The IXP1200 Network Processor in a Remote Access Server

Remote Access Server - the Opportunity

Remote access used to be the domain of the big enterprise supporting its telecommuting and mobile users. But in 1998 and 1999 the market shifted rapidly from the enterprise to the Internet Service Provider (ISP.) Now companies are taking advantage of distributed ISP modem banks for their telecommuters and mobile users, and bringing the IP connections back into the enterprise over the Internet through VPNs, tunnels or using encryption.

Because the demand for Remote Access Server (RAS) ports has moved from the enterprise to the ISP, the size of the system has increased, and the cost per port has come down. Now large rack-based systems provide hundreds of ports for dial-in service. At the same time, dial-in speeds have increased, and additional features like voice-over-IP and fax-over-IP are being introduced.

The challenge for the RAS system designer is to bring down the cost per port, while increasing functionality and flexibility. Clearly a programmable solution is required, since the number of protocols is high, new features are being added and standards for VPNs, encryption and tunneling are not stable. Lower cost often means a dedicated ASIC to handle traffic flows. But an ASIC design means determining features up front, and hoping that feature requirements don't make the ASIC obsolete too soon.

Current architectures often simplify the line card, and then provide a central switching point to route traffic. A centralized architecture allows line cards to be inexpensive, but does not allow the system to scale up to large port counts. A distributed computing approach would solve this problem if the right type of processing element can be found.

Figure 1 is a block diagram for a RAS system using distributed computing. Each modem blade is connected to two busses. The packet bus, on the left, moves all user data traffic. It is a high speed, throughput oriented bus. The PCI bus, on the right, connects blades to the host processor. This bus handles updates, statistics and intermittent functions that need good connectivity, but not the high performance throughput offered by the packet bus.

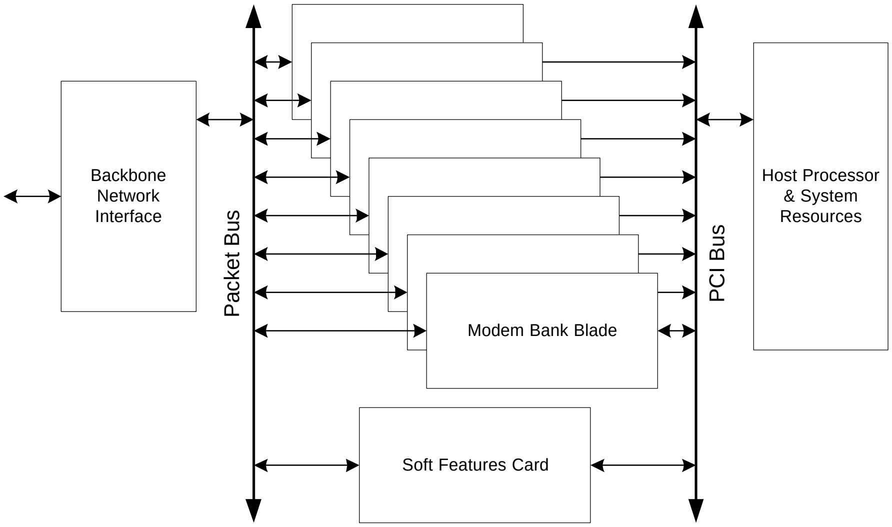

Figure 1 - Chassis Based Remote Access System

The Packet Bus, on the left, moves all packet data between the backbone and each blade, or between blades. Traffic flowing in each RAS connection (not shown) is processed locally, and forwarded to the packet bus. There is sufficient processing power on each blade to route packets at Layer 3 or above as necessary. PPP protocols can be terminated, tunnels can be terminated, and packets can then be routed via IP to their proper destination, all without involving the host processor.

Additional processing blades can be added to this architecture to add higher level features to the system. A processor card connecting to the packet bus could implement features like an email POP, an FTP server, web servers or a web switch, web caching, news group support and backup. IP based conferencing services (voice, video and/or data sharing) could also be integrated in this manner. Soft cards with easily modified feature sets could be added in what ever configuration was appropriate for each customer.

The second bus, on the right side, is a slower speed bus which connects each blade to the system host processor. The host processor can be used to configure blades, to monitor diagnostics, collect billing information and other lower speed functions. Additionally, resources such as encryption chips can be placed on the PCI bus to enable higher speed encryption and decryption of data streams. User authentication can be done on the host processor, as well as functions like DNS and DHCP if those are to be integrated into this system.

The processor for the blade must be highly integrated to reduce blade and per-port cost. Integrated memory interfaces and bus structures will reduce the cost of implementation by reducing parts count and board space. Busses must be well recognized standards that allow a simple host processor interface, and support the addition of specialty chips such as encryption or compression devices.

This embedded processor must have the capability of handling the bandwidth of all ports combined, and be able to route that traffic at level 3 or perhaps higher, depending on the functionality required. Since numerous ports will be supported by each blade, multiple streams of communication must be handled concurrently. Forwarding tables need to be calculated, or downloaded from the host processor, to keep each local forwarding function operating correctly.

A new type of processor is coming onto the market designed specifically to handle high throughput, high functionality and high integration challenges like the RAS modem blade. These processors are called Network Processors.

The Network Processor

The network processor is a new breed of CPU designed specifically for the demands of network switching equipment. It is designed specifically to handle the high speed, constantly changing data of the networking environment.

Traditional CPUs are designed based on locality of reference; they make the assumption that data recently referenced is likely to be referenced again in the near future. A cache helps make these systems run fast by keeping recently referenced information close to the processor. Network data, however, pours through a processor, is seen once, and is never seen again. Thus the traditional CPU design bogs down in the face of constant streams of new data.

A network processor takes a different approach. High-speed data movers collect data from network devices and move it directly to memory for queuing. While data is being moved to memory, it is selectively copied to special high speed engines that are able to parse the packet data, and make on the fly decisions about how to forward this data. Multiple high-speed engines allow the network processor to handle the many simultaneous threads of traffic flowing in a high-speed network. Sophisticated memory interfaces insure that the data can flow in and out of the processors memory without creating a bottleneck.

In most current network switches, this high speed switching function is handled by dedicated ASIC or custom silicon devices. ASICs can provide the high speed necessary to keep up with network traffic, but their flexibility is limited to the features its designers envisioned when it was designed. In the fast moving data networking arena, this can mean early obsolescence.

A network processor is by definition programmable. The high-speed engines that parse and manage data traffic run an instruction set designed to give the programmer complete flexibility, and provide the power and speed needed to handle simultaneous high speed data streams. This programmability means that a design based on a network processor can come to market quickly, and can be modified through software updates to track the latest standards and provide value-added features to networking equipment for years to come.

Lastly, a network processor includes an embedded microprocessor to handle the less time critical and more sophisticated aspects of network forwarding. Jobs such as determining the current routing of the network, load balancing across multiple lines and running congestion algorithms need to be done on an embedded processor that has a tight communications link to the data streams and to the data engines.

IXP1200 Network Processor in the Modem Blade

Figure 2 is a block diagram of the Level One IXP1200 Network Processor in the modem blade application. Three separate bus structures integrated into the IXP1200 Network Processor provide the connectivity needed for this application. The IX Bus provides a 4.2 Gigabit bidirectional path designed to move packet data in and out of the IXP1200 data memory. The integrated SDRAM and SRAM busses (shown here as the pointer and data memory combined) optimize the bandwidth of standard SRAMs and SDRAMs with an extremely low part count interface. The built-in PCI bus provides a control path for host processor connection, or connection to industry standard peripheral devices.

Six high-speed forwarding engines in the IXP1200 Network Processor are used to build up packets, terminate line protocols and then route packets to their destination. These fully programmable engines allow the system designer to bring a product to market quickly, but still add and update the feature content of the design over time.

A high performance processor in the IXP1200 Network Processor handles all additional local processing requirements, such as updating forwarding tables, negotiating protocols, and collecting both statistics and billing information.

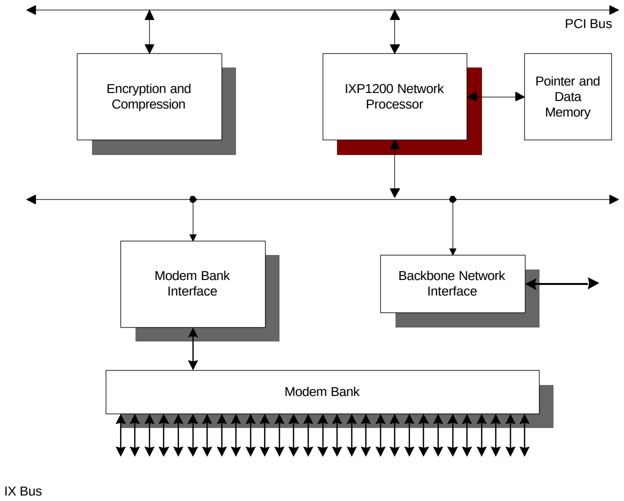
PCI Bus
Encryption and Compression
IXP1200 Network Processor
Pointer and Data Memory
Modem Bank Interface
Backbone Network Interface
Modem Bank
IX Bus

Figure 2 - Remote Access Concentrator Blade

This application benefits from the highly integrated nature of the IXP1200, taking advantage of its high speed bus interfaces and integrated ARM architecture compatible Intel StrongARM Core.

This combination of features allows the modem blade to support layer 3 processing and to remain low cost at the same time.