no delays. Thus, the PLC operates at its maximum scan rate because it is only reading/writing the plant's control devices and Reflective Memory's memory.

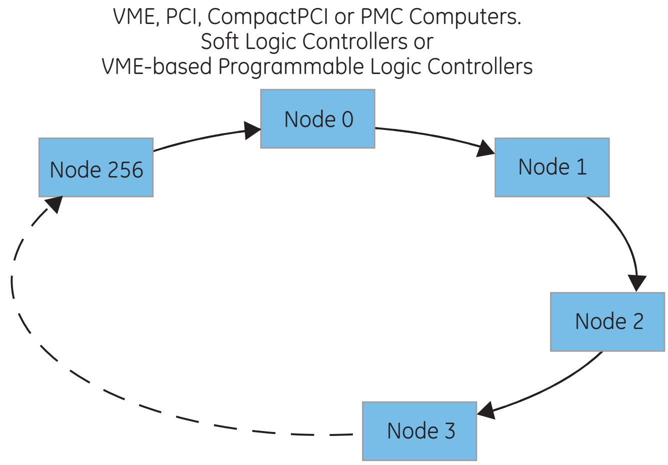

Figure 6. Reflective Memory Ring Architecture connects up to 256 separate network nodes in real time

Part VII
Reflective Memory Implementation

Ring Architecture Network — No Data Collisions

GE Fanuc Embedded Systems' 5565 product line provides a data-insertion, ring-architecture network operating at rates of 2.12 Gigabaud over a fiber-optic ring. Because Reflective Memory is not a collision-based bus arbitration system as most Ethernet systems are, it avoids the complexities required for queuing and checking data packets.

The Reflective Memory ring ensures proper connectivity and does not impose additional loading restrictions or termination requirements. Distance between nodes may be up to 10 km. These attributes allow data transfer rates of 170 MB/s for Reflective Memory, compared to about 100 MB/s for Gigabit Ethernet (excluding protocol overheads).

Interrupt Capability Allows Data Synchronization Between Nodes

Reflective Memory allows any network node to interrupt any other node or all nodes. Up to four different interrupts may be assigned and are user-definable per interrupt as to function, priority, and vector. These interrupts may be used for any function, such as mailbox interrupts or data synchronization across the network. When enabled, interrupts typically are used to interrupt one or all the nodes on the network after data has been transferred.

The use of interrupts is not required. Reflective Memory powers up with its interrupts disabled. This allows Reflective Memory to be software transparent and not require driver software to be loaded unless interrupts are needed.

Bus Support

For high performance applications, GE Fanuc Embedded Systems offers a 170 Mbyte/s product line, the 5565 series. The 5565 series is available for the VMEbus, PMC, and PCI bus available in most modern PCs and workstations and CompactPCI systems.. The 5565 series supports two connectivity options to meet economic and performance needs:

Data Transfers

The fiber-optic Reflective Memory communication network is a fiber-optic ring as shown in Figure 6. Data to be transferred is placed in packets with other system information and passed from node to node. Each node stores the data received from the previous node then retransmits the data in a packet to the next node in the network. Data insertion by any node may occur at any time.

Packets of data flow around the ring in a stream passing through each node. Within each node, the packets are unpacked, checked for errors, and the data stored in the Reflective Memory. The packets are then reformed and passed to the next node. If data is written to the Reflective Memory by a local CPU or DMA device, then new packets are formed and inserted into the stream of packets flowing through the node.

When a data packet has completed the ring and returns to the originating node, it is recognized and removed from the network. The result is a highly efficient real-time LAN with minimal hardware delays to complete data transmission to all the computer memories in the network.

Interrupt Transfers

In addition to transferring data between nodes, Reflective Memory will allow any node to generate an interrupt on any other node. These interrupts would generally be used to indicate to the receiving node that new data has been sent and is ready for processing. These interrupts are also used to indicate that processing of old data is completed, and the receiving chassis is ready for new data.

Error Management Techniques Provide Superior Data Integrity

GE Fanuc Embedded Systems Reflective Memory boards have extensive error detection and notification facilities.

The error rate of the network is a function of the rate of errors produced in the optical portion of the system. This optical error rate depends on the length and type of fiber-optic cable. When a node detects an error, the erroneous transfer is removed from the system and an interrupt is generated, if armed.

For systems where even this minuscule error potential is unacceptable, Reflective Memory can be operated in a redundant transfer mode in which each transfer is transmitted twice. In this mode of operation, the first of the two transfers is used unless an error is detected, in which case the second transfer is used. If an error is detected in both transfers, the node removes the transfer from the system.

The fiber-optic Reflective Memory boards also employ a network monitor bit that can be used to verify that the data has properly traversed the ring. This bit can also be monitored to measure network latency (the time for data to be transferred and stored in all computer memories on the network).

Automatic Fiber-Optic Bypass Switches Enhance Network Reliability

GE Fanuc Embedded Systems produces a fiber-optic switch that enables a Reflective Memory network to continue operating if any node fails. These products automatically bypass a network node if it ceases operating to ensure that a node failure does not crash an entire network.

Operating Systems

With driver support for various operating systems, GE Fanuc Embedded Systems installation programs allow the user to select from various processor/OS/hardware combinations. Please contact GE Fanuc Embedded Systems for the latest list of available drivers and operating systems supported.

Summary

Reflective Memory is an optimal way to share data in time-critical applications ranging from data acquisition and process control to advanced simulation. Reflective Memory networks provide a real-time networking capability that surpasses most communications technologies for low-latency and deterministic performance. Reflective Memory networks connect these systems with minimal update delays and no access restrictions, to enable multiple, remotely located nodes to share a single data set in real time.

GE Fanuc Embedded Systems is dedicated to Real-Time Networking and has developed a complete solution to connect most nodes together in real-time. GE Fanuc Embedded Systems' Reflective Memory product line offers deterministic performance which is independent of the operating system or processor used.

Read memory, write memory, it's that simple.

GE Fanuc Embedded Systems Information Centers

Americas:
1 800 322 3616 or 1 256 880 0444

Asia Pacific:
86 10 6561 1561

Europe, Middle East and Africa:
+49 821 5034-0

Additional Resources

For more information, please visit the GE Fanuc Embedded Systems web site at:

www.gefanucembedded.com

01.07 GFT-630