DIGITAL PROCESSING POOL FOR JTRS SOFTWARE RADIO:

IN-MISSION FLEXIBILITY AND EFFICIENT TECHNOLOGY INSERTION

Michael Kosmicki (Mercury Computer Systems, Chelmsford, MA; mkosmicki@mc.com)
Stephen Pearce (Mercury Computer Systems, Chelmsford, MA, spearce@mc.com)

ABSTRACT

This paper provides a summary of the main features of technical operation of a pool of processing platform, including:

1. Switch fabric and system software supporting:
   - Maximum utilization of all processing resources for waveform and processing tasks
   - Deterministic, wideband data communication simultaneously among all processing resources
2. FPGA in-mission dynamic reconfiguration
3. Component-based middleware enabling 1 and 2 above
4. Support of software communications architecture (SCA) personalities
5. Small form factor packaging

This paper also describes how these technical performance features support the needs of joint tactical radio system (JTRS) and software radio, including numbers of simultaneous waveforms and channels, scalability, flexibility, commercial off-the-shelf (COTS), technology insertion, economical spiral support, and low life-cycle cost.

1. INTRODUCTION

Supporting all of the JTRS software radio processing requirements (present as well as those envisioned for the future) in a physical envelope small enough for field deployment will require extremely efficient and flexible use of a heterogeneous pool of processing resources: field programmable gate arrays (FPGAs), digital signal processors (DSPs), and general-purpose processors (GPPs). Radar and signals intelligence (SIGNT) real-time, embedded multicomputing approaches have been deployed in the field for more than 10 years and can be readily adapted to build economical JTRS software radios. Switch fabrics for simultaneous wideband interprocessor data transfer combined with software component-based middleware can provide a platform fully supporting the range of flexibility and reuse required by modern deployed software radio applications. This architecture also permits flexible processor reprogramming and FPGA dynamic reconfiguration, as well as quick, cost-effective technology insertion to support spiral upgrades. As a consequence, the only limitation in number of simultaneous waveforms and channels is the *total* amount of processing among *all* elements of the pool. All processing elements are wideband interconnected in a manner minimizing probability of blocking – software/firmware has a level of position independence with respect to processing execution location. The system software can ensure that all processors can be fully utilized for processing tasks – extensive use of efficient fabric aware hardware DMA controllers minimizes processor load for data flow management. The "multi-channel modem" is one of the configurations readily implemented in this fashion.

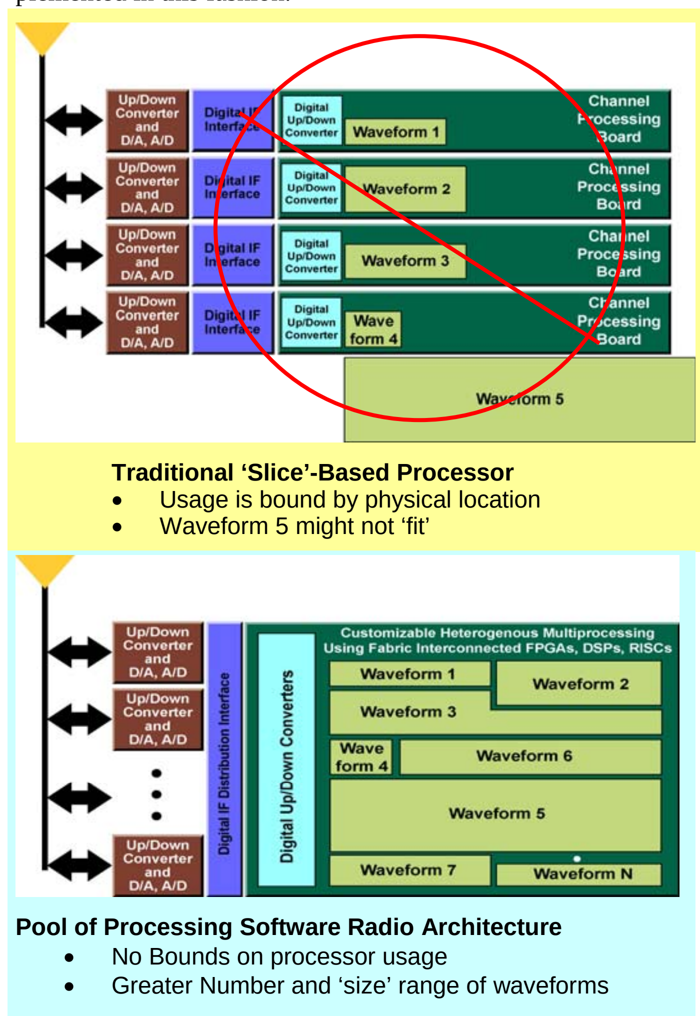

Figure 1. Pool of processing architecture (bottom) avoids severe restrictions of the traditional "slice" approach (top).

2. LIMITATIONS OF TRADITIONAL "SLICE"-BASED SDR DESIGNS

Several severe restrictions are inherent in the traditional "slice" approach (used by many current SDR designs) shown in the top half of Figure 1.

• **Very low degree of flexibility:** 'Slice'-based SDR approaches have fixed resource allocations per slice. Inefficient mapping of processing requirements to processing hardware is inherent in such approaches. Each channel has a fixed set of 'standard transceiver' processing hardware dedicated to it. Only one waveform is processed per transceiver module, and a given waveform can access only the processing in a single module. Processors of one module do not "talk well" to processors of any other module.

• **Number of simultaneous channels is severely limited:** The number of transceiver hardware modules limits the number of channels that can be processed. A fixed amount of processing hardware is dedicated to a channel, independent of the bandwidth or the GFLOPS or GOPS needed for that waveform or channel. Some JTRS waveforms, such as most legacy narrow band waveforms, will use little of the processing in such a "standard transceiver" processing module (or 'slice'). Other, wider band waveforms will more fully utilize the channel processing resources. More complex wideband emerging and future waveforms might not even "fit" within the processing capability available in a single module (e.g., in the "slice" design picture, where Waveform 5 is shown in the top half of Figure 1.

In short, the inflexibility of the "slice" design will prevent support of many JTRS goals.

3. POOL OF PROCESSING BASED DESIGN APPROACH – FLEXIBILITY AND FUTURE-PROOFING

The pool of processing software radio approach, shown in the bottom half of Figure 1, uses processing elements such as FPGAs, DSPs, and GPPs. These are interconnected by way of a wideband switched fabric (NOT a bus) with appropriate system software such that data can be simultaneously transferred among processing elements irrespective of what board they are on. This produces a heterogeneous pool of processing resources that can be flexibly accessed in the process of executing any channel or any waveform. This results in:

• **Extreme flexibility:** Physical location does not bound reconfigurable usage of any or all computing resources.

• **Real-time component middleware:** Ensuring flexible interprocessor routing of data as well as facilitating the offloading of data transfer tasks from the processors (typically implemented using hardware DMA controllers). This software control is available to the radio user to enable reconfiguration, which can be changed in-mission and software downloaded to independently reconfigure parts of the processing resource pool.

• **Ability to augment such a radio with additional simultaneous channels and/or future waveforms**, **including those with greater bandwidth**: This requires additional hardware only if the *sum* of waveform processing required throughout the radio exceeds what is there already.

• **Multi-channel "modem" cards - readily realizable**: On any given processing card, processor capability not used by one waveform is free for use by one or more other waveforms.

• **The only limitation in number of simultaneous channels is the TOTAL amount of processing among ALL elements of the pool:** A maximum number and range of combinations of waveforms can be processed with a given amount of processing hardware, as shown by the tight packing of waveform processing areas in the bottom half of Figure 1. A switch fabric provides wideband, densely interconnected data communication paths with minimum and deterministic latency among all processing elements and facilitates wideband I/O with the processing pool.

4. SWITCH FABRICS – ENABLERS FOR SCALABILITY

Figure 2 shows how processing requirements become severe as soon as combat radios support more than simple narrowband waveforms.

JTRS and other software radios need to provide multiple simultaneous voice, data, and video links in an interference-choked environment. Additional processing demands will arise from presently emerging JTRS complex communications waveforms as well as from those of the future. Also, space-time adaptive processing (STAP) approaches to interference reduction as well as multi-mission requirements all require yet further additional processing.

Switch fabrics enable full and dense non-blocking, wideband interprocessor communication among all processing elements at all locations on all boards. An example of such a switch fabric is RACEway, an industry standard design that has been field proven for 10 years in radar and SIGINT field deployments, or RapidIO® the higher-speed evolution of the RACEway switch fabric.

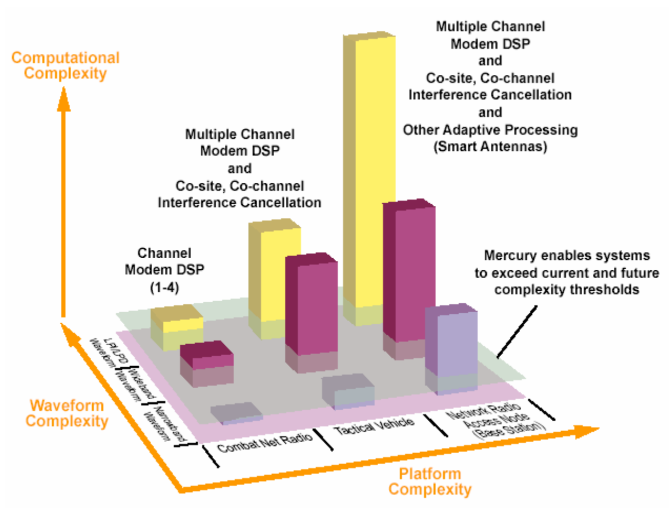

Figure 2. Processing requirements for JTRS and software radio.

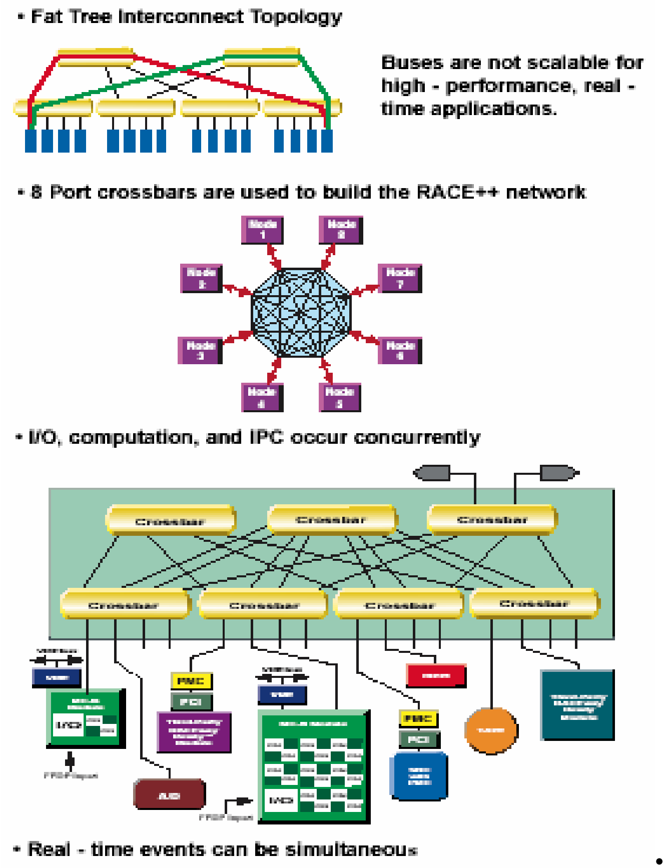

Figure 3. RACE++ and RapidIO switch fabric interconnects.

Figure 3 depicts an example of a switch fabric interconnection. Switch fabrics support the following features of a truly flexible processing pool:

• Simultaneous communication paths among processing elements; switch fabric (e.g., RACE++® or RapidIO®) node connection topology provides this through its switch hardware and control software. Each path shown in the figure represents a simultaneous data transfer between processing elements. In current configurations, RapidIO provides approximately 1 GByte full duplex (simultaneously in each direction) for each link. Higher rates are defined for future RapidIO versions.

• Processing elements (FPGA, DSP, GPP) most suited to particular types of processing can be optimally utilized as part of multiple applications (e.g. waveforms) executing within the overall multiprocessor pool.

• The access to such processing elements is unrestricted (not restricted by board, module, or 'slice' boundaries). Processing elements of any kind at any location can be used on a nearly equal footing for execution of an application or waveform.

• Data bisection bandwidth is highly scalable (whereas buses are inherently limited). See figure 4. Unlike bus-based architectures, the bandwidth capabilities grow with the size of the switch fabric. Bandwidth between processing elements within such solutions is a resource provided by the switch fabric.

Companion system software and middleware provide the ability to assign and manage processing and data transfer among

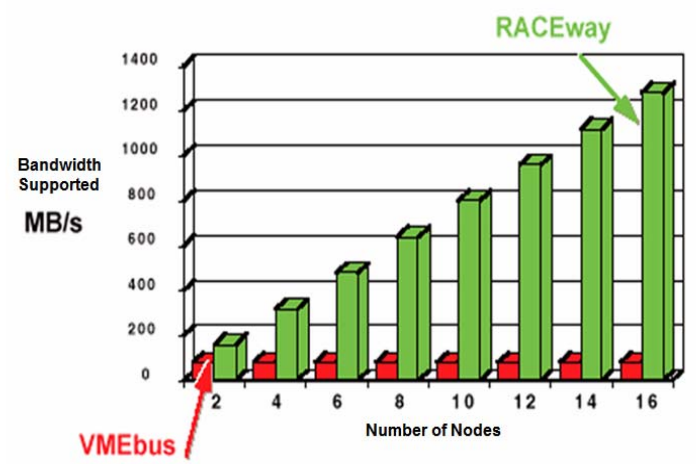

Figure 4. RACEway switch fabric is highly scalable. Buses (e.g., VME) do not scale. RapidIO has 4x, full-duplex bandwidth of RACEway and scales in the same way.

of all the elements, irrespective of their physical locations among boards. This flexibility of assignment can also permit reconfiguration supporting graceful degradation of performance if a processing element fails, as shown in Figure 5.

This network topology can also be dynamically varied at run time, since the data is moved in packets and adaptive re-routing can be achieved at all packet boundaries in all crossbar switches.

Processing elements used in RapidIO switch fabrics also contain advanced direct memory access (DMA) units. These are programmable, such that data movements between processing elements of data residing within memory systems local to each processing element are handled by the DMA units and switched fabric crossbars. This frees each processing asset to concentrate on processing data thus improving processing efficiency.

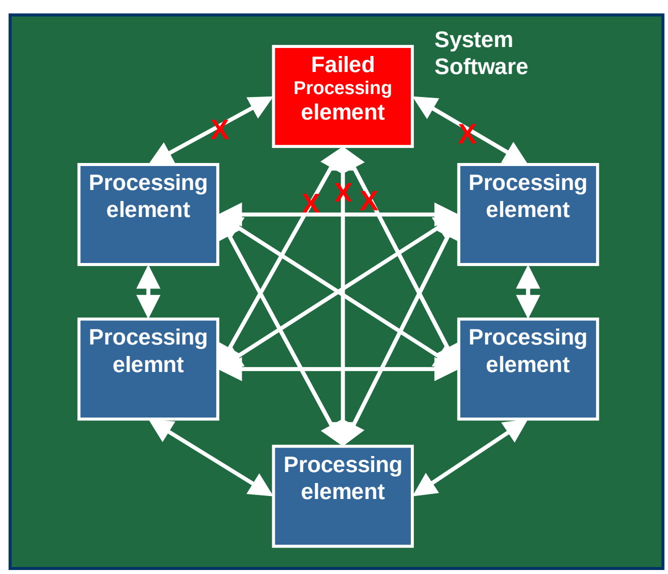

Figure 5. Pool-of-processing architecture can prevent hard failures.

5. COMPONENT MIDDLEWARE SUPPORTING FPGAs, DSPs, AND GPPs IN FULLY HETEROGENEOUS PROCESSING UNDER FULL SCA-COMPLIANCE

5.1 Middleware

Fielded radios have specific deployment constraints such as cost, weight, size, power, shape and channel capacities that can dictate some optimizations around specialized processor technologies (such as DSPs or FPGAs or even ASICs) rather than using only general purpose processors. Some functions or pieces of algorithm or waveform are best executed in an FPGA, while other functions are best handled in a DSP or GPP chip. However, since software radios must run many different waveforms, including presently unanticipated examples, the mix of processing element types is a system-design compromise based on anticipated waveform workload and physical deployment requirements. The component middleware in the radio must allow waveforms to utilize the elements available given the workload. Thus different implementations of the same waveform must be supported, using different amounts and types of processing resources. SCA-compliant component infrastructure (e.g. Mercury's) can adjust these boundaries as appropriate for each waveform workload to maximize overall radio performance. This flexibility will include the ability for a waveform to utilize FPGA resources when available, while using general-purpose processing resources in configurations where FPGA resources are fully utilized by other waveforms.

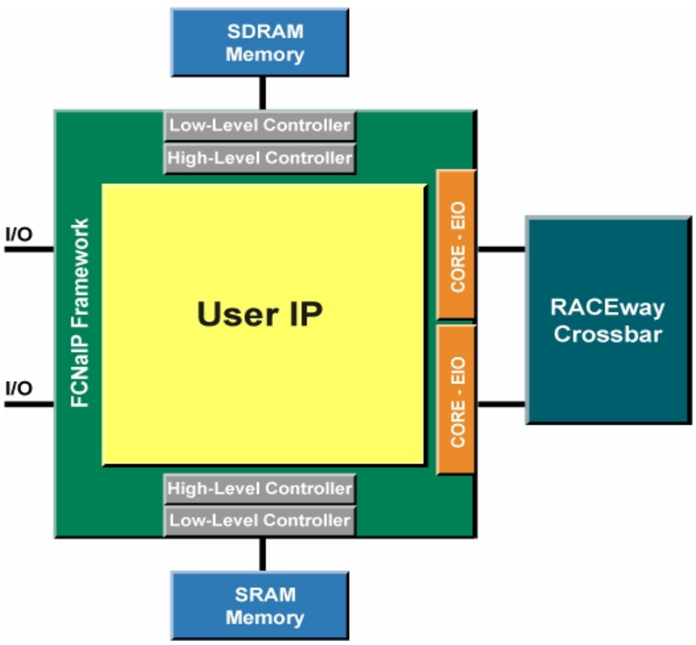

Figure 6. FPGA infrastructure for component integration.

To maximize consistent use of processing resources including FPGAs, the component middleware offers both software drivers and on-chip IP blocks (Figure 6) that will enable seamless interconnection and interchangeability between waveform components running on FPGAs and those on other processors.

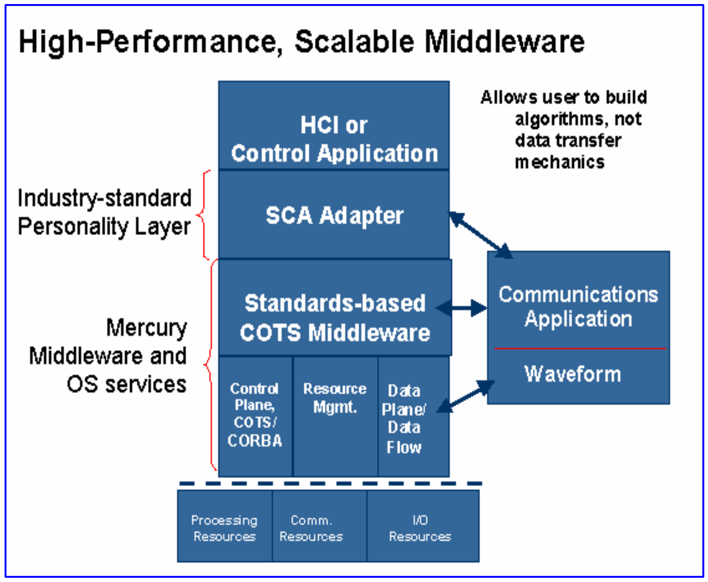

Figure 7. An example middleware structure.

This will include the rapid in-system loading/reprogramming of FPGA component implementations in tens of milliseconds without disturbing other wave- forms or other parts of the running radio. This functionality is available today for legacy RACE++ fabrics and will be supported soon in RapidIO offerings. Optimized memory and specialized DMA controllers are also provided.

Through high-performance, scalable middleware, SCA-compliant waveform components that are part of a portable waveform can be re-targeted for FPGAs, without disturbing the rest of the waveform or the software that controls or uses the waveform. This process can be non-disruptive to the waveform or its portability certification. The process can be reduced to re-coding the specific individual component's algorithm/logic without requiring the integration of the FPGA-based algorithm into the broader system environment, i.e., "It is just another component compiled for a particular processor." Rather than positioning this task as an "electrical engineering design task of large magnitude," it becomes an "algorithm optimization for a particular processor type."

Key aspects to the "mainstreaming" of these more specialized processor technologies (FPGAs, DSPs, and ASICs) are depicted in Figure 7 and include:

• Having infrastructure on all processor types that consistently exploits the pooling and fabric capabilities,
• Using common data communication and signaling protocols between components regardless of processor type, and
• Integrating the design and development process for FPGA or DSP-based components into the overall waveform, development, integration, and validation processes.

Re-implementing algorithms as components in different languages for specific classes of processors (e.g. in VHDL, or DSP-C or AltiVec™-C) will not preclude returning such component implementations to the repository of portable waveforms, since such alternative implementations can be beneficially re-used for different deployment programs. These technologies are becoming mature enough to define standards for re-use.

6. PHYSICAL PACKAGING/SIZE

6.1 3U Baseline Package

By exploiting emerging processor technologies significant reductions in size, weight, and power can be realized with a 3U solution while the processing capabilities of a much larger 6U systems can be retained (see Figure 8).

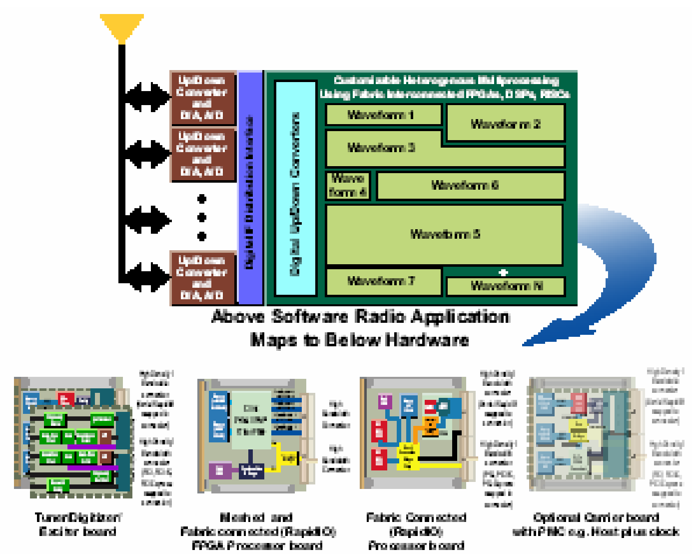

Figure 8. A COTS example of near-term mapping of pool-of-processing to 3U board solutions for software radio or for SIGINT applications.

While the SCA-compliant component middleware supports a variety of form factors, including FPGA support, the 3U form factor is generally considered suited to most medium-sized radio configurations. Figure 9 depicts an example where 3U cards including receiver/exciter channels of RF along with up to four processing cards can be provided within an ARC-210 Warrior envelope.

7. POOL-of-PROCESSING BENEFITS SUMMARY FOR JTRS AND SOFTWARE RADIO

The Pool-of-Processing approach affords:

• **Lower life-cycle costs through future proofing technology insertion spiral upgrade support:** As waveforms and software techniques evolve, reprogramming can be flexibly implemented without changing hardware. Densely interconnected heterogeneous pool-of-processing elements**,** system software, and component middleware provide a very flexible processing platform. Processing element reassignment is easy, and any element on any board can be used and reassigned in-mission. The radio operation is defined and constrained by software performance, not hardware.

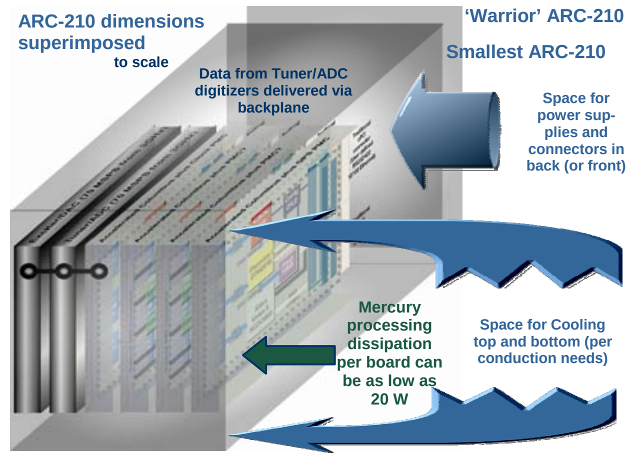

Figure 9. Up to four processing cards along with a dual RF up/downconverter can fit into an ARC-210 Warrior envelope using 3U form factors.

• **Protection against JTRS Program setbacks due to design changes:** As the JTRS program progresses past hardware platform design, if a change in waveform design or processing element mix is desired, the scalability of the pool-of-processing approach enables the project to proceed with minimum setback. The platform design does not have to change, just the software or the processing element types used in the processing mesh.

• **Jointness and Interoperability (including legacy waveforms):** Since the POP platform is easily configured through software.
• **"Multi-channel modems" – straightforward realization:** On any given processing card, processor capability not used by one waveform is completely free for use by one or more other waveforms. Any processing element on any card can operate on any waveform or portion thereof under the direction and control of the system software.
• **Scalability:** A large number of communications channels and a large number of waveform types can be simultaneously operated without redesign.
• **Dynamic Reconfiguration and Flexibility:** Design, Pre-Mission, and IN-mission radio reconfiguration.
• **Radio Reconfiguration – Changes in Workload:** As the mission progresses, re-mapping can further optimize the algorithm allocation between FPGAs and GPPs.
• **Computing asset reconfiguration – Adaptation to the loss of one or more processing elements**: The required processing can be reapportioned among the remaining processors to carry the burden.
• **Network-level reconfiguration:** Due to flexible interprocessor and I/O connections.
• **Software reuse:** Software component-based middleware allows use of software from any third-party vendor that writes to the component boundary specs. Algorithms, waveforms, or portions thereof that have thus been designed can be readily "plugged into" the system.

8. INFOSEC SUPPORT

Supporting MLS with a pool-of-processing approach is under investigation by at least one team of INFOSEC and multiprocessing expert companies – in consultation with NSA. In the meantime, some radio primes have shown great interest in using the pool-of-processing approach for the black side of the JTRS radio in the immediate term and providing a red side solution later as part of a technology upgrade spiral. This viewpoint includes the perspective that most of the radio processing takes place in the black side. Therefore, even if the pool of processing approach is used only for the black side for now, a great deal of benefit will immediately accrue to the JTRS program.

9. POOL-OF-PROCESSING IN DEFENSE PROVEN IN DEFENSE DEPLOYMENTS

Figure 10. Multicomputers for battlefield communications.

For more than 10 years, COTS high-performance embedded real-time digital signal and image multiprocessing has been deployed in critical roles in a wide range of defense applications, providing reconnaissance and surveillance platforms with real-time computational power for radar, sonar, and SIGINT signal processing. This has been the on-board digital signal processing approach for a large number of airborne ISR platforms including Global Hawk, JSTARS, Predator, Rivet Joint, U-2R, and the U.S. Army's Prophet System. Such deployments are evidence that pool-of-processing is a low-risk approach for JTRS radio.