SIMULATOR COST REDUCTION USING A DISTRIBUTED I/O AND DISTRIBUTED POWER ARCHITECTURE

John McCrillis, Kevin Cahill, Joseph Kerkes, Clint Steffan
BAE SYSTEMS Flight Simulation and Training
Tampa, FL

Abstract

A typical full flight simulator has 1,200 or more Input/Output (I/O) points, comprised of a combination of digital inputs (DIs), digital outputs (DOs), analog inputs (AIs), analog outputs (AOs), synchros, and resolvers. In a conventional I/O system, discrete wires are run to each I/O point through a series of connectors, cables, and distribution panels before arriving at an I/O cabinet where the signal is digitized and transferred via high-speed bus to the host computer. The system is complex and is a bottleneck for both hardware development and hardware/software integration (HSI). For example, when several panels share a common distribution panel, every panel must have its wire assignments complete before the distribution panel can be complete and HSI cannot begin until all the connected systems are ready to be powered. To eliminate these bottlenecks and to reduce the overall complexity of the system, BAE SYSTEMS has investigated various embedded computer system products for decentralizing the I/O system. We have concluded that commercially available products can effectively meet the computational and networking requirements of a distributed I/O system for flight simulators and do so at lower cost than the typical centralized system. However, further analysis of this concept revealed additional savings by decentralizing the power distribution as well. The result was the development of a unique I/O-Power module that fundamentally alters simulator development. This paper discusses the analysis and development work conducted to draw this conclusion.

Biographical Sketches:

John McCrillis is currently a Project Engineer for BAE SYSTEMS, having worked in the simulation industry for 16 years. He has simulator experience with V-22, UH-60, OH-58, AH-64, F/A-18, and F-14 simulators. He has a BSAE (‘78 Cal Poly) and an MSME (‘84 Purdue).

Kevin Cahill is the Director of Research and Development at BAE SYSTEMS Flight Simulation and Training. He has held a number of positions over the past 15 years including Electronics Engineer, Project Engineer, and various engineering management positions. He has published a paper on digital communications at I/ITSEC '89. He has a BSEE ('86), an MSEE ('92) and an MSEM ('00) from the University of South Florida.

Joe Kerkes is a Senior Computer Systems Engineer with BAE SYSTEMS for the past 18 years, specializing in embedded system applications and design.

Clint Steffan is Senior Electronic Design Engineer at BAE SYSTEMS. He received a BSEE from University of South Florida in 1984 and taught electronics at Tampa Technical Institute for two years before taking a design position at BAE SYSTEMS, where he has been for the past 15 years. His design work includes several RADAR simulation interfaces, as well as a number of custom 'black-box' interfaces.

SIMULATOR COST REDUCTION USING A DISTRIBUTED I/O AND DISTRIBUTED POWER ARCHITECTURE

John McCrillis, Kevin Cahill, Joseph Kerkes, Clint Steffan
BAE SYSTEMS Flight Simulation and Training
Tampa, FL

CURRENT WIRING METHODS

Full flight simulators make extensive use of computer input/output (I/O) to interface the host computer to the cockpit and other major simulator systems. The typical trainer can have more than 1,200 I/O points comprised of digital inputs (DIs), digital outputs (DOs), analog inputs (AIs), analog outputs (AOs), synchros, resolvers, and other specialized signal converters. In a conventional I/O system discrete wires run from each I/O point through a series of connectors, cables, and distribution boxes or trays before arriving at the centralized I/O system where the signals are digitized and transferred to the host computer. The I/O system is often based on a VME buss, comprising of one or more 21-slot chassis.

The typical VME chassis is populated with a CPU and disk drive for data packaging, reflective memory for communication with the host computer, and specialized I/O cards for signal digitization. Signal information is passed through numerous connections before reaching the I/O point. The connections are cumbersome to troubleshoot should there be a problem, and extensive cabling requires numerous drawings, all of which must accurately represent the system as built. One benefit however is the accessibility of the I/O boards. The VME rack allows easy access for test and replacement of boards.

Distribution boxes or distribution trays like the hypothetical one shown in Figure 1 are a necessity because most cockpit panels need a mix of a few different I/O types and most I/O boards are high density of a single I/O type. The term *panel* is used generically in this paper to mean any self-contained component with I/O requirements. It may be an aircraft panel, instrument, IOS control, gauge cluster, circuit breaker panel, or other control and monitoring device. Figure 1 makes it apparent that any change to the I/O distribution system must also address power distribution and light dimming, both of which take advantage of the distribution box or distribution tray.

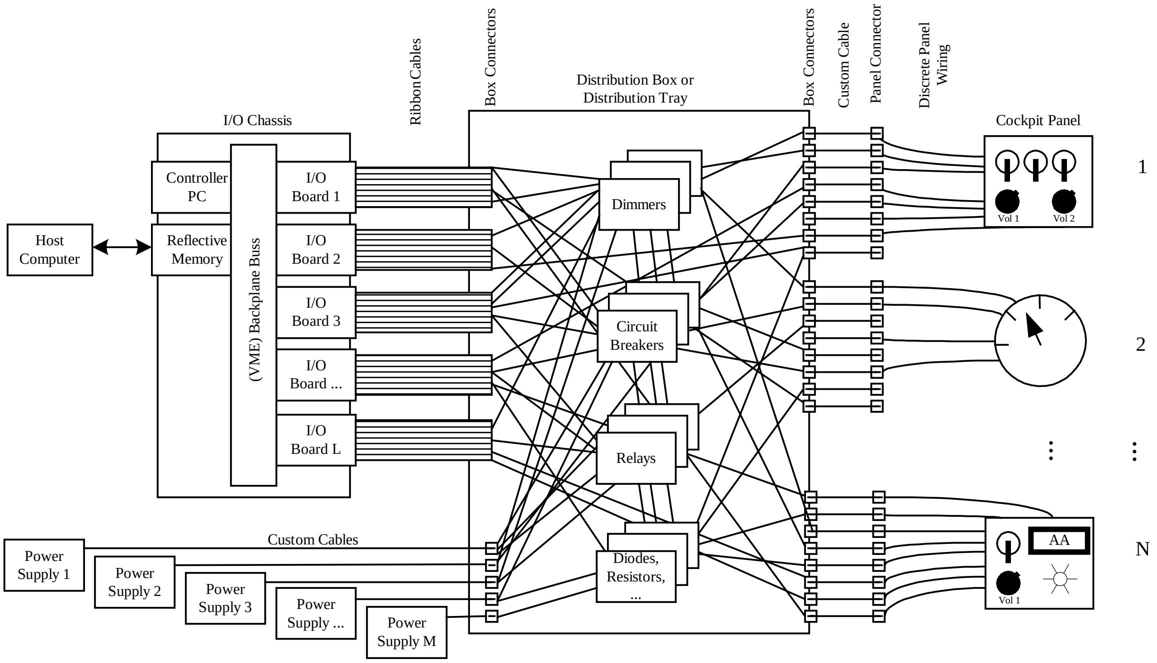

Figure 1. Centralized I/O System

Distribution boxes like the one shown in Figure 2 are used to accomplish I/O and power distribution. A typical simulator requires four distribution boxes and a series of power supply trays. The distribution boxes and power supply trays normally fill the perimeter of the motion base frame as shown in Figure 3.

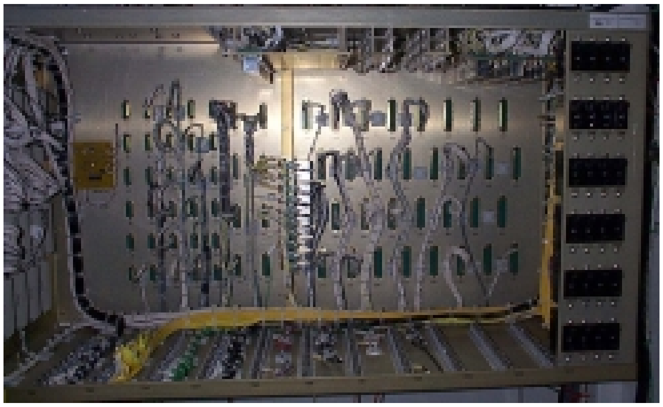

Figure 2. Distribution Box

The distribution box concept enables centralized I/O and power to be distributed as needed to each simulator panel. It also provides a location to mount auxiliary hardware such as light dimmers, power interruption relays, and simple signal processing components. Disavantages of the distribution box include electrical engineering design and documenting costs, and recurring manufacturing costs. The design schedule is also hampered by the bottleneck that requires connected systems to be completely designed before any drawings can be released. Additionally, production is impeded because nothing can be powered until everything passing through distribution box is ready to receive power. As a result, testing cannot start until everything is ready to be powered, delaying hardware/software integration (HSI).

DISTRIBUTED I/O OBJECTIVE

The objective of this project was to analyze the cost impact of replacing the current centralized acquisition system with a distributed I/O system.

REQUIREMENTS ANALYSIS

Panel requirements were derived from an analysis of three aircraft: a Boeing 737-300, a Kaman SH-2G, and a Bell AH-1F. These cockpits were chosen because of data availability and they represented three different airframe manufacturers, giving a diverse perspective of aircraft wiring. The cockpits were analyzed on a per panel basis for I/O type, I/O count, and power requirements. Each panel was equated to a standard 5.75-inch panel width and the lighting wattage normalized to watts/inch for the standard width panel. Results of the study are summarized in Table 1. The requirements analysis quickly lead to a list of important considerations, all of which are discussed further on in this paper:

- Node resolution
- Accessibility
- Light dimming
- Power distribution
- Commercial vs. custom I/O
- Mechanical mounting

Node Resolution

A node is a data acquisition point in the simulator that transmits digitized signals via a high-speed bus. In general, this comprises a CPU (although non-intelligent nodes were considered), Ethernet connection (other busses were evaluated), all the necessary I/O digitization hardware, and power. The node resolution issue drives many other design decisions. Table 2 classifies three broad classes of possible node resolution.

The low-resolution solution is the traditional one node, centralized I/O approach and is the baseline which cost and performance were compared. The medium resolution approach divides the simulator into a number of zones, generally at the same resolution as the natural clustering of panels and instruments in the cockpit. The high-resolution approach has a data acquisition node at each panel. The medium and high-resolution configurations were analyzed with respect to the baseline assuming the panel requirements derived from the aircraft studied.

Figure 3. Distribution Boxes in Baseframe

Table 1. Panel Analysis Summary

	Area	Height for a standard 5.75" width	DI	Source DO	DO	AI	AO	Edge lighting power
	Inches^2	Inches	#	#	#	#	#	Watts/ Panel Inch
Min/Panel	3	0.57	1	1	1	1	1	0.26
Max/Panel	79	13.81	91	9	92	38	11	2.46
Avg/Panel	17.12	2.98	14.63	4.40	7.91	2.73	2.30	1.79
Median	13.87	2.41	8.00	4.00	3.50	1.50	2.00	1.90
Std Dev	13.89	2.42	19.66	3.21	12.71	5.57	2.03	0.49
Cockpit	1387	241	1609	22	585	120	115	0.00
Design objective		5	16	4	16	4	4	10

Table 2. I/O Node Resolutions

Resolution	Typical Node	Nodes per Simulator	Example Nodes	Comment
Low	I/O Chassis	1	I/O Chassis	Traditional centralized I/O
Medium	Console	6-12	Center Console, IOS, Pilot Instruments, Copilot Instruments, Left Console, etc.	Intermediate solution
High	Panel	40-80	Comm1 Panel, Comm2 Panel, Nav1 Panel, HSI Instrument, Fuel Gauges, Circuit Breaker Panel, etc.	One node per panel

Medium Resolution Nodes (Console Level)

Initially, about ten medium resolution nodes seemed to be the best solution since it matched the natural panel clusters of the simulator. A data acquisition system would be mounted near the node and cables run to each panel as required. However, when power and light dimming requirements were evaluated, the same distribution problems associated with centralized I/O occurred, only on a smaller scale. There was no reduction in the number of cables and connections, only their length. Although there would be some bottleneck improvements with design release and testing, ten small acquisition systems was not a significant cost improvement over centralized systems.

High Resolution Nodes (Panel Level)

With about 75 panels in each simulator, it seemed that a data acquisition node at every panel would be wasteful and mounting all this additional hardware would be difficult. Some panels only had a few switches and lights and were physically very small. How could an acquisition system at each panel be cost-effective to control so few features? Due to the plummeting costs of CPUs, it is now more economically feasible than ever before to implement distributed I/O. Dollar per I/O point, PC hardware is less costly than hardware currently available for the VME bus. But there is still a significant amount of non-recurring engineering (NRE) for mounting the acquisition system hardware.

Another possibility that had not been explored to offset this additional cost was distributing the power as well as the I/O. Although distribution of power was not originally considered, it soon became the dominant issue and offered greater savings potential than distribution of I/O alone. With smaller nodes, power requirements can be met with electronic components mounted on a board. A single specialized electronics board could potentially provide all the power requirements for all nodes in an entire simulator, eliminating single source power supplies and electrical distribution boxes. The power board would receive power from a standard 28 VDC source and create all other node voltages onboard. Wiring a cockpit would be reduced to plugging a 28 VDC and bus cable to a couple of cards mounted on the panel. Discrete wiring would be done solely within the panel at a bench. The NRE for mounting the board was still an additional cost but now we were eliminating even more NRE for a net cost savings that was significant. The key to making this concept succeed was identifying the proper voltages and wattages so the power board(s) did not have to be designed for each program and could be purchased in bulk.

PROPOSED SOLUTION

Acquisition System Hardware Selection

Before a power board could be designed, the type of I/O acquisition system had to be selected. From the beginning, it was assumed that PCs would probably meet our needs. We made this assumption because the market is very large and extremely competitive, creating a broad selection of hardware and software at low prices. We investigated STD, VME, ISA, PCI, compact PCI, EISA, PC/104, and Multibus formats before selecting PC/104. The selection was based on hardware cost/I/O point, form factor, hardware and software availability, and market size.

Although we have gone forward with the PC/104 format for the distributed I/O project, conceptually, we are not tied to this format. Should a case arise where a compact PCI or other format serves a requirement better than PC/104, the distributed I/O architecture supports this alternative without redesign. For example, a 19-inch rack mount PC filled with standard PC ISA DI boards may be used as a node to read the state of circuit breakers should it be more cost effective. This flexibility is a major feature of the distributed I/O architecture that we hope will not get lost in the PC/104 details discussed throughout the remainder of this paper.

PC/104

The PC/104 standard is a full PC CPU on an approximately 3.5-inch square circuit board. A unique self-stacking connection system allows the ISA bus to extend through board connectors instead of a backplane and card cage. CPU, I/O, and other boards can be stacked together to create a truly modular system that is extremely space efficient as shown in Figure 4. The PC/104 form factor fits between standard aircraft panel support rails, allowing us to mount the acquisition hardware directly to the panel. The hardware can be mounted either vertically or horizontally on a bracket and can be enclosed or not. Additionally, no special cooling is required, since most commercial PC/104 hardware is designed to operate at 80ºC. Access to the hardware is gained by removing the aircraft panel from the console by its face, a highly desirable feature from a test and maintenance perspective.

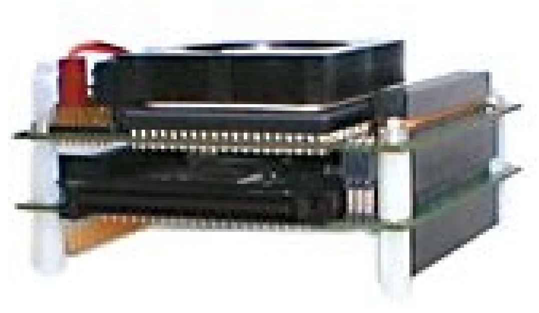

Figure 4. PC/104 Stack

PC/104 is in wide use with numerous applications, primarily in the embedded systems market. Consequently there are vast quantities of hardware to select from such as power supplies, CPUs, buses (Ethernet, ARINC, 1553, etc.), I/O cards (DI, DO, AI, AO, synchro, resolver), video, and others. A three-card stack with a power, CPU/Ethernet, and I/O card can be created solely from commercial products to meet most of the needs of the trainer community. The disadvantage is a fully commercial stack would probably be three or more cards and it would probably require a unique combination of boards for each node as well as unique cabling. Unique components translate directly to NRE cost. To address this issue we began investigating the development of a power board that could become a standard designers could use without having node development cost. To conserve space, we proposed putting some I/O on the power board and thus we began developing a unique power board that included a standard set of I/O as well.

Power Considerations

Most aircraft panels require power for internal operation and dimmable lighting. Additionally, the PC/104 stack needs power. As part of the I/O analysis, power requirements were analyzed to identify power board requirements. The results of this analysis are summarized in Table 3 for a five-inch panel with a three level PC/104 stack. Experience suggests that aircraft lighting is fairly uniform and can be readily normalized to watts/inch**2. For larger panels where more power is required, additional power boards can be added to the stack. This feature is consistent with panel sizing, bigger stacks equate to bigger panels.

Table 3. Five-inch Panel Worst Case Power Requirements

Component	Voltage		Watts	Amps
Edge lighting	Variable	0-5 or 0-28 VDC	10 (2watts/in)	2, 0.5
Indicator lamps (24 bulbs)	Variable	0-5 or 0-28 VDC	27 (40 ma/bulb @ 28V)	5.5, 1.
EL panels	Variable	115VAC 400 Hz	-	0.02
LCD lamps	Variable	+5 VDC	0.025	0.005
Internal operations	Constant	±5, ±10, ±12, ±15, +28, +40 VDC	Varies	Varies
AI	Constant	±5 VDC	0.5	0.1
PC/104 stack (3 levels)	Constant	+5, -5, +12, -12 VDC	30, 3, 36, 12	6, 0.6, 3, 1

I/O-Power Board

A three-board PC/104 stack is about 2.25-inches high when covered. This height is not desirable for most applications where space is a premium. If a node could be made small enough to mount directly to most panels, it would minimize cabling and allow access by pulling the panel from its Zeus rail mounting. Essentially, the panel wiring would end in a connector that mated directly to the node I/O board(s). A minimized solution was theorized to be a single PC/104 CPU (with Ethernet) stacked with a single I/O-Power board. The I/O-Power board would provide:

- DIs
- DOs
- AIs
- AOs
- Source DOs
- +/-5 VDC (interruptible)
- +/-15 VDC (interruptible)
- Variable 0-5 VDC (lighting)
- Variable 0-28 VDC (lighting)
- 115 VAC 400 Hz

The mix of I/O points and the current supply capability of the power are sized so that 90% of the panels encountered could be handled with a single board. Panels with higher I/O counts or different I/O requirements would be handled with more boards or commercial boards with other I/O types (synchros, RVDTs, etc). Panels with higher power or lighting requirements could stack more I/O-Power boards or use commercial alternatives.

Integrating power conversion circuits into the node I/O-Power board effectively distributes power conversion and creates the same benefits for power that distributed data acquisition does for I/O. A single common voltage (28V) is supplied to each node, all other required voltages are created on board. The bulk power 28 VDC supplies are standardized to a single size and type, and are readily scaled by adding more supplies as demand requires. Savings are realized from larger quantity orders of a single part, reductions in inventory and spares requirements, as well as manufacturing savings due to reduced wiring complexity. Although a single board solution that meets all the requirements is not possible, it is possible to develop a few reconfigurable boards that do meet all the requirements. The I/O-Power board design philosophy was to develop a set of boards that provided the designer with a set tools that could meet the majority of requirements.

The advent of high-speed, high efficiency, miniaturized switching power components was a necessary precursor to the design of our I/O-Power board. Before the availability of such components, this concept would have been impractical, as circuit power density would have been too low to be usable. The use of lower efficiency devices would have posed a far greater heat dissipation obstacle as well. The current state of surface mount technology allows even greater functionality gains by enabling high-density circuit design on both sides of the circuit board at a reasonable cost.

In order to attain the component and power densities necessary to make the I/O-Power board a feasible solution, a few primary design decisions were made:

- The 0.8” high stack-through bus connector was used in order to allow the use of vertical through-hole SIM power boards and other components that exceeded the standard 0.6” PC/104 stack spacing.
- Specialized intelligent power MOSFET devices were used for the DO’s in order to conserve space, and reduce the discrete component count, while providing built-in over current and over-temp protection, with fault detection and reporting capabilities.

In order to provide the multiple voltage outputs at the required power levels, the only choice was to use switching power conversion modules and circuits. Where the particular voltage function was available in off-the-shelf modules with a sufficiently small PCB footprint, they were used. For voltage functions not commercially available, discrete circuits were implemented. Use of commercially available modules reduced PCB size, as well as design effort. A notional diagram of the final board layout is shown in Figure 5.

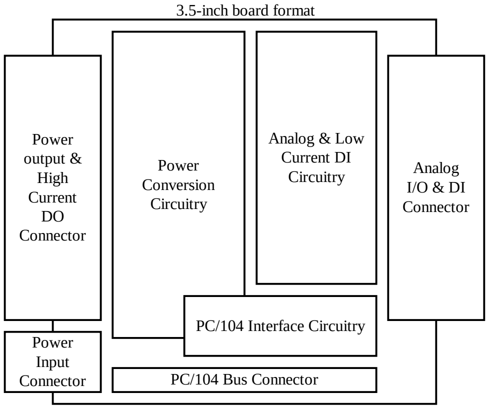
3.5-inch board format
Power output & High Current DO Connector
Power Conversion Circuitry
Analog & Low Current DI Circuitry
Analog I/O & DI Connector
PC/104 Interface Circuitry
Power Input Connector
PC/104 Bus Connector

Figure 5. I/O and Power Board Layout

The choice of the intelligent power devices fits with the need for reliability and maintainability for distributed systems. By protecting against momentary and long-term faults, the reliability of the system is significantly improved. The provision of rapid fault detection and isolation similarly enhances maintainability as well. Additionally all power outputs are monitored for over-current, over-temperature and out-of-tolerance by means of electronic circuit breaker circuits and a multi-channel system voltage and temperature monitor IC, with shutdown and fault reporting capability.

The Distributed I/O-Power Board Solution

The resulting Distributed I/O system diagram is shown in Figure 6. All distribution boxes have been removed and all custom wiring besides the panel wiring itself have been replaced with commercial Ethernet cables and 28 VDC power cables. The commercial CPU/bus board with custom I/O-Power board provides the benefits of using the ubiquitous PC architecture and the minimal connections of a simulation specific design. The availability of commercial PC I/O boards extends this capability tremendously. Even in the small volume of an aircraft panel, the PC/104 commercial standard provides a broad range of I/O options when the custom card lacks a particular I/O type.

Mechanical Considerations

Panel level resolution assumes the node acquisition hardware is attached to the panel in order to minimize cables and connections and eliminate design and test bottlenecks. Generally, this is not a problem when the panel is simulated or is a refurbished aircraft panel. Typically with centralized systems, a bracket and connector are added to the panel. In the proposed distributed system, the bracket holds the two-stack 3.5-inch square boards. The wires from the panel switches and lights terminate in the I/O-Power board connector. A self-contained system like this permits wiring and testing on a bench versus on the simulator. It also allows the software engineer to do hardware/software integration with individual panels, a new concept that has not been fully exploited.

Of course, panel mounting of the hardware can only be done if there is sufficient space behind the panel and the hardware can pass through the panel opening. In cases where this is not possible, the node must be mounted remotely. The extra cable required to remotely locate the node from the panel is simply a standard pin to pin extension of the I/O-Power board connector that only needs to be specified for length. There will be cases where an unmodified aircraft panel requires a customized cable between the panel connector and the I/O-Power board connector. This NRE is unavoidable no matter what type of I/O system is employed.

Remotely located nodes can present an accessibility issue since the node will not be available by pulling the panel from its mounting rails. This will generally be an exception, but consideration must be given to locating the remote acquisition hardware in accessible areas.

Remote mounting poses a challenge to the distributed architecture, specifically the node resolution. If the acquisition hardware has spare capacity, why not group it with a couple other panels and save some hardware cost? Although this is feasible, it comes at the cost of additional NRE for the unique cable that connects the I/O-Power board with multiple panels. It also has the disadvantage of having multiple panels being dependent on the same acquisition hardware. The dependence on the same hardware creates maintenance problems when troubleshooting. Although a purist will resist grouping multiple panels to a single node under any circumstance, it is something that only good judgement will resolve on a case-by-case basis.

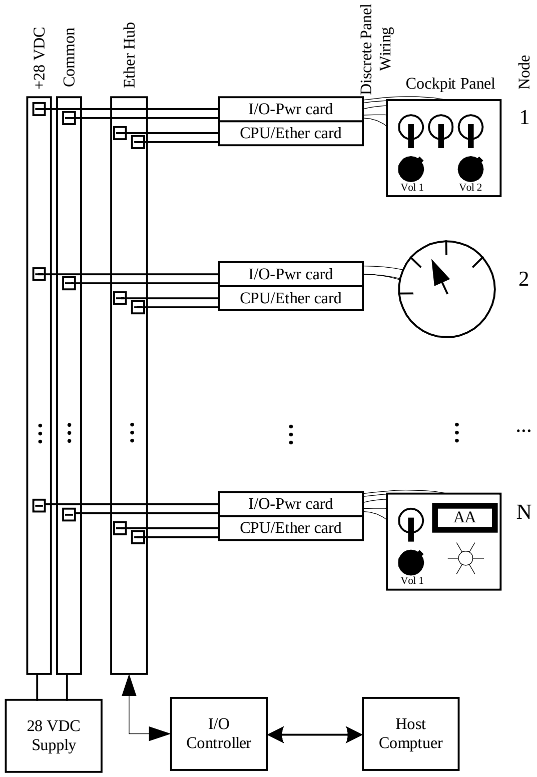

Figure 6. Distributed I/O architecture

The increased engineering and drawing effort required to produce a node stack bracket and mounting drawing for each panel, compared to a connector bracket for most panels, is expected to be more than offset by the reduction in wiring diagrams, wire lists, and distribution box drawings. The recurring mechanical assembly of these brackets would be more than offset by the decrease in the recurring cable manufacturing, trainer wiring, and distribution box assembly and wiring.

NETWORKING

Critical to the successful deployment of a distributed I/O system is the ability to network 40 to 80 data acquisition systems reliably and with deterministic data transmissions. Many factors such as bandwidth, NRE cost, reliability, maintainability, and manufacturing costs influenced the network design. The network requirements imposed are:

- 40 to 80 I/O nodes per simulator.
- Deterministic 60 Hz update rates.
- Low-cost commercial-off-the-shelf hardware and software components.
- Nodes may be removed/installed without affecting the rest of the network.

Network Protocol Selection

In order to move real-time I/O data between the simulation processor and several nodes mounted in the cockpit, a high-speed, deterministic network would be required. The ubiquitous Ethernet was the first choice since its popularity provides the highest hardware and software compatibility and availability. It was available on practically any computer hardware from rack mount PCs to PC/104 CPU cards. Unfortunately, the non-deterministic nature of the protocol was a problem. After evaluating other types of network buses, Ethernet still seemed the most promising, in spite of its shortcomings.

Deterministic Ethernet Networking

To overcome the unpredictable arbitration scheme of Ethernet, a software protocol was employed that allowed network devices to access the bus only when commanded to do so. This protocol does not inhibit CSMA/CD, it merely places restrictions on the higher level software, like applications that control when data is sent to the Ethernet device. To be effective the following rules were established:

- Only Distributed I/O specific devices could be attached to the network.
- Each device would have a unique address (node ID).
- One device would be designated the controller all other devices would be considered remote nodes.
- No remote node may access the network (bus) until the controller has sent it a request.
- Remote nodes would have a predetermined amount of time to respond to a controller request.

A prototype network consisting of three nodes, connected via 10BaseT Ethernet, was constructed for obtaining empirical network throughput and timing measurements. In this prototype network, one node was designated the controller, the remaining two served as remote nodes.

The remote nodes would manipulate the output I/O hardware according to data received from the controller and send input data acquired from I/O hardware back to the controller. The controller determines timing for these transactions. Since remote nodes must not respond until the controller first sends a packet, they are essentially synchronized to the controller.

Analysis of the prototype network determined that two milliseconds was the best round-trip time that could be achieved for a minimal size (64 bytes) packet to be sent to a remote node and a response packet received by the controller. A worst-case time is 6 milliseconds for the same setup. Given a cycle time of 16.7 milliseconds and worst-case time of 6 milliseconds to complete the transmission for a single exchange meant that only two remote nodes could be supported reliably by this system. Obviously, the 10Mbit bandwidth was not being utilized effectively.

Hold the TCP/IP

In an effort to reduce latency, the software was rewritten without using TCP/IP. Since the network driver interface was well documented and the utilization of the existing drivers would provide some hardware independence, the network access module of the application code was revised to interface directly to the device driver rather than the socket library.

After the software modifications were completed, the best-case round-trip time was 0.45 milliseconds and worst-case time was 0.6 milliseconds. The modification proved to be a prodigious step in the development of Distributed I/O; it improved performance by an order of magnitude. In this configuration, the system could support over 20 remote I/O nodes with the minimum sized data package at the highest required rate. Many more nodes may be added to the system if some nodes can use lower rates.

A Tale of Two Networks

Reflective memory was chosen to provide the interface between the Distributed I/O and the other simulation processor(s). This asynchronous interface ensures precise timing predictability by keeping timing issues on one system separate from others. Reflective memory synchronization software (handshaking) can be deployed if synchronization between systems is desired. Multiple Distributed I/O networks can be easily integrated into the same reflective memory network.

Each Distributed I/O controller node essentially straddles two networks, the reflective memory network and an Ethernet network. Each Ethernet network may comprise 1 to 253 remote nodes. Although the addressing scheme accommodates 254 distinct nodes, the practical number of remote nodes is somewhat less. Data transfer rate and packet-size are factors that must be considered when determining the maximum number of nodes on a Distributed I/O network.

Network Configuration

Numerous parameters of a Distributed I/O network are configurable and organized in hierarchical categories of: network, node, board and channel. Each network contains configurable parameters such as number of remote nodes and iteration rate. Remote nodes use parameters such as number of I/O boards, I/O board base addresses, and I/O board types. Channel configuration parameters include data scaling and formatting information.

Configuration of a Distributed I/O network is accomplished by the controller node, which parses a configuration file to determine configuration parameters for itself as well as all remote nodes attached to its network. The configuration file is extracted from the database detailing each I/O point with an off-line utility. The database holds addresses, engineering unit scaling, variable names, and other pertinent information.

Diagnostics and Failure Reporting

All diagnostic and failure reports are propagated to the central diagnostic system by the controller node via reflective memory. Each remote node performs diagnostic testing after the configuration phase is completed. Diagnostic failure messages are stored in a circular queue and one diagnostic code may be sent to the controller per frame. Data and configuration response packets have space reserved for one diagnostic code and up to three replaceable parameters. During real time operation, errors will be queued and propagated in the same manner.

Real-Time Initialization

For each remote node that is described in the configuration file the controller broadcasts a request for the node’s Ethernet MAC address using the MAC Address Query (MAQ) protocol. The remote node, with the node ID that matches that of the MAQ packet, responds by sending a MAQ Response packet. The remote node also stores the MAQ packet source address. This address is used for all further transmissions. The MAQ Response packet contains the responding node’s MAC address. The controller then stores this address in a table to reference this remote node in the future.

After a node MAC address is determined, configuration packets are sent. The remote node responds to each configuration packet with a configuration response packet, to let the controller know the packet was received. Configuration information received by the remote node is stored and used for real-time data conversion and scaling.

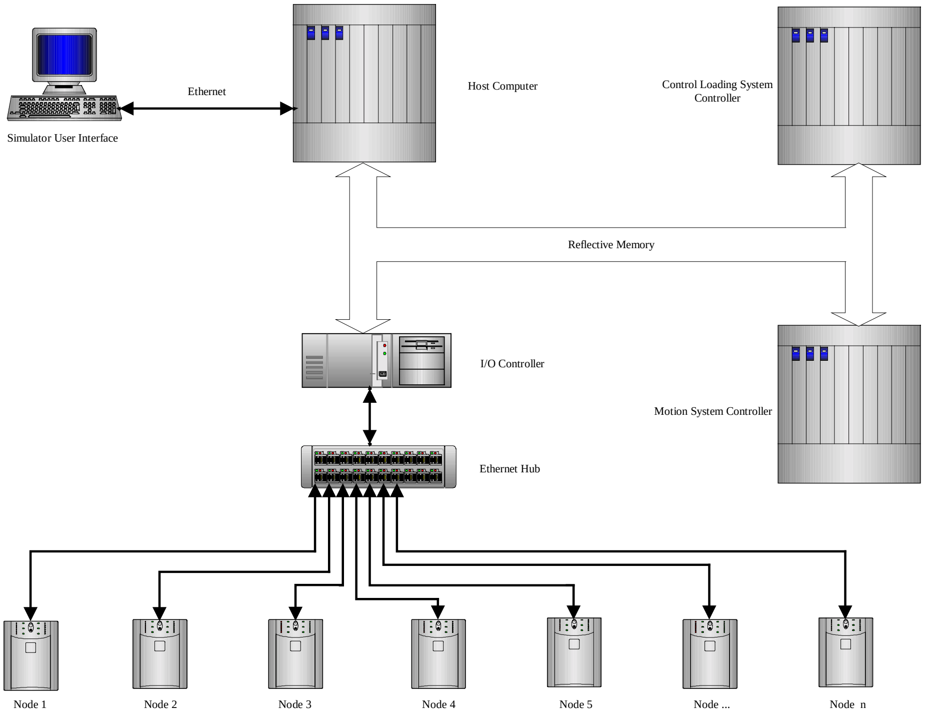

Figure 7. Distributed I/O Architecture Integrated Into A Full Flight Simulator

Real-Time Operation

The real-time data transfer begins after a remote node is configured and diagnostic testing has been performed. I/O output data is sent to the remote node from the controller. In response, the remote node returns the I/O input data to the controller that was processed in the last frame. At this time the remote node begins processing the output data and sending it to the I/O interface, then the I/O inputs are scanned in, processed, and stored in a packet structure in preparation for the next frame's data transfer. The remote node then waits for the next packet from the controller signifying the start of another frame.

To ensure the system operates deterministically, no remote node in the network may transmit on the Ethernet until it first receives a transmission from the controller; at which time it replies with a single packet, appropriate to the inquiry. That is, if the controller sends a data packet, the designated remote node must return a data packet. Likewise, if the controller sends a configuration packet the only response acceptable is a configuration response packet. The remote node must perform all of its data processing and I/O scanning between packet receptions from the controller.

In an effort to achieve maximum CPU usage, the controller always processes the previous node received packet, and the next node transmit packet while waiting for the current node response packet to arrive. For example, if the controller has just sent a packet to remote node 3 the controller will then process (move to reflective memory) the data previously received from node 2. Next, the packet destined for node 4 (or whichever node is next) will be constructed. The controller waits for a packet from node 3 or a timeout. The cycle then repeats at this point by sending the previously constructed packet to node 4 then processing the data received from node 3, etc.

CONCLUSIONS

The evolution of data acquisition hardware has made it feasible to decentralize simulator I/O. The design and wiring advantages associated with standard power and bus connections to each panel make it inevitable that more and more systems will use Distributed I/O technology both within and outside the simulator industry. Networking software is critical to the successful implementation of a distributed system and deterministic Ethernet has been shown as an economical solution.

There is still more to do, iterating designs to reduce cost, researching smaller components, and automatically linking the network configuration with electrical drawings. We have shown a great potential for distributed I/O and distributed power fundamentally changing simulator designs and reducing cost.