The Next Generation of Military Tactical Radios: What it Means to be Fully Software Defined

Manuel Uhm

Vice President, Marketing

Committee Chair & Member of the Board - Wireless Innovation Forum

2013-01-09

Coherent Logix™

Enabling Low Power Software Defined Systems

Photo Courtesy NASA

Agenda

- Coherent Logix Profile
- Current SDR Realities & Limitations
- Current & Next Generation Tactical Radio Architectures
- The HyperX SDS Processor
- SDR-enabling Processor Comparisons
- Conclusions

Coherent Logix Profile

Maker of low-power, high performance, C-programmable processors (HyperX™) and RF chipsets (*rf*X™) for the embedded systems market – enabling low-power, real-time software defined systems.

Wireless

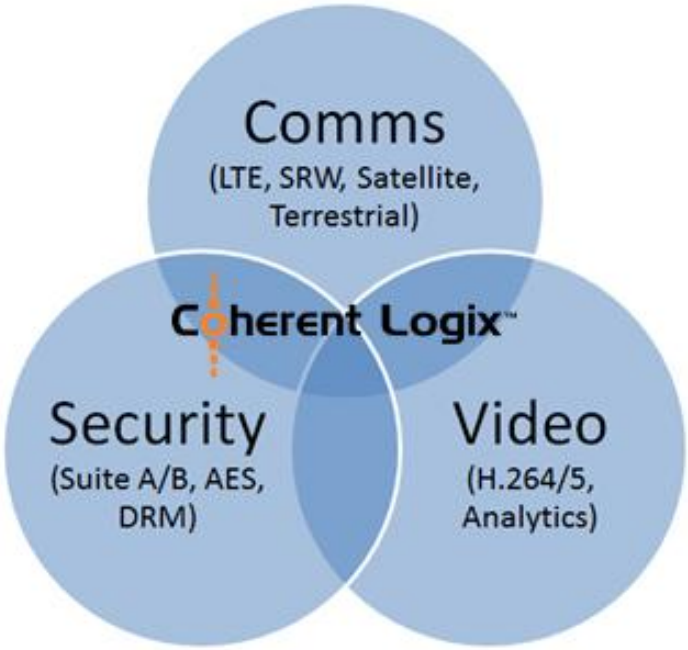
Image / Video

Mil / Aero

High-Rel / Rad-Tol

Current Application Focus

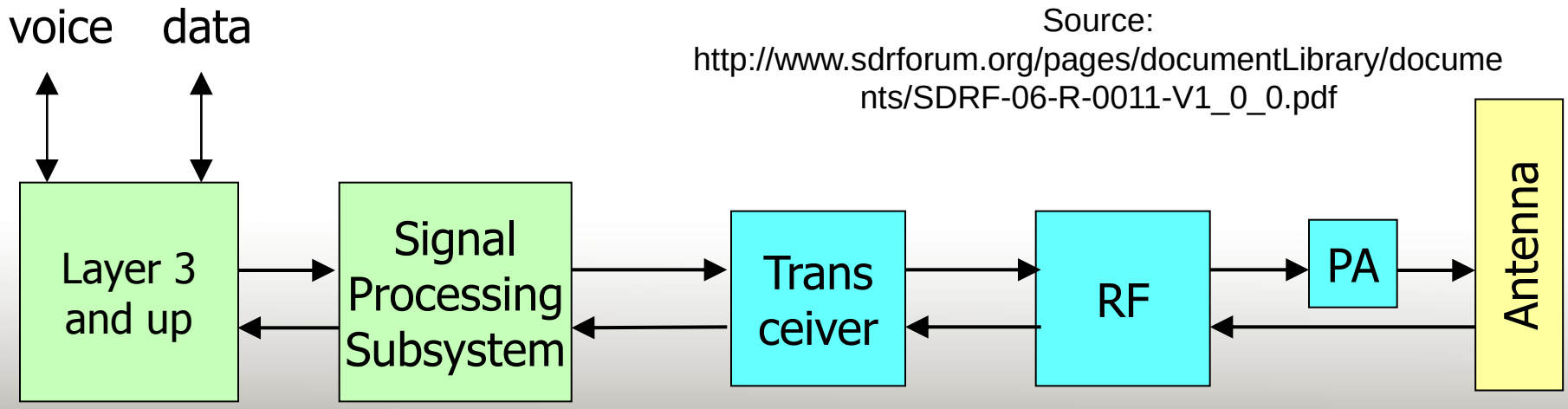
- Dual-use technology to handle wireless data explosion
  - Convergence of wireless and video
  - Convergence of communication and computation
  - Secured delivery of wireless data
  - Low power requirement is everywhere
- Enables a disruptive change in mobile infrastructure, mobile devices, intelligent cameras, and networked sensors
- Scalable systems
  - From racks to watches
  - Emphasis on small form factor low power heterogeneous computing platform



What is ***Software Defined Radio*** (SDR)?

According to the latest definition agreed upon between the IEEE P1900.1 and the Wireless Innovation Forum, an SDR is:

Radio in which some or all of the Physical Layer Functions are software-defined



Source: http://www.sdrforum.org/pages/documentLibrary/documents/SDRF-06-R-0011-V1_0_0.pdf

Current SDR Realities & Limitations

Most SDRs today are only partially software defined

- **Some use programmable ASICs with hardware accelerators**
- **Some use DSP SoCs that have hardware accelerators**
- **Even most military radios use FPGAs**

In general, performance is driving the need for partial hardware SDRs. However, the downside is:

- **Lack of flexibility and future upgradability**
- **Fail to fully benefit from the development cost savings of leveraging code reuse which results in longer time-to-market and higher development costs**

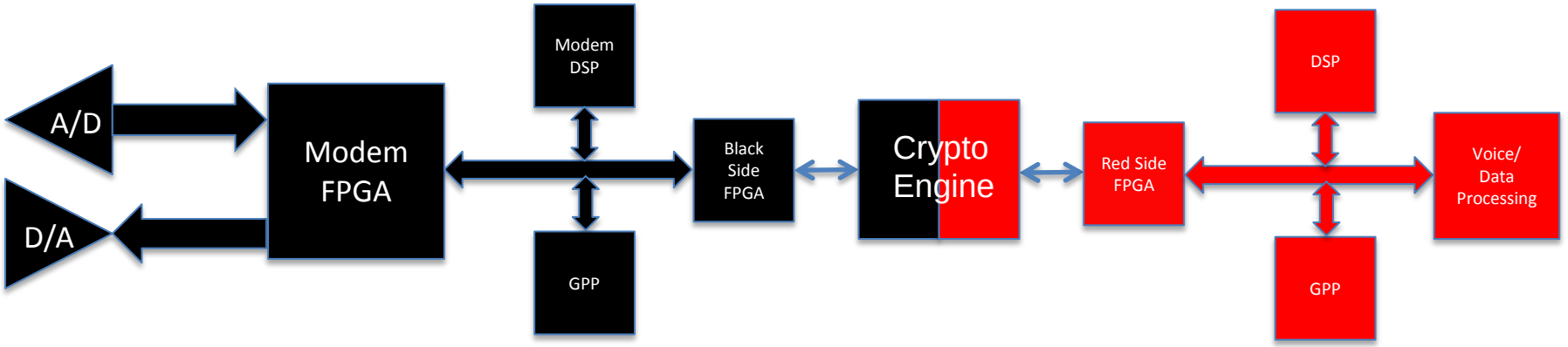
What is required to bring the vision of a fully software defined radio to reality?

Introducing…the HYPERX™ processor

A very high performance, ultra-low power multicore (100) processor that:

- **has comparable power efficiency to an ASIC - better than DSPs and FPGAs, and much better than GPPs and GPUs,**
- **has the processing performance of an FPGA to do tasks that normally require hardware accelerators completely in software, such as LTE turbo decoding and H.264 CABAC,**
- **has the ease-of-use and C programmability of a GPP, resulting in faster time-to-market**
- **may be software upgraded after deployment to support new air interfaces, codecs, advanced algorithms, or niche variants (i.e., LTE MBMS for broadcast or 4:2:2 chroma format) which other processors are not capable of supporting today,**
- **can scale from both a hardware (i.e., I/O) and software perspective (i.e., code reuse),**
- **is low latency and 100% deterministic,**
- **is highly secure with advanced digital rights management and security features.**

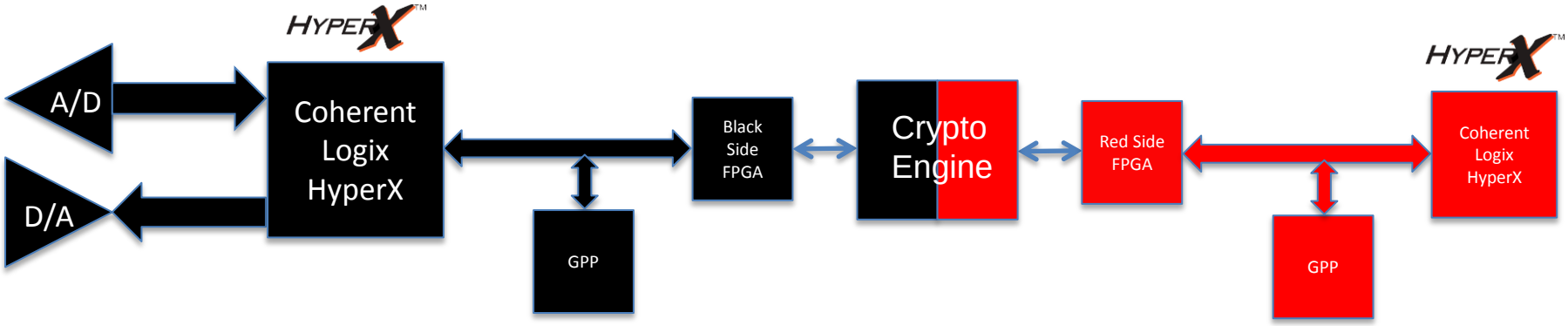
Current Tactical Radio Modem Architecture



Proven architecture, but:

- **SWAP (Size, Weight And Power) inefficient**
- **Extremely lengthy software development cycle**
- **No heterogeneous tools available to program, test or debug at system level, so test and verification is a extremely difficult and tedious**
- **Very limited code portability – even moving VHDL to a next generation FPGA family is non-trivial**

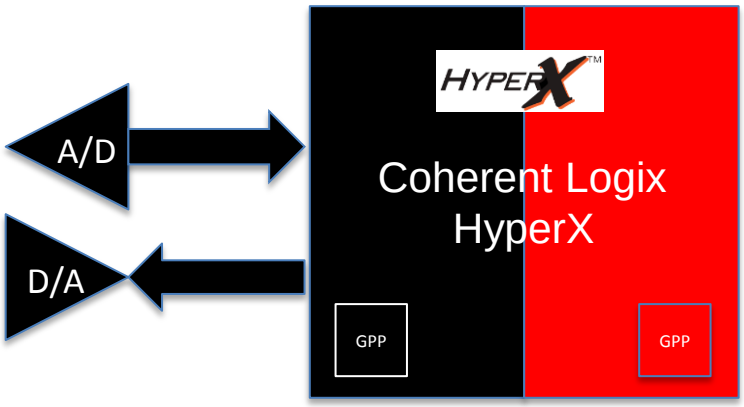
Next Generation Tactical Radio Modem Architecture



Proven architecture:

- **Coherent Logix involved in JTRS and porting of MILCOM waveforms for JPEO**
- **HyperX processor is field proven**
- **Much more highly SWAP (Size, Weight And Power) efficient**
- **Major reduction in software development cycle**
  - "5x productivity improvement…as compared to FPGA…"
- **No VHDL required for modem software – all modem code in ANSI C**
- **Can program, test and debug modem software at system level, reducing the time for test and validation**
- **Significant increase in code portability – all code in ANSI C**

Ideal Tactical Radio Modem Architecture



Currently under development:

- **Coherent Logix with US DoD**
- **Likely next generation 28nm device**
- **Ideal SWAP (Size, Weight And Power) solution**
- **Massive reduction in software development cycle**
- **No VHDL required at all – all code in ANSI C**
- **Can program, test and debug at system level, reducing the time for test and validation**
- **Significant increase in code portability – all code in ANSI C**

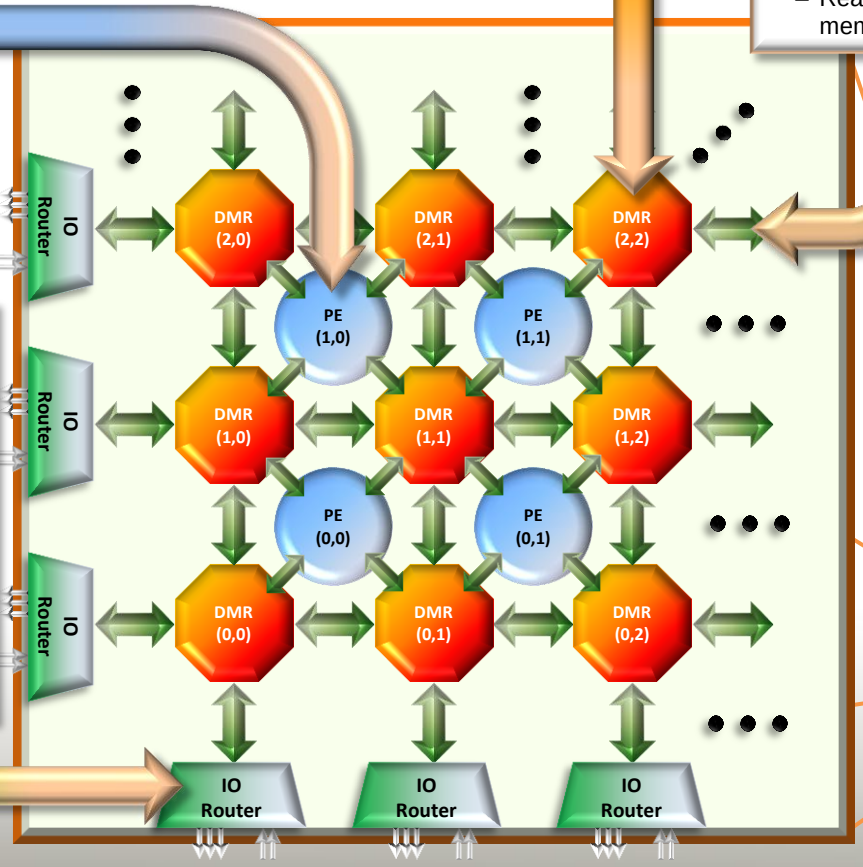
The Engine – What is the HyperX Processor: hx3100

100 Processing Resources (PEs)
- GPP/DSP w/ Variable clock to 600MHz +
- Supports data types: 8, 16, Nx16-bit integer, & 32-bit floating point
- 400KB of total on-chip program memory
  - Each PE supported directly by 4KB
- @ 500MHz
  - 50,000 MIPS
  - 50 16-bit GMACS
  - 100 8-bit GMACS
  - 25 GFLOPS

121 Data-Memory-Routers (DMRs)
- Memory Embedded in Network or Network Embedded in Memory Architecture
  - Hierarchical, Multi-Dimensional Communications
  - Physically Flat Memory
- 968KB of total on-chip data memory
  - 8KB data memory per DMR

Dynamic On-chip Memory-Network
- Autonomous data movement
- Instantaneous bandwidth on demand
- Real-time adaptable to support multiple memory and communication topologies

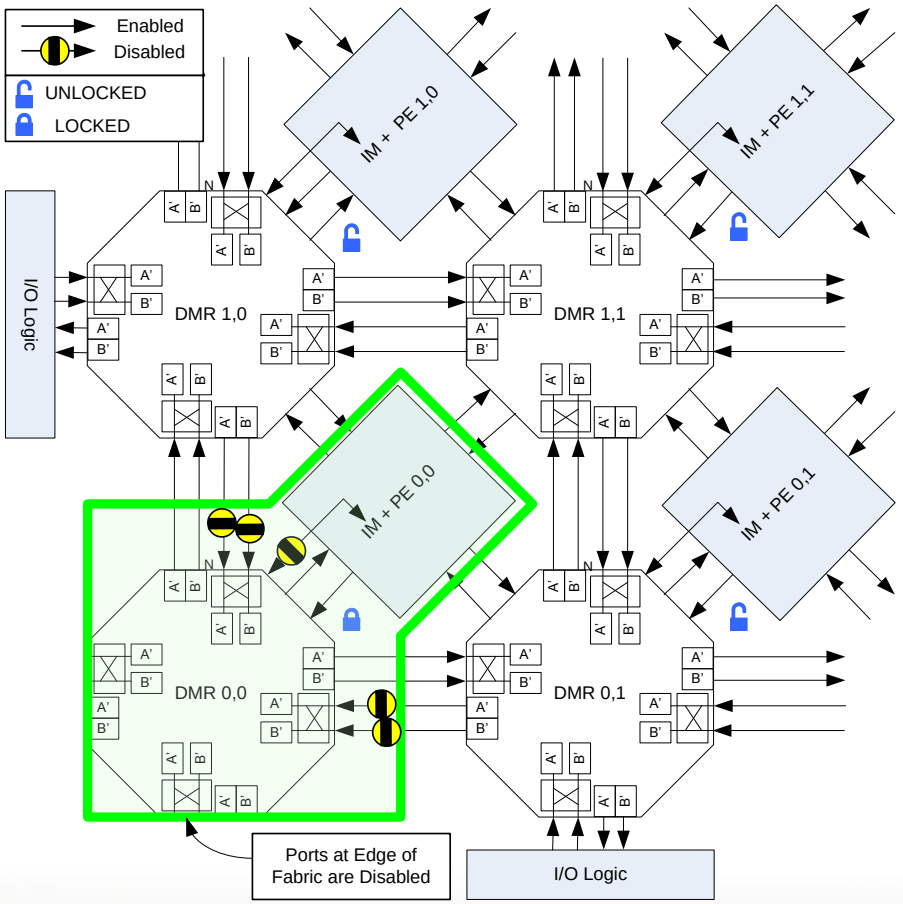
IO Routers
- 16 Multi-function General Purpose IO Channels
  - Physically Programmable
    - LVDS (EIA-644) and CMOS
  - Logically Programmable
    - GPIO, SYNC, ASYNC, & Multi-chip provides seamless chip-to-chip support *without* glue logic that would compromise performance or break the programming model
- 8 High-Speed External Memory IO Channels
  - 8 Programmable Controllers
  - Supports DDR2
  - Access up to 64 GB of total off-chip memory
- 24 programmable timers



Performance
- 32-64 16-bit GMAC/s/W
- 64-128 8-bit GMAC/s/W
- 16-32 GFLOP/s/W

Red/Black Data Separation via Security Barriers

- Security barriers provide for logically and physically “walling off” areas of the hx3100 processor
- Barriers are made up of a set of locks on individual data transfer ports on the DMRs
- Once the security barriers are constructed they may be locked from further changes until the chip is reset
- Based on the programming chosen, a chip reset can simply rebuild the barriers before any external access is allowed
- Boot Controller can program the access limitations for each access type

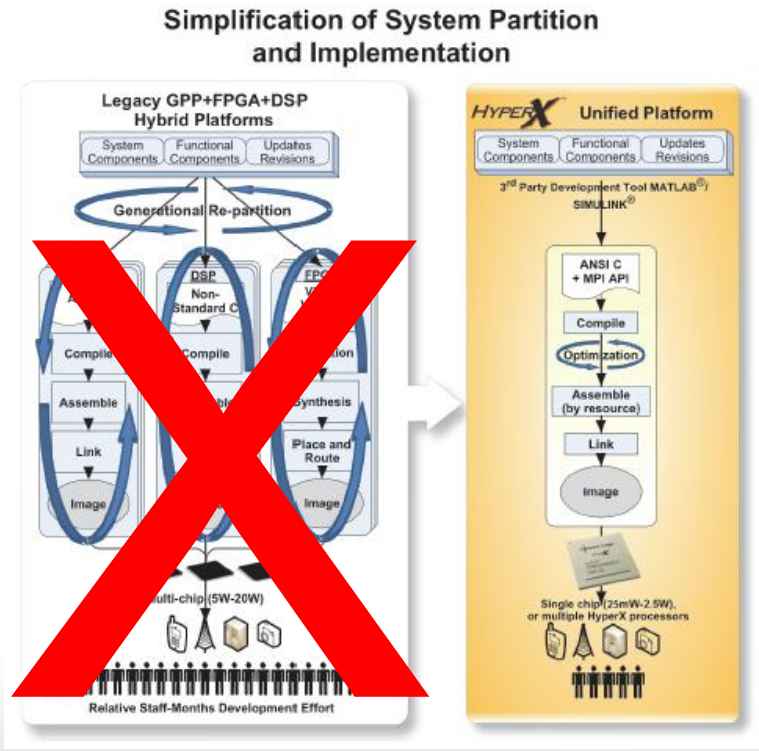


Note: DMR(0,0) is secure because after locking the SBS configuration there is no serial bus access to DM, IM -- all R/W is either by PE or DMA and the DMAs may only be operated by the PE(0,0), which is secured.

Unified Design Methodology Advantages

The design methodology is more important than the processor!

- **Program in ANSI C or Simulink**
- **Your C-based golden reference model is the basis for your design**
  - No “throw-away” code or need to redesign in VHDL
- **Very fast simulation in software or hardware enables rapid design iterations**
  - No need for behavioral synthesis or timing closure
- **Supports multi-chip designs**
  - Program and debug at the system level
  - No need to integrate and debug disparate designs on heterogeneous devices

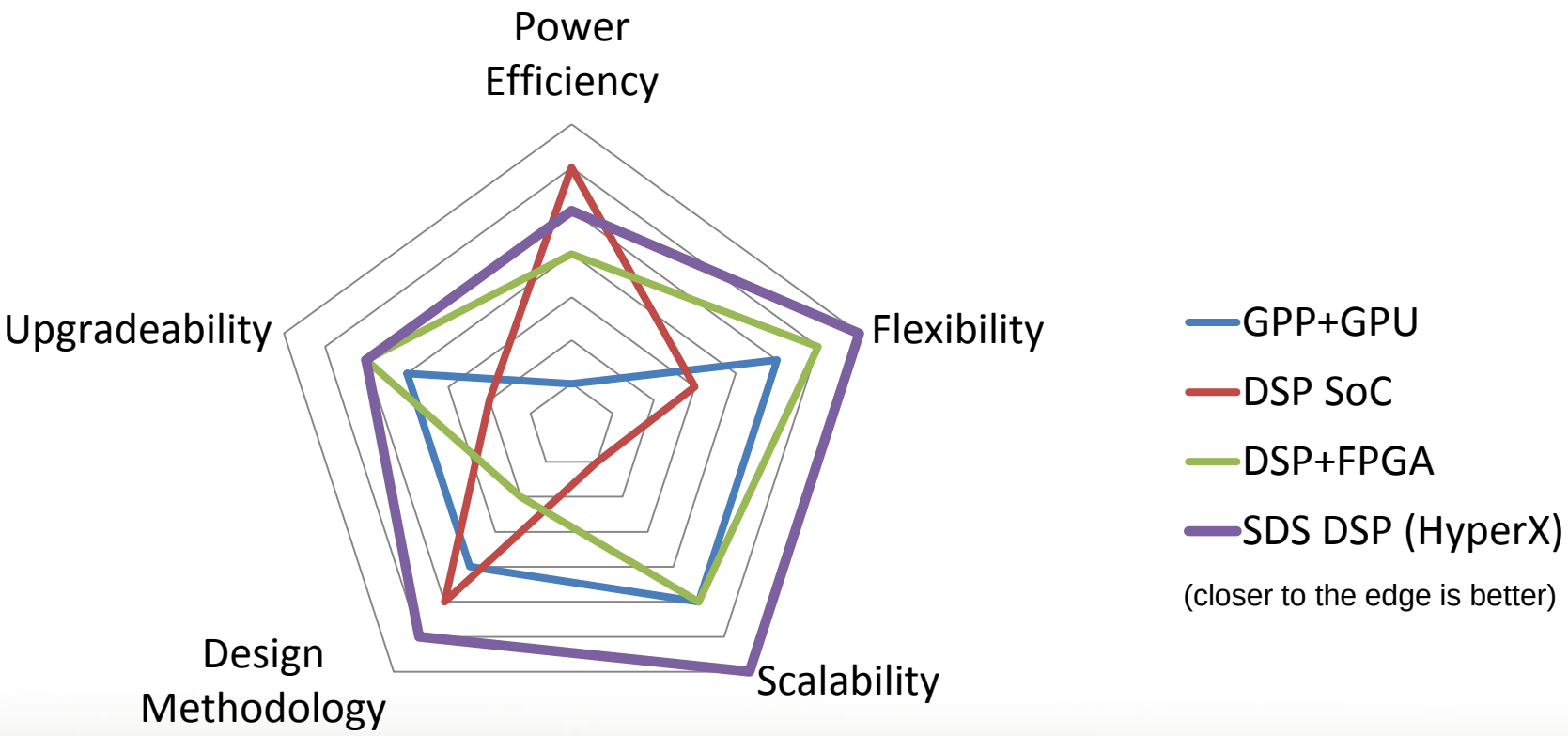


Development Savings - Customer Quote

"Application development for HyperX is done via standard ANSI C programming language constructs and within a GUI-based development environment provided by Coherent Logix. Due to the use of standard programming techniques, development productivity is higher as compared to FPGA development. Recent HyperX and FPGA development efforts within {company name redacted} indicate a 5X productivity improvement when developing for the HyperX."

This is a quote from a prime contractor customer developing US tactical radios for military communications in a report to US authorities.

Processor Options for Software Defined Radios



Processor Competitive Comparisons by Requirements

	GPP + GPU	DSP w H/W Accelerators (SoC)	Hybrid DSP + FPGA	SDS HyperX DSP (Software Defined System)
Power Efficiency (Performance/W)	Very poor. GPUs not optimized for wireless processing and are very power hungry. Too much power	Very good, but only for a small, self-contained system that only needs to do exactly what they are designed to do. Lacks flexibility	Good (between options 1 and 2) since FPGAs consume more power than a hardware accelerator.	Good (between options 1 and 2) since HyperX consumes more than a hardware accelerator.
Flexibility (multi-mode, profiles, levels, etc.)	Limited due to lack of optimization. Unable to perform compute intensive tasks like turbo decoding.	Severely limited due to H/W accelerators for functions such as turbo decoding, FFTs, and MIMO.	Good flexibility but hardware/software co-design requires careful partitioning a priori.	Very flexible due to ability to use software acceleration for both wireless and video processing.
Upgradeability (future proof, able to support new algorithms or standards)	Limited due to lack of optimization. Unable to perform compute intensive tasks like turbo decoding.	Severely limited due to H/W accelerators. Obsolete with new air interface waveform.	Possible, but difficult as it may require a total system repartitioning and rewrite.	Very upgradeable due to ability to use software acceleration for both wireless and video processing.
Scalability (adding more capability as required)	Scales but the right mix of devices difficult to determine a priori.	Severely limited due to I/O constraints designed to operate as a single chip.	Scales but the right mix of devices is difficult to determine a priori.	Very scalable. Multi-processor implementations scale with no glue logic.
Design Methodology (time-to-market, development time)	Very difficult due to lack of heterogeneous design/debug environment and lack of support for wireless/video processing.	C reprogrammability enables fast simulation and iterations, but little to no tools support for multi-chip designs (design or debug).	Very difficult due to lack of heterogeneous design/debug environment and need to use VHDL/RTL for FPGAs. Very slow iterations due to lengthy place and route	Very good due to homogeneous system-level design/ debug environment. C reprogrammability enables fast simulation and iterations. Hard to program

An SDS DSP is the best choice for a fully software defined radio.

Conclusions

- **Current Software Defined Radios are limited by being only partially software defined**
- **They are not fully flexible and do not fully benefit from the development savings of leveraging code reuse which results in longer time-to-market and higher development costs**
- **A Fully Software Defined Radio achieves the original vision but requires a high performance, low power processor that is easy to program**
- **The HyperX processor is the enabling technology for Fully Software Defined Radios**

Thank you!

Please visit our booth to see our
Software Defined System demonstration.

Manuel Uhm
Vice President, Marketing
manuel@coherentlogix.com
+1 (408) 600-1456

Coherent Logix™

Enabling Low Power Software Defined Systems

Photo Courtesy NASA