Architecture For Cognitive Radios

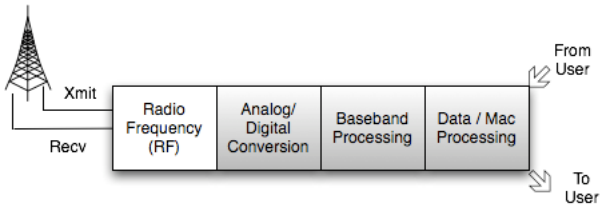
Dirk Grunwald

Department of Computer Science
Department of Electrical and Computer Engineering
University of Colorado at Boulder

Outline

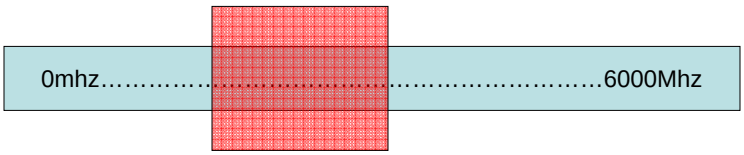
- Software & Cognitive Radios
  - What are they?
  - Why might we care
- Alternative Architectures & Implementation Challenges

An Abstract Radio



- When Computer Scientists talk about SDR/CR, they usually mean the shaded parts
- RF front-end includes tuners, filters & analog circuitry
- CR's depend significantly on RF front-end

Radio Types

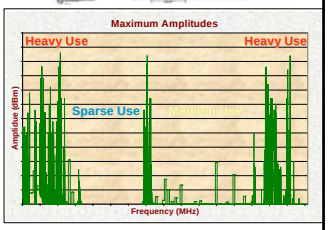


Sampled Band

Types of radio
- Hardware Radio (HR)
- Software Controlled Radio (SCR)
- Software Defined Radio (SDR)
- Ideal Software Radio (ISR)
- Ultimate Software Radio (USR)

Cognitive Radio
- "Sense and React to Environment"
- Skeptics: Just 802.11?
  - Rate adaptation?
  - Power control?
- Implicit meaning
  - Frequency adaptation
  - Protocol adaptation (*e.g.* FDM to TDM to contention)

Frequency Agility in Cognitive Radio



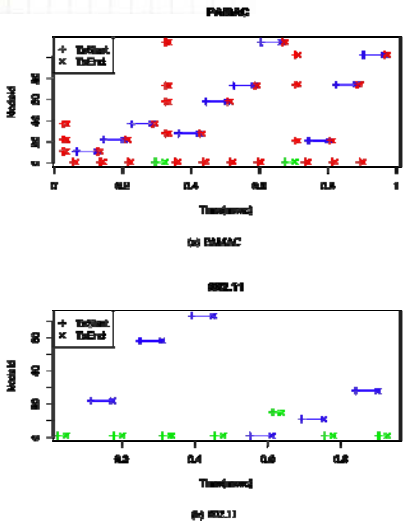
- Spectrum is fully allocated
- Urban measurements:
  - > 75% never used
  - > 90% unused on average
  - Rural areas - even more
- Cognitive Radio:
  - Avoid Licensed users
  - Communicate in “white spaces”

Hedge Uncertainty

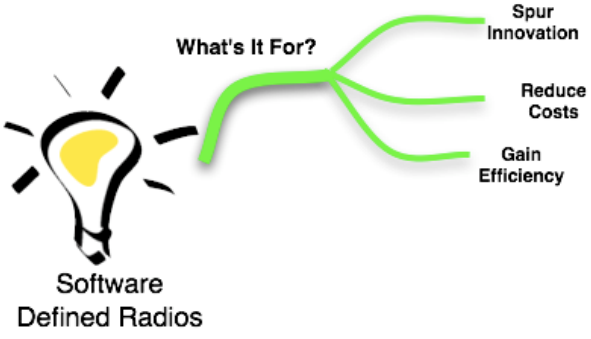
- WiMAX, LTE & 802.22 basically need the same functionality
  - Scalable OFDMA
- Why pick a winner when you don’t have to?
  - picoChip has already demonstrated WiMAX / LTE combined basestation

Cross Layer Innovation

- PHY and MAC layers defined by standards
- Long, time consuming process
- Innovation is limited because “it must work everywhere”
- SDR liberalizes that constraint



Software Radio Motivations



- Motivations conflated with business and implementation realities
- What radios in a network should use SDR?
- Can we have a cognitive radio that isn’t a software defined radio?

Framing the "Systems" Issues For Cognitive Radios

- Who, other than DARPA is really going to use wide-band cognitive radios?
- How will "constrained cognitive radios" affect systems and user interfaces?
- Will there be a 90%/10% rule in RF? How do we address each part?
- What does a cognitive need to do?

- Summarizing E^2RII White paper

  Unless you are adhoc or have existing infrastructure, dynamic spectrum access likely occurs
  - To balance broadcast (DVB) and two-way communication
  - To allow the gradual transition of technologies
  - To exploit relatively stable location-and-time varying spectrum (white spaces)

Implementation Methods for SDR & Cognitive Radio

Stations & Mobile Nodes

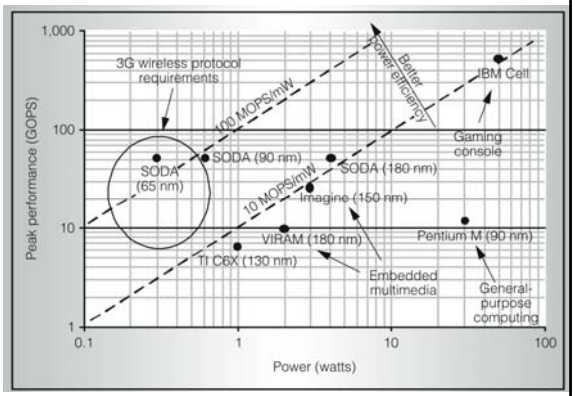
- Power over-riding issue
- ASIC
  - *SoftMAC*
- DSP / ASIP
  - *Soda*
  - *Coherent Logix*
- Hybrid
  - Combination of DSP, FPGA & GPP

Access Points & Towers

- Flexibility, longevity
- FGPA
- General Purpose CPU
- Massive Multi-Core
- GPGPU

PHY Processing Demands

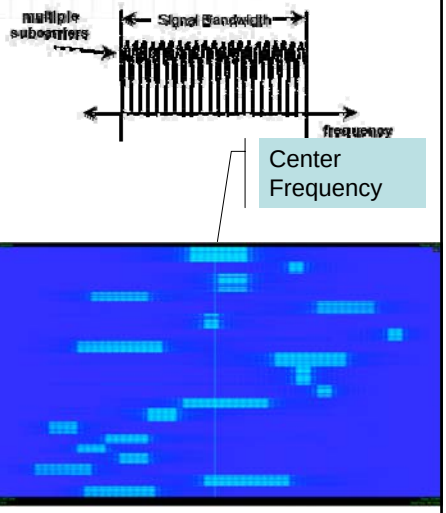
- Most DSP's provide 10-30 GOP/s per mW
- 3G requires ~100 Mops/mW with current architectures



From: *SODA: A high-performance DSP Architecture For Software-Defined Radio, Mudge Et Al*

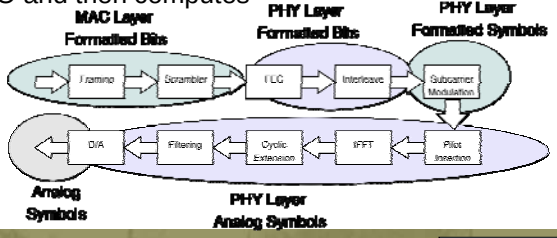
(very) Basics of OFDM

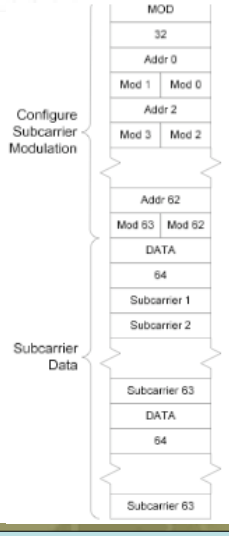
- Used in many modern PHY's
  - WiMAX, 802.22, LTE, 802.11, DVB
- Resistant to multipath
- Flexible bandwidth allocation (OFDMA)



Pure Software Cognitive Radios

- Are “Pure Software” radios possible for reasonably complex wide-band waveforms?
- Not most energy efficient solution, but proper co-design reduces costs
- DYSPAN’07 OFDMA transmitter design highlight conventional CPU bottlenecks - I/O and then computes





Jeff Fifield, Dirk Grunwald, and Douglas C. Sicker, “Experiences With a Platform for Frequency-Agility” IEEE/ACM DySPAN

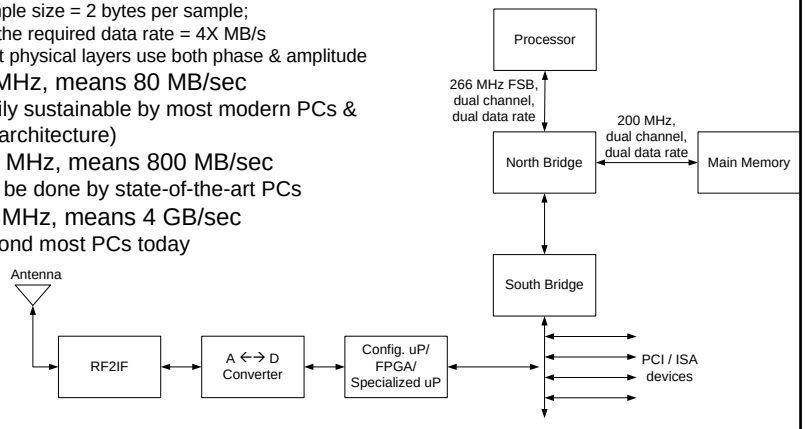
PHY I/O Bottlenecks

Bandwidth = X MHz,
- Required sampling rate = 2X MS/s,
- Sample size = 2 bytes per sample;
- So, the required data rate = 4X MB/s
- Most physical layers use both phase & amplitude

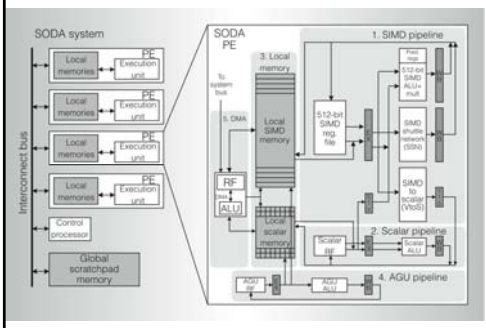
X = 10 MHz, means 80 MB/sec
- easily sustainable by most modern PCs & I/O architecture)

X = 100 MHz, means 800 MB/sec
- can be done by state-of-the-art PCs

x = 500 MHz, means 4 GB/sec
- beyond most PCs today



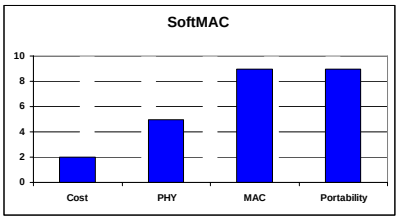
ASIP Alternative



- ASIP uses special instruction & organization
- E.g. scatter-gather units optimized for FFT
- Support for needed datatypes
- VLIW / SIMD

SoftMAC – ASIC Based Software Controlled Radio

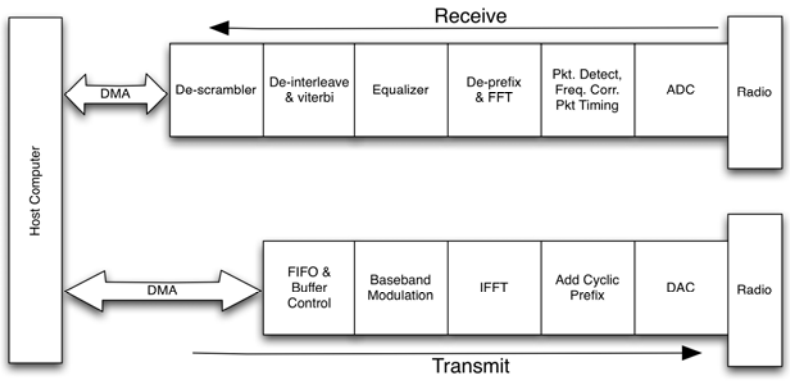
- Based on Atheros/MADWIFI drivers
- Uses software control interface to existing ASIC and MAC processor
- **Surprising** flexibility at MAC layer
  - Adaptive FEC
  - TDMA based MAC
- Intel Labs using similar technique on 802.11n chipset



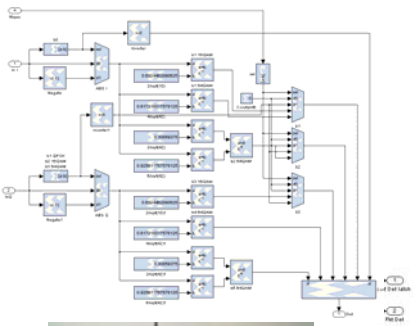
Price ≈ $60

Michael Neufeld, Jeff Fifield, Christian Doerr, Anmol Sheth, and Dirk Grunwald, "*SoftMAC---Flexible Wireless Research Platform*", Fourth Conference on Topics In Networking (HOTNETS-IV), Nov 2005

Wireless Building Blocks



FPGA Based SDR

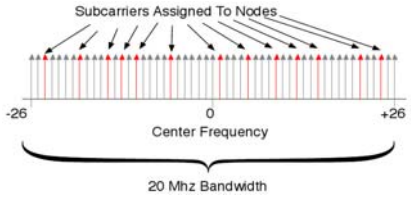


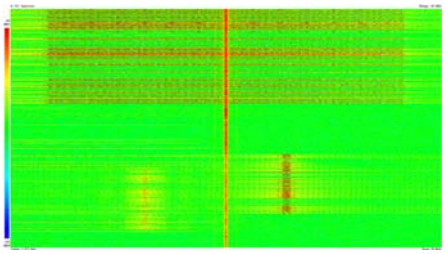
- Splitting the receiver design into FPGA+software works, but is inflexible
- Moore's law tells us to ignore the compute bottleneck, and to focus on power, flexibility and I/O
- Intra-system communication delay and variance requires tight integration with O/S

An Intelligent Physical Layer For Cognitive Radio Networks, Aveek Dutta, Jeffrey Fifield, Graham Schelle, Dirk Grunwald, Douglas Sicker, WICON 2008

Why Have Such A Flexible SDR?

- You don't know what you want until you have the opportunity to try it out
- Example: using radio platform to build low-overhead channel signaling and ranging

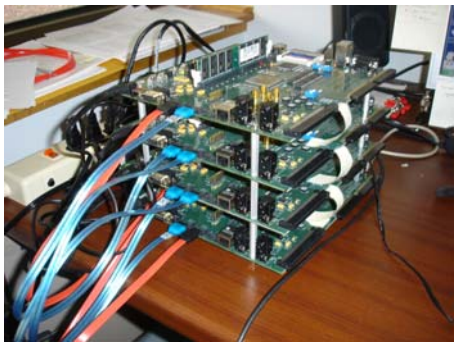




PHY-Aided MAC, Dola Saha, Aveek Dutta,, Dirk Grunwald, Under review 2008

Multi-FPGA Systems

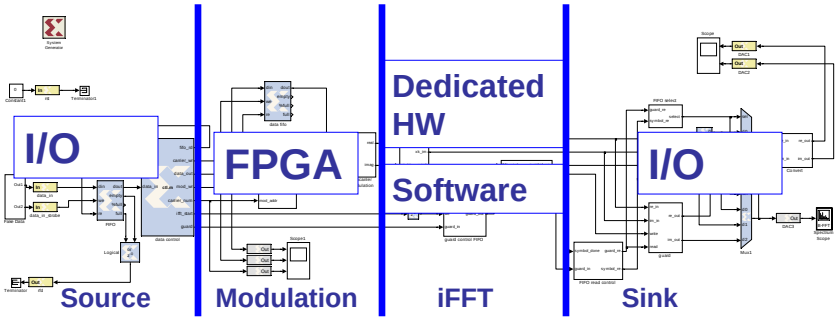
- Multi-chip FPGA Organization



Graham Schelle and Dirk Grunwald, "**Abstracting Modern FCCMs To Provide A Single Interface to Architectural Resources**", Proceedings 2007 Intl. Symp. On Field-Programmable Custom Computing Machines

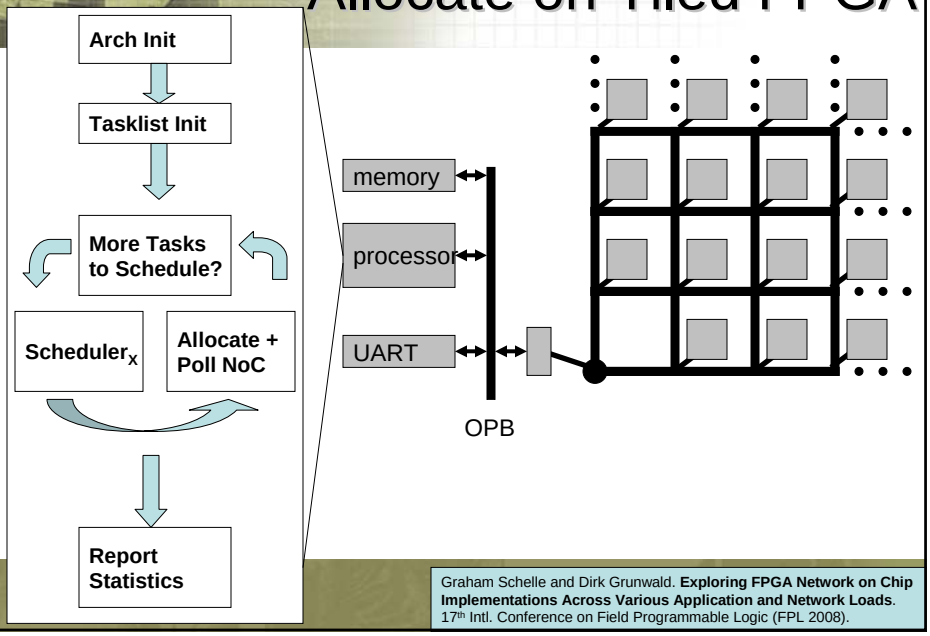
Pipeline Partitioned Design

- Decompose monolithic design
  - E.g. Source → Modulation → iFFT →Sink
- Latches become network messages



Graham Schelle, Jeff Fifield and Dirk Grunwald, "**A Software Defined Radio Application Utilizing Modern FPGAs and NoC Interconnects**", Proceedings 17th Intl. Conference on Field Programming Logic

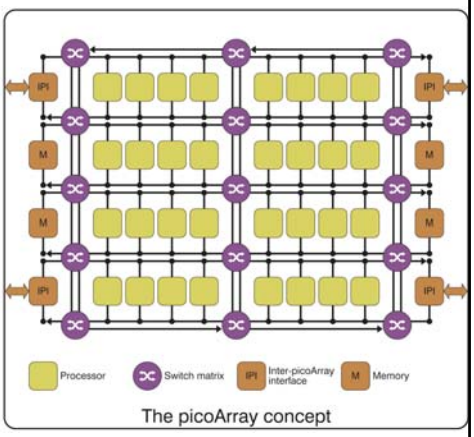
Allocate on Tiled FPGA



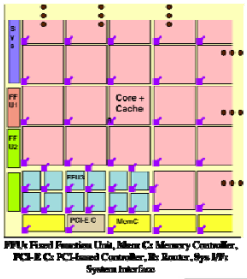
Graham Schelle and Dirk Grunwald. **Exploring FPGA Network on Chip Implementations Across Various Application and Network Loads**. 17th Intl. Conference on Field Programmable Logic (FPL 2008).

picoChip – Tiled ASIP

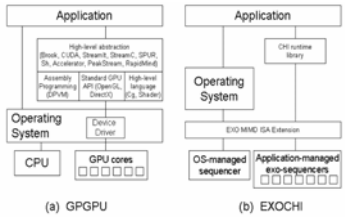
- Similar design goals
  - Many general purpose nodes
  - DSP accelerators (FEC, pre-amble detect, trellis)
- Similar desgns from coherent Logix, Ambric, *etc*
- Power profile still high for handset

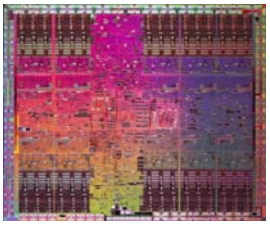


Many-core Tiled Architectures Becoming Mainstream



- Intel Tera-Server
- Exo-Chi Acclerator Exo-Skeleton
- GPGPU





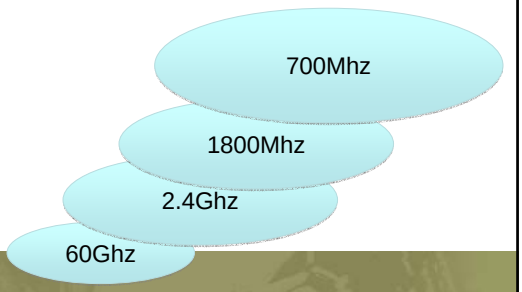
NVIDIA GT200 - up to 240 cores (graphics, HPC)

Where do these architectural alternatives lead us?

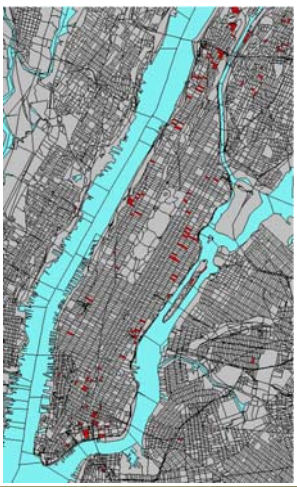
3 Goals

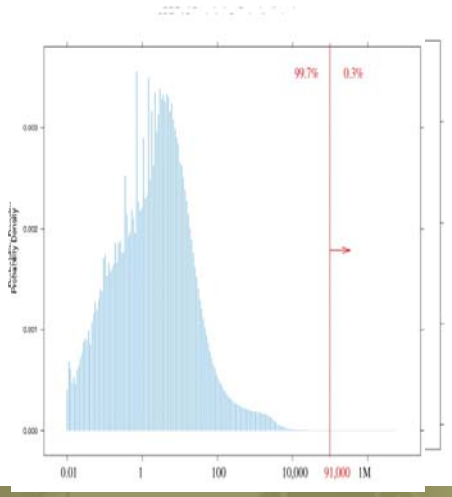
- Innovation
  - Most flexible programmable interface
- Efficiency
  - ASIC / ASIP
- Cost
  - ASIC / ASIP

- Physics matters
  - Power & Attenuation
  - Penetration and coverage
- Varying spatial scales of access and spatial reuse



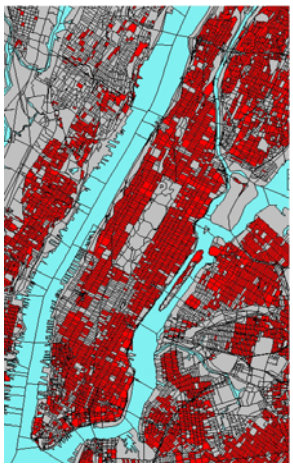
10% of the population using VoIP (on 802.11g) at once

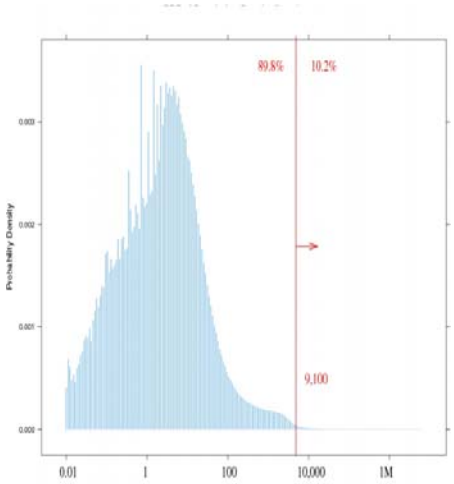




9,100 people / sq. km. is an estimated threshold density for 10% of the population to be able to use VoIP concurrently.

100% of the population using VoIP (on 802.11g) at once





91,000 people / sq. km. is an estimated threshold density for 100% of the population to be able to use VoIP concurrently.

In the near future, “links” will remain as a likely abstraction

- The 90%: the three levels of wireless imply three “more dedicated” radios
  - LTE / 802.11 / ?
  - Standards process
  - Power / Die / Antenna efficiency
- The “system abstraction” will change, but not radically
  - May need policy for operation
  - Definitely need policy for use
- The 10%: everything else
- The non-dominant technology RF protocols can be handled by commodity computing platforms
- Task for systems:
  - handle breadth of protocols with reasonable efficiency and maximal flexibility
  - Schedule and allocate resources
  - Not clear this will always be done by existing O/S
  - Same thing needed for graphics

Software Challenges

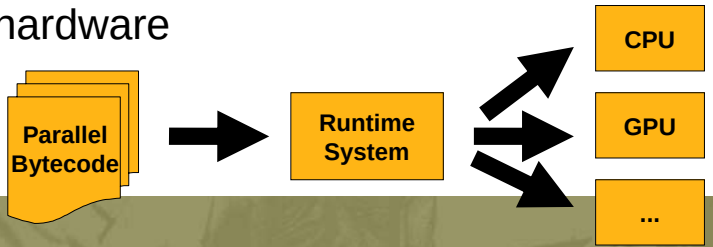
***"... performance may not scale forward with new micro-architectures and, in some cases, may regress."* - Intel[1]**

- Parallel software performance is related to core count, core types, communication latency, memory design, ISA extensions, and more...
- Programmer cannot tune their software for all possible combinations of the above
- Impossible to tune for future architectures

[1] Ghuloum, et. al. Future-Proof Data Parallel Algorithms and Software on Intel Multicore Architecture. *Intel Technology Journa*l, Volume 11, Issue 4, 2007

Proposed Solution

- End user executes parallel bytecode
- Correct mapping of available parallelism to available hardware resources occurs during software installation or at runtime
- Decouple bytecode from physical hardware



If I was a betting man..

- Future SDR / cognitive handsets include
  - Tunable wideband front end with selecting input / output
  - Signal detection (probably ASIC)
  - ASIC / ASIP components for standard links
  - Custom algorithms built using integrated GPGPU / Tiled computational resource
- Base-stations will use tiled resources
  - Increasingly based on commodity (tiled) hardware