Military and Aerospace Programmable Logic Devices (MAPLD) Conference
Sept. 15-18, 2008, Annapolis, MD

Reformation, Formulation, and the Future of Reconfigurable Computing

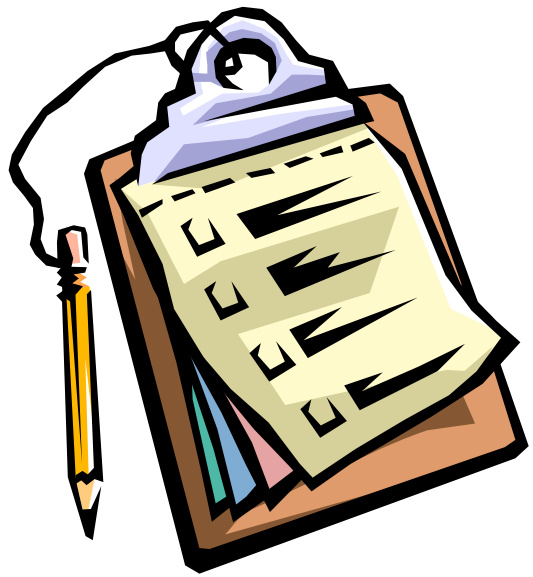
Alan D. George, Ph.D.

Director, NSF CHREC Center

Professor of ECE, University of Florida

Outline

- What is CHREC?
- Architecture Reformation
- Application Reformation
- Formulation
- Conclusions



What is CHREC?


plane

What is CHREC?

- NSF Center for High-Performance Reconfigurable Computing
  - Unique US national research center in this field, established Jan'07
  - Leading research groups in RC/HPC/HPEC @ four major universities
    - **University of Florida** (lead)
    - **Brigham Young University**
    - **George Washington University**
    - **Virginia Tech**
- Under auspices of I/UCRC Program at NSF
  - **Industry/University Cooperative Research Center**
    - CHREC is supported by CISE & Engineering Directorates @ NSF
  - CHREC is both a National Center and a Research Consortium
    - University groups serve as research base (faculty, students, staff)
    - Industry & government organizations are research partners, sponsors, collaborators, advisory board, & technology-transfer recipients

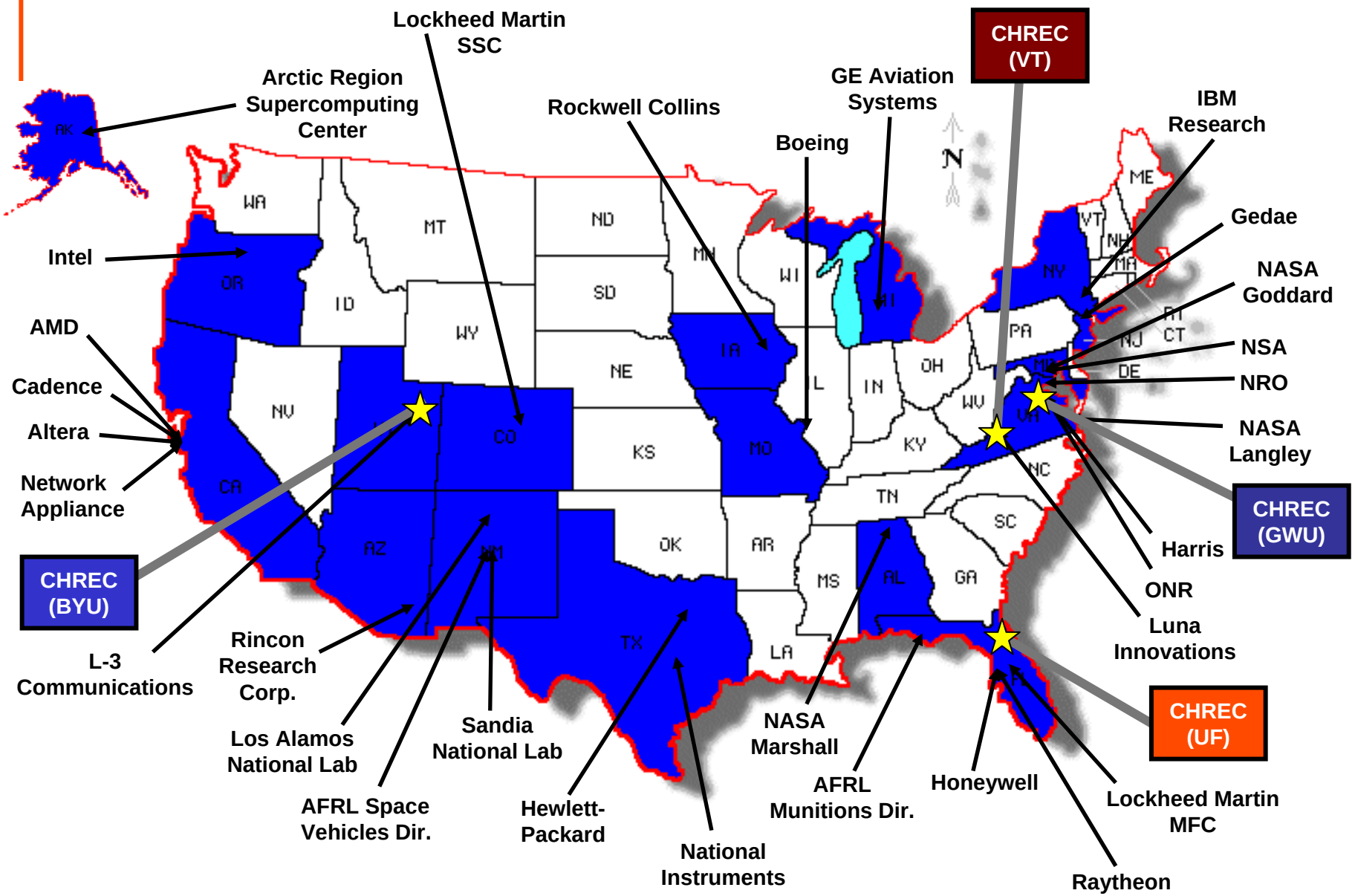
CHREC Members

BLUE = founding member since 2007
ORANGE = new member in 2008

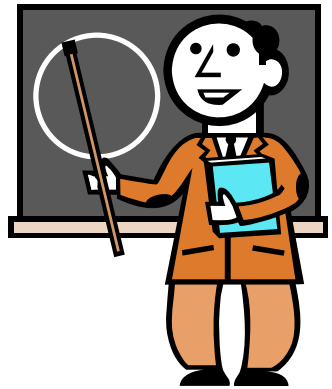
1. AFRL Munitions Directorate
2. AFRL Space Vehicles Directorate
3. Altera
4. AMD
5. Arctic Region Supercomputing Center
6. Boeing
7. Cadence
8. GE Aviation Systems
9. Gedae
10. Harris Corp.
11. Hewlett-Packard
12. Honeywell
13. IBM Research
14. Intel
15. L-3 Communications
16. Lockheed Martin MFC
17. Lockheed Martin SSC
18. Los Alamos National Laboratory
19. Luna Innovations
20. NASA Goddard Space Flight Center
21. NASA Langley Research Center
22. NASA Marshall Space Flight Center
23. National Instruments
24. National Reconnaissance Office
25. National Security Agency
26. Network Appliance
27. Office of Naval Research
28. Raytheon
29. Rincon Research Corp.
30. Rockwell Collins
31. Sandia National Laboratory NM

>30 members with >40 memberships in 2008

* Underline = member supporting multiple CHREC memberships & students in 2008.


Lockheed Martin SSC
Arctic Region Supercomputing Center
Rockwell Collins
GE Aviation Systems
CHREC (VT)
IBM Research
Boeing
Gedae
Intel
NASA Goddard
AMD
NSA
Cadence
NRO
Altera
NASA Langley
Network Appliance
CHREC (GWU)
Harris
CHREC (BYU)
ONR
Luna Innovations
L-3 Communications
Rincon Research Corp.
CHREC (UF)
Los Alamos National Lab
Sandia National Lab
NASA Marshall
AFRL Munitions Dir.
Honeywell
Lockheed Martin MFC
AFRL Space Vehicles Dir.
Hewlett-Packard
National Instruments
Raytheon
AK
WA
OR
CA
ID
NV
MT
WY
AZ
CO
NM
ND
SD
NE
KS
OK
TX
MN
IA
MO
AR
LA
WI
IL
MS
MI
IN
KY
TN
AL
OH
WV
GA
SC
NC
VA
PA
NY
ME
VT
NH
MA
RI
CT
NJ
MD
DE
N

CHREC Faculty



- **University of Florida (lead)**
  - **Dr. Alan D. George**, Professor of ECE – ***Center Director***
  - **Dr. Herman Lam**, Associate Professor of ECE
  - **Dr. K. Clint Slatton**, Assistant Professor of ECE and CCE
  - **Dr. Ann Gordon-Ross**, Assistant Professor of ECE
  - **Dr. Greg Stitt**, Assistant Professor of ECE
  - **Dr. Saumil Merchant**, Post-doc Research Scientist
- **Brigham Young University**
  - **Dr. Brent E. Nelson**, Professor of ECE – ***BYU Site Director***
  - **Dr. Michael J. Wirthlin**, Associate Professor of ECE
  - **Dr. Brad L. Hutchings**, Professor of ECE
  - **Dr. Michael Rice,** Professor of ECE
- **George Washington University**
  - **Dr. Tarek El-Ghazawi**, Professor of ECE – ***GWU Site Director***
  - **Dr. H. Howie Hwang**, Assistant Professor of ECE
  - **Dr. Vickram Narayana**, **Dr. Proshanta Saha**, and **Dr. Harald Simmler**, Post-doc Research Scientists
- **Virginia Tech**
  - **Dr. Peter Athanas**, Professor of ECE – ***VT Site Director***
  - **Dr. Wu-Chun Feng**, Associate Professor of CS and ECE
  - **Dr. Francis K.H. Quek**, Professor of CS

CHREC features a strong team of >40 graduate students spanning our four university sites.

2008 CHREC Projects

Fault Tolerance (3)
- Reconfigurable Fault Tolerance and Partial RTR (F4)
- High-Reliability Design Tools & Techniques (B3)
- Reliable RC DSP/Comm Systems (B4)

Device Studies (4)
- Device Characterization (F5)
- Heterogeneous Architectures for HPEC RC (B2)
- Process-to-Core Mapping for Adv. Architectures (V2)
- Partial RTR for HPRC (G7)

Productivity Concepts (4)
- System-Level Formulation (F1)
- Model-Based Engineering Framework (V1)
- Runtime Performance Analysis (F2)
- Intelligent Deployment of IP Cores (G6)

Productivity Studies (3)
- Case Studies in Multi-FPGA App Design (F3)
- Library Portability for HLL Acceleration Cores (G5)
- Core Library Framework (B1)

(where F=Florida, B=BYU, G=GWU, V=VaTech)

Architecture Reformation

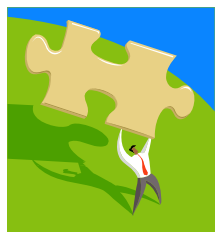
Multicore and Manycore

"RC was multicore when multicore wasn't cool."

"I was country when country wasn't cool." – Barbara Mandrell

Architecture Reformation

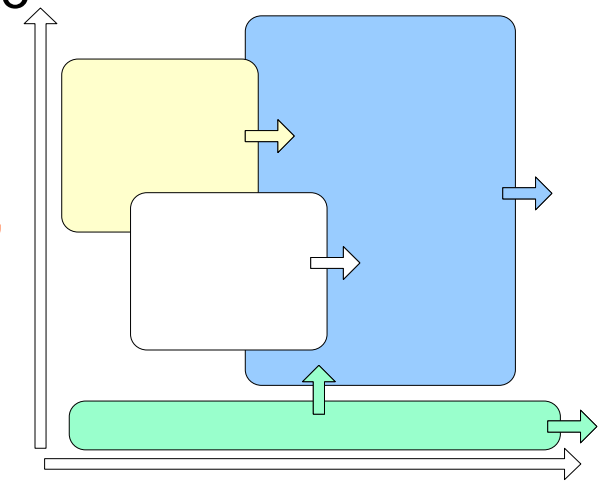
- End of wave (Moore's Law) riding f_{clk} + ILP (CPU)
  - Explicit parallelism & multicore the new wave
- Many promising technologies on new wave
  - Fixed & reconfigurable multicore device architectures
- Many R&D challenges lie on new wave
  - Tried & true methods no longer sufficient; complexity abounds
  - Semantic gap widening between applications & systems
    - e.g. App developers must now understand & exploit parallelism



- Inherent traits of fixed device architectures (FMC)
  - App-specific: inflexible, expensive (e.g. ASIC)
  - App-generic: power, cooling, & speed challenges (CPU)
  - Many niches between extremes (Cell, DSP, GPU, NP, etc.)
- Reconfigurable architectures promise best of both worlds
  - Speed, flexibility, low-power, adaptability, economy of scale, size
  - Bridging embedded & general-purpose computing, superset of fixed

What is a Reconfigurable Computer?

- System capable of changing hardware structure to address application demands
  - Static or dynamic reconfiguration
  - Reconfigurable computing, configurable computing, custom computing, adaptive computing, etc.
  - Often a mix of conventional fixed & reconfigurable devices (e.g. control-flow CPUs, data-flow FPLDs)
- Enabling technology?
  - Field-programmable multicore devices
  - FPGA is “King” (but space is broadening)
- Applications?
  - Vast range – computing and embedded worlds
  - Faster, smaller, less power & heat, adaptable & versatile, selectable precision, high comp. density



FPGA
ECA
FPCA
FPOA
MPPA
TILE
XPP
et al.

Opportunities for RC?

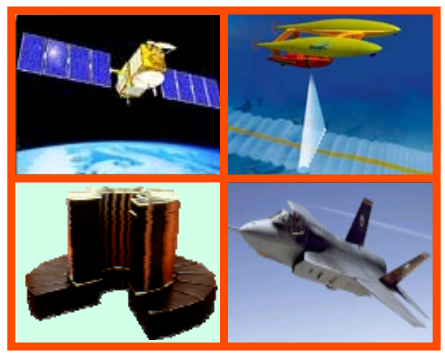
10-100x speedups with
2-10x energy savings
not uncommon

From Satellites to Supercomputers!

When and Where to Apply RC?

Performance ↑
Power ↓

- When do we need?
  - When performance & versatility are critical
    - Hardware gates targeted to application-specific requirements
    - System mission or applications change over time
  - When the environment is restrictive
    - Limited power, weight, area, volume, etc.
    - Limited communications bandwidth for work offload
  - When autonomy and adaptivity are paramount



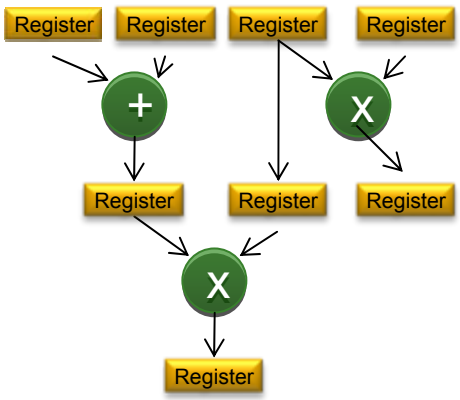
- Where do we need?
  - In conventional servers, clusters, and supercomputers (HPC)
    - Field-programmable hardware fits many demands
    - High DoP, finer grain, direct data-flow mapping, bit manipulation, selectable precision, direct control over H/W (e.g. perf. vs. power)
  - In space, air, sea, undersea, and ground systems (HPEC)
    - Embedded & deployable systems can reap many advantages w/ RC

Multicore/Manycore Taxonomy

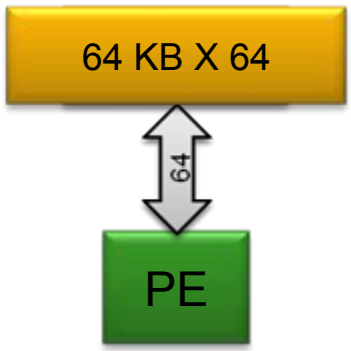
Riding the new MC wave of Moore's Law

Reconfigurability Factors

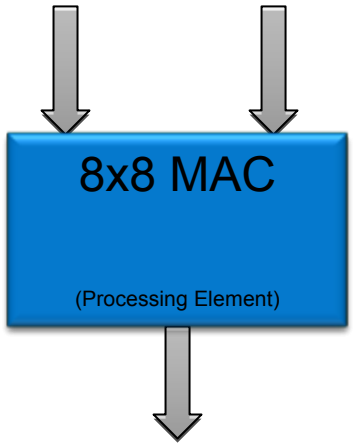
Datapath



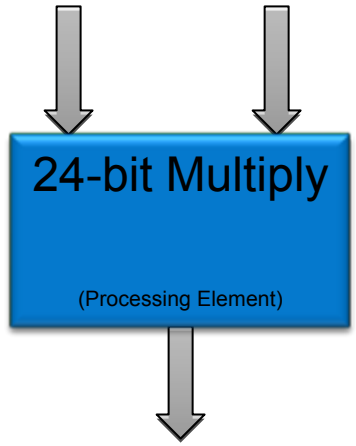
Device Memory



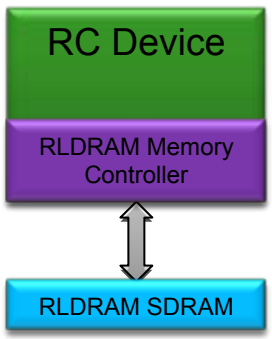
PE/Block



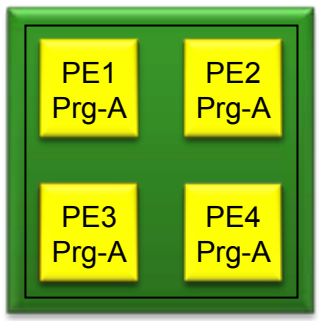
Precision



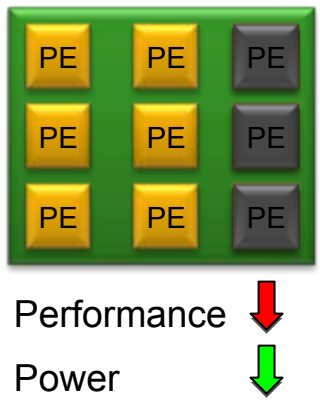
Interface



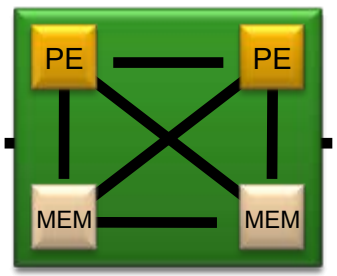
Mode



Power

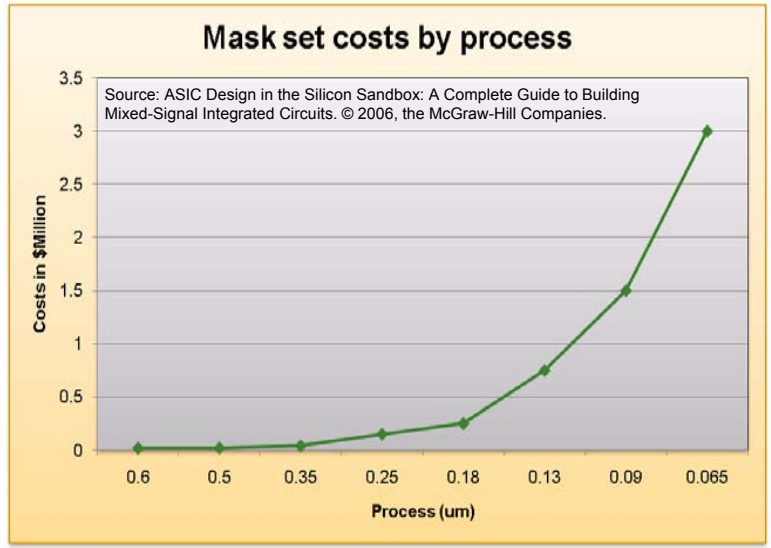


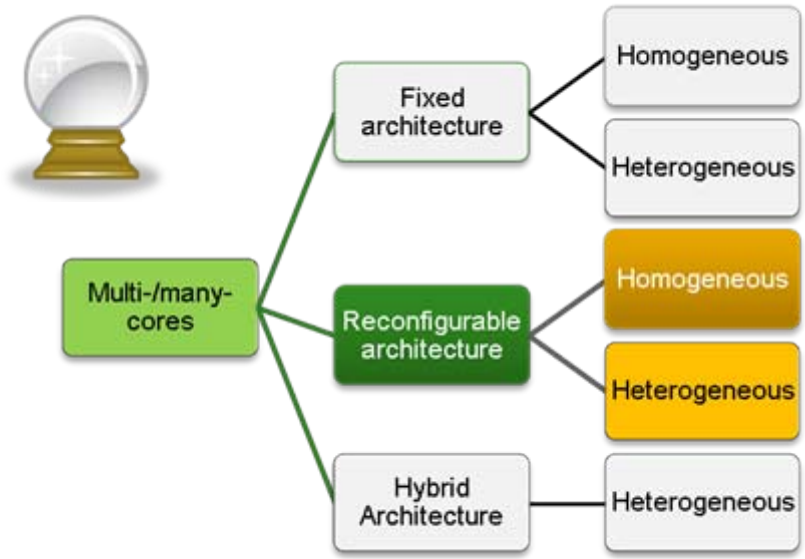
Interconnect



Future Convergence

- Rising development costs & other factors drive convergence
  - As seen in many other technologies
- Device architecture convergence?
  - Manycore is driven by densities
  - Heterogeneous?
    - Cell as initial example
    - Intel and AMD both cite heterogeneous MC in their future
    - To extent complexity is manageable
  - Reconfigurable
    - Performance + energy + versatility
    - Adaptive for many apps, missions
    - Ideal for long life-cycle systems
    - Avoids limitations of fixed architectures
    - Must manage issues of heterogeneity





RC: Vital Technologies for Future

- Mission versatility (adapt as needs change)
  - Fixed devices are burdened with fixed choices, limited tradeoffs, cannot adapt over long lifecycle
- Mission performance (speed, power, etc.)
  - One of several metrics under study @ CHREC (F5-08) is Computational Density per Watt (CDW)
  - e.g. on CDW, FPGA devices found consistently superior to FMC devices (CPU, Cell, GPU, etc.)
  - See RSSI'08 paper (and upcoming HPEC'08 talk) for details w/ HPC & HPEC devices, respectively

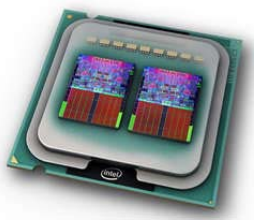
RSSI'08: FPGA consistently best in class (CDW)
- ✓Bit-level Gops/W (~28× vs. best FMC)
- ✓16-bit integer Gops/W (~17×)
- ✓32-bit integer Gops/W (~8×)
- ✓32-bit float Gops/W (~4×)
- ✓64-bit float Gops/W (~2×)

RSSI'08: Devices Studied	
130 nm FMC	ClearSpeed CSX600
	Freescale PowerPC MPC7447
90 nm RMC	Altera Stratix-II EP2S180
	ElementCXI ECA-64
	Mathstar Arrix FPOA
	Raytheon MONARCH
	Tilera TILE64
	Xilinx Virtex-4 LX200
	Xilinx Virtex-4 SX55
90 nm FMC	IBM Cell BE
	Intel Xeon 7041
	Nvidia Tesla C870
65 nm RMC	Altera Stratix-III EP3SL340
	Altera Stratix-III EP3SE260
	Xilinx Virtex-5 LX330T
	Xilinx Virtex-5 SX95T
65 nm FMC	Intel Xeon X3230

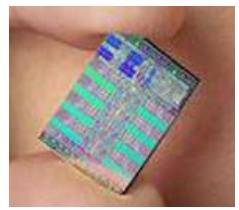
J. Williams, A. George, J. Richardson, K. Gosrani, and S. Suresh, "Computational Density of Fixed and Reconfigurable Multi-Core Devices for Application Acceleration," *Proc. of Reconfigurable Systems Summer Institute 2008* (RSSI), Urbana, IL, July 7-10, 2008.

Application Reformation

"If You Build It, They Will Come"

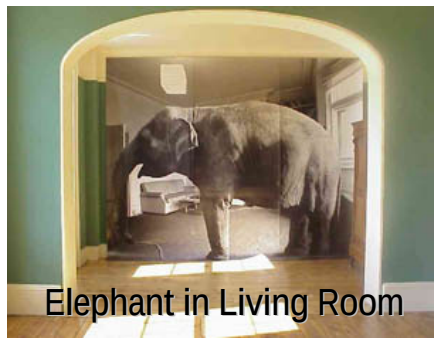


Source: *http://i.cdn.turner.com/sivault/image/2001/06/17/001234484.jpg*



Application Reformation

- Dawn of reformation in application development methods
  - Driven by architecture reformation; complexity management
  - Holistic concepts, methods, & tools must emerge
- Semantic gap widening between apps & archs
  - MC world (fixed or RC), explicit parallelism
    - Architectures increasingly complex to target by apps
    - New to fixed MC world, familiar to RC/FPGA & HPC worlds
  - Optimizing compiler ≠ parallelizing compiler
    - ***Domain scientist*** involved in comp. structure of their app
- How do we bridge semantic gap?
  - Focus upon computational fundamentals
    - Formal models, complexity management via abstraction, encapsulation
  - Learn lessons from other engineering fields
    - e.g. aerospace engineers do not flight-test first, why must we?
  - Build basis for an RC engineering discipline
    - Leverage where practical for fixed MC world



DARPA Studies @ CHREC

- Research roadmaps for app development on FPGA systems
  - Bridging app/arch semantic gap
    - Prevalent challenge of multicore
  - RC to revolutionize DoD missions
- DARPA studies by CHREC
  - Two independent studies
  - Roadmap results integrated
- Focus areas
  - Study underlying tools limitations
    - Theory, practice, technologies
  - Formulate strategic research paths
    - Revolutionary, impactful
  - Craft proposed research roadmaps
    - Highlight DARPA-hard challenges

Titles of Two Studies for DARPA

- **Exploration of a Research Roadmap for Application Development & Execution on FPGA-based Systems**
- **Future FPGA Design Methodologies and Tool Flows**

Update: Workshop held on 6/05/08

- Sponsored by DARPA
- Capstone event for both studies
- >50 experts in attendance
  - ➢ Morning presentations
  - ➢ Afternoon breakout groups
- Outcome: program research roadmap
  - ➢ Integration of both studies

Key Questions for DARPA Program

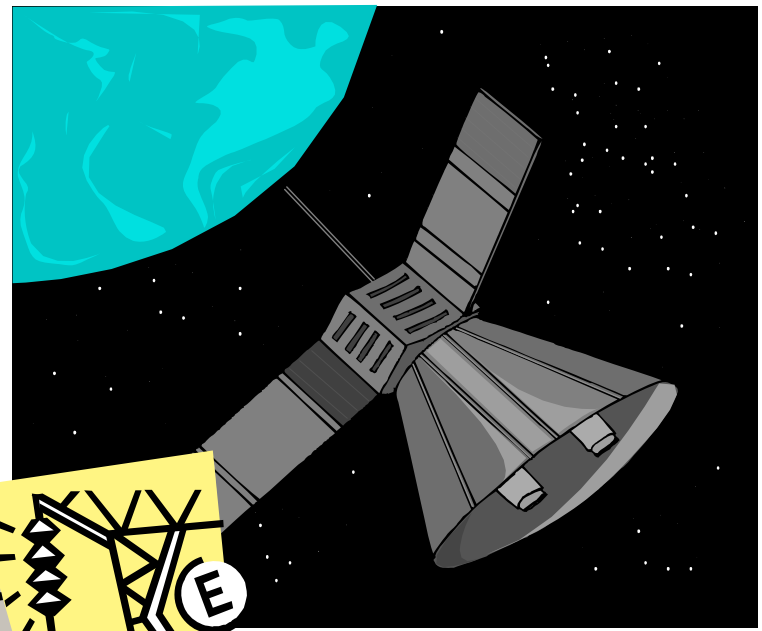
- Why is FPGA-based reconfigurable computing (RC) of increasingly critical importance to DoD?
  - Performance, power, versatility, weight, size, cost
    - 2007 formation of CHREC is proof: >30 industry, government, & university research partners, many DoD-related
- What is #1 challenge of RC for DoD?
  - Programmability: limiting factor, semantic gap
    - From deployed systems for warfighters to DoD supercomputing
- Is this challenge unique to RC for DoD?
  - Absolutely not: in general, all multicore architectures (FMC *and* RMC) are facing similar fundamental issues
    - How to productively express & exploit hardware parallelism in a manner suitable for app developers including domain scientists?

FMC = fixed multicore/manycore devices
RMC = reconfigurable multicore/manycore devices

Formulation

Starting point for reformation in application development

Principal Challenge is Complexity



Seat-of-pants formulation

"Sail west until landfall made, all the while hoping that you don't fall off the earth."

Strategic formulation

"Strategically explore various approaches, predict outcomes, study tradeoffs, choose best."

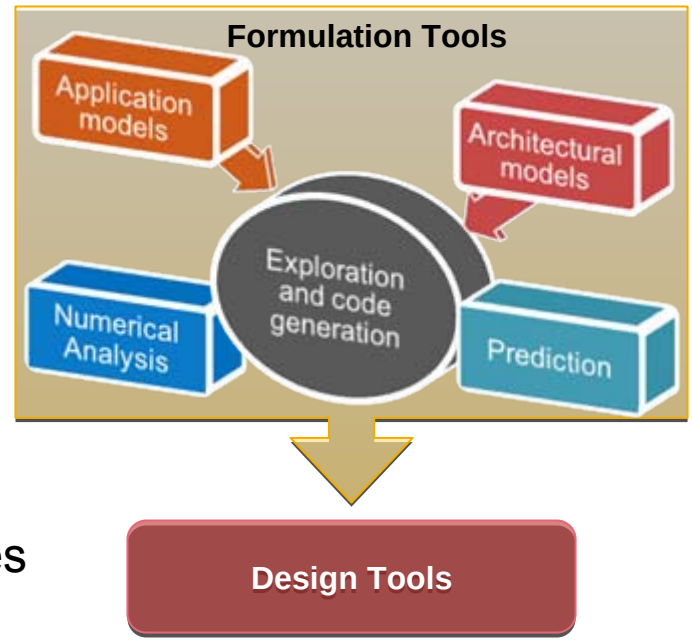
FDTE Model

"We need a change in mindset, not simply another programming language."

Formulation
Design
Translation
Execution

I. Formulation

- Strategic exploration
  - ***Not*** coding in traditional sense
- Parallel algorithm exploration
  - Control structures (wide, deep)
  - Data structures (elements, precision, layout)
- Parallel architecture exploration
  - As mapping targets of parallel algorithm
  - Base characteristics (e.g. DoP, OPS, B/W)
- High-level performance prediction
  - Supports tradeoff analysis (alg, arch, both)
  - Memory hierarchy, data locality, bottlenecks
  - Analytical, simulative, or combo
- Feeder to **D**esign phase
  - Patterns, templates, code generation, libraries
- *Theme: strategic design decisions*



FDTE Model (continued)

II. Design

- Linguistic design semantics & syntax
- Graphical design semantics & syntax
- Hardware/software coding, co-design

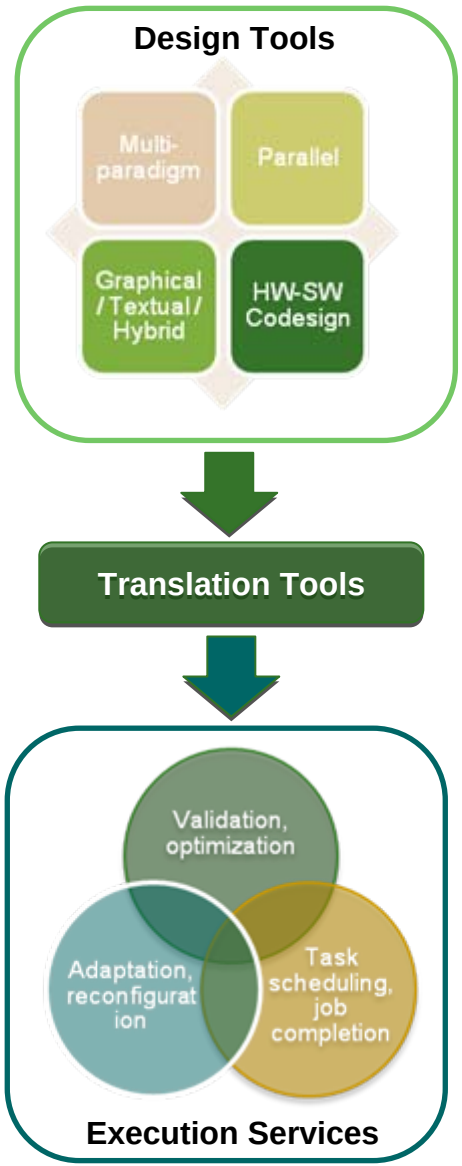
III. Translation

- Compilation
- Libraries & linkage
- Technology mapping (synthesis, PAR)

DTE phases traditionally used for "seat of pants" formulation, but increasingly *inefficient* and *inappropriate*.

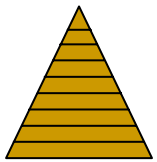
IV. Execution

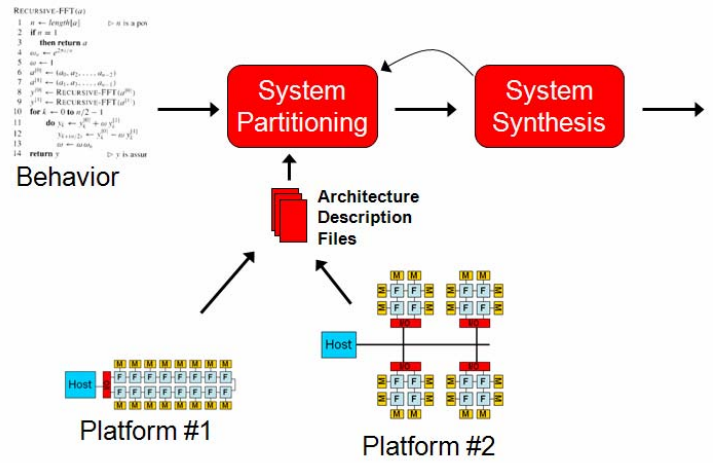
- Test, debug, & verification
- Performance analysis & optimization
- Run-time services



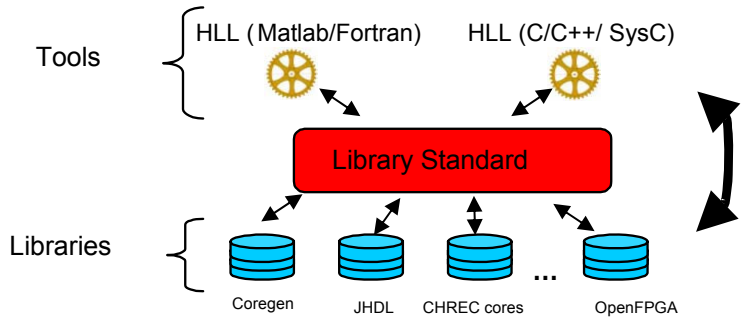
ART Model

- **Abstraction**
  - Reduce detail required to specify computations by raising design abstractions
    - Leverage emerging concurrent models of computation
    - Remove circuit-level details
    - Support multi-FPGA synthesis

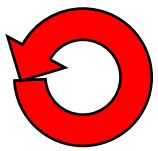


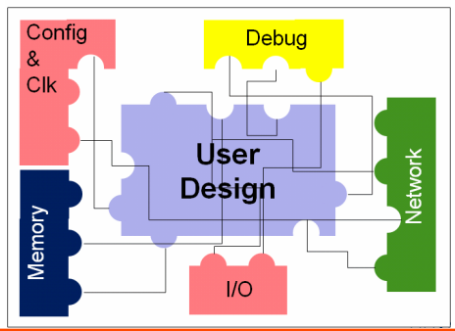


- **Reuse**
  - Substantially increase amount of design reuse at all levels of design flow
    - Library reuse standards
    - Dual-layer compilation
    - Interface synthesis

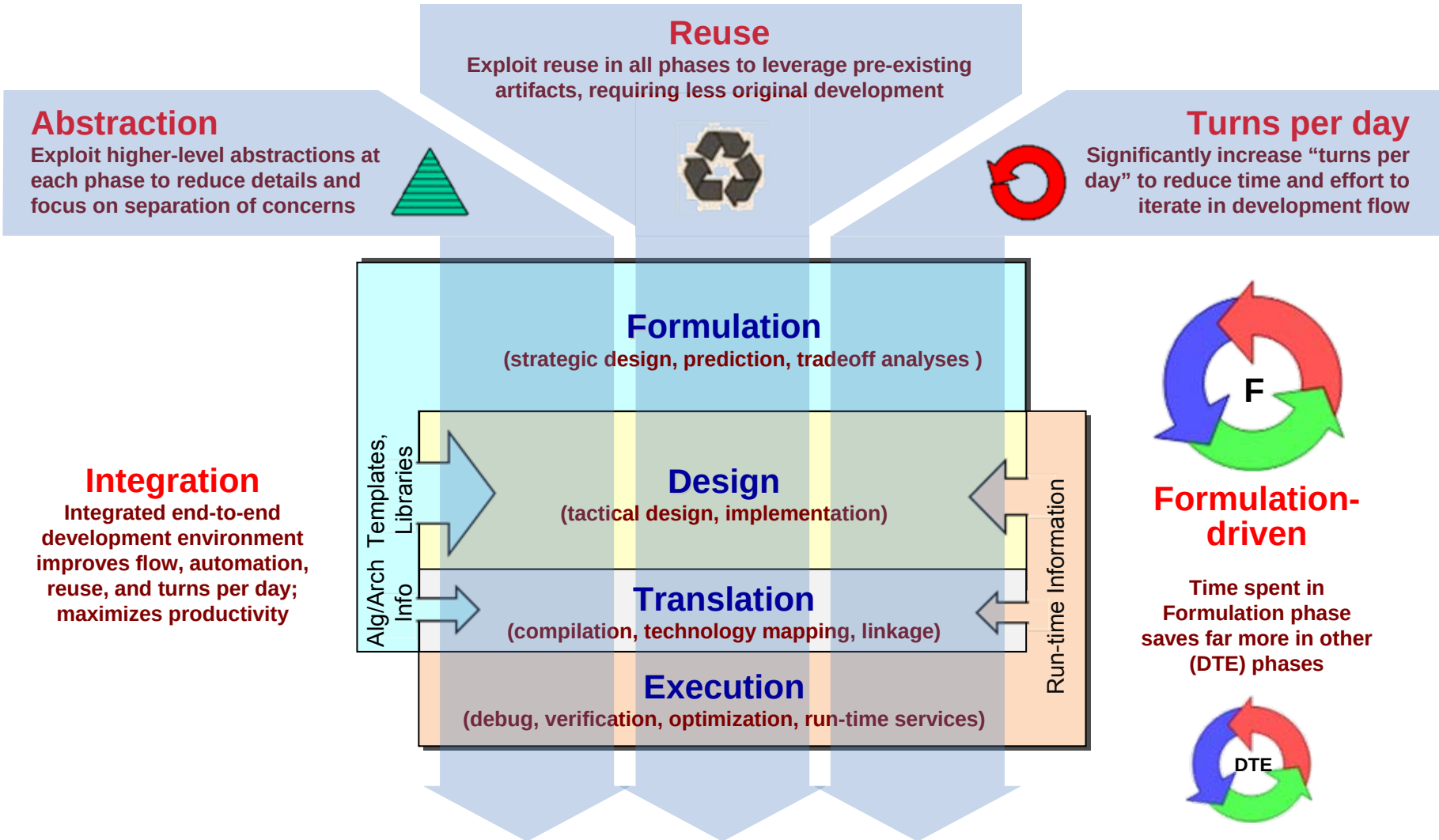


- **Turns per day**
  - Increase ease of design debug & deployment via many more “turns per day”
    - Platform services
    - Firmware
    - High-level abstraction debug





Integrated Research Roadmap for Proposed DARPA Program



Projected productivity impact on order of 20×

Conclusions

Future of RC

- Determined by outcome of two Reformations
  - Architecture Reformation – news is very encouraging
    - RMC inherently superior to FMC devices (CPU, GPU, Cell, etc.) when performance, energy, & versatility are all paramount (e.g. HPC & HPEC)
  - Application Reformation – outcome is TBD
    - Historically, more complex to target apps to RMC devices
    - For RC to become a full-fledged paradigm of computing, must overcome major challenges in app dev productivity
  - Each reform relates to half of productivity ratio
    - Utility of technology vs. Cost of development

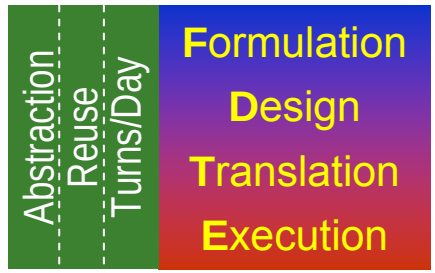
$$\Psi = \frac{U}{C}$$

- **Excel with *U***
- **Compete on *C***

- Predicting the future
  - R&D success with application reformation for RC?
    - **YES**: Development costs drop, user domain vastly expands, big impact! ☺
    - **NO**: Remains as niche technology, perhaps also appliance technology

Conclusions

- RC making inroads in ever-broadening areas
  - HPC and HPEC; from satellites to supercomputers!
- As with any new field, early adopters are brave at heart
  - Face challenges with design methods, tools, apps, systems, etc.
  - Fragmented technologies with gaps and proprietary limitations
- Research & technology challenges abound
  - Productivity, device/system arch., FT, RTR, PR, etc.
  - CHREC sites & partners leading key R&D projects



- Industry/university collaboration is critical to meet challenges
  - Incremental, evolutionary advances will *not* lead to ultimate success
  - Researchers must take more risks, explore & solve tough problems
  - Industry & government as partners, catalysts, tech-transfer recipients

Acknowledgements

We express our gratitude for support of CHREC by

- National Science Foundation
  - Program managers & assistants, center evaluator, panel reviewers
- CHREC Industry and Government Partners
  - >30 members holding >40 memberships in 2008
- University administrations @ CHREC sites
  - University of Florida
  - Brigham Young University
  - George Washington University
  - Virginia Tech
- Equipment and tools vendors providing support
  - Aldec, Altera, Celoxica, Cray, DRC, Gedae, GiDEL, Impulse, Intel, Mellanox, Nallatech, SGI, SRC, Synplicity, Voltaire, XDI, Xilinx

Thanks for Listening! ☺

- For more info:
  - www.chrec.org
  - george@chrec.org
- Questions?