



The End of CMOS Scaling will be Good for Space Computing

Fault Tolerant Spaceborne Computing
Employing New Technologies
May 29, 2008

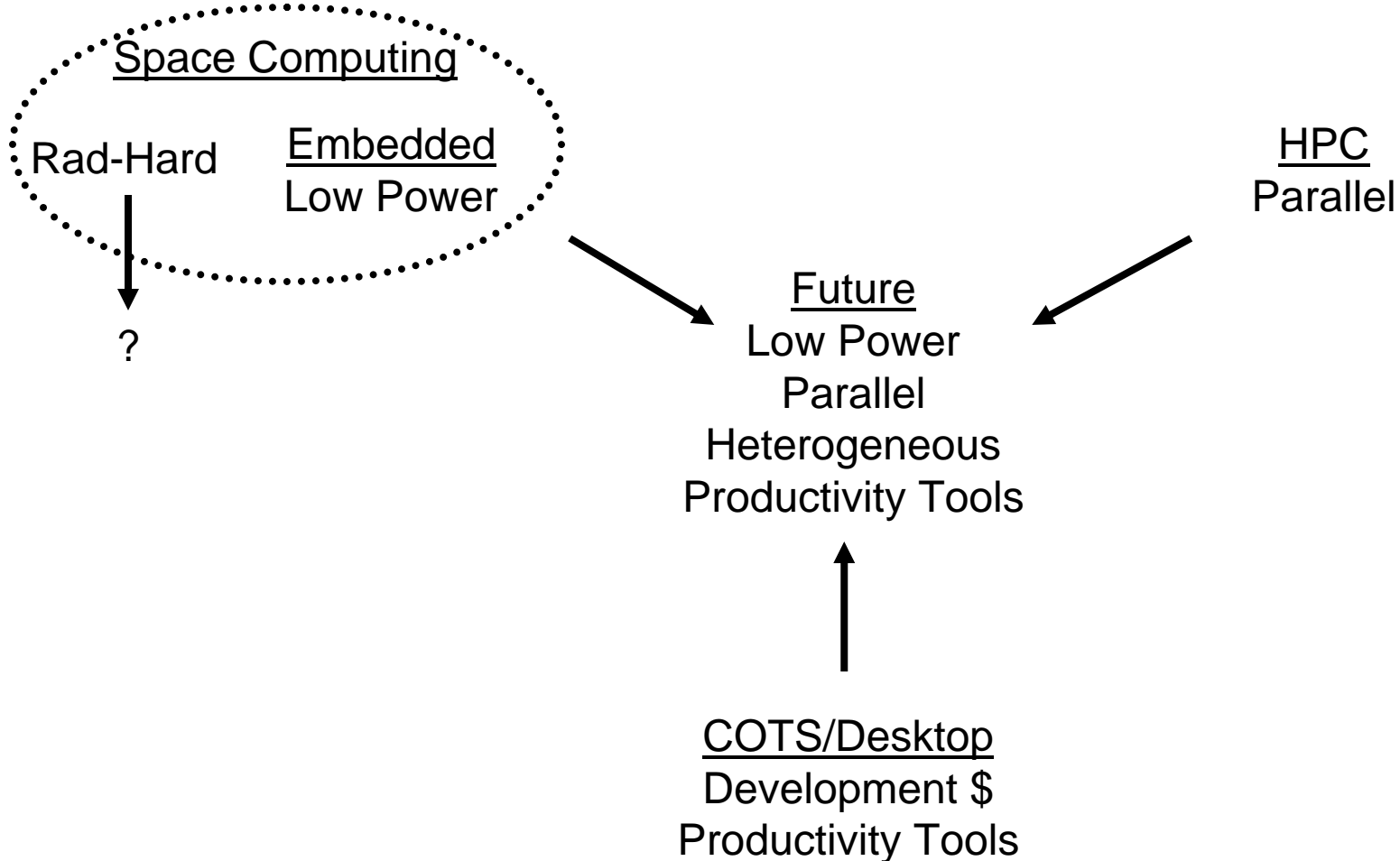
Sandia National Laboratories

Erik DeBenedictis (Sandia)

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

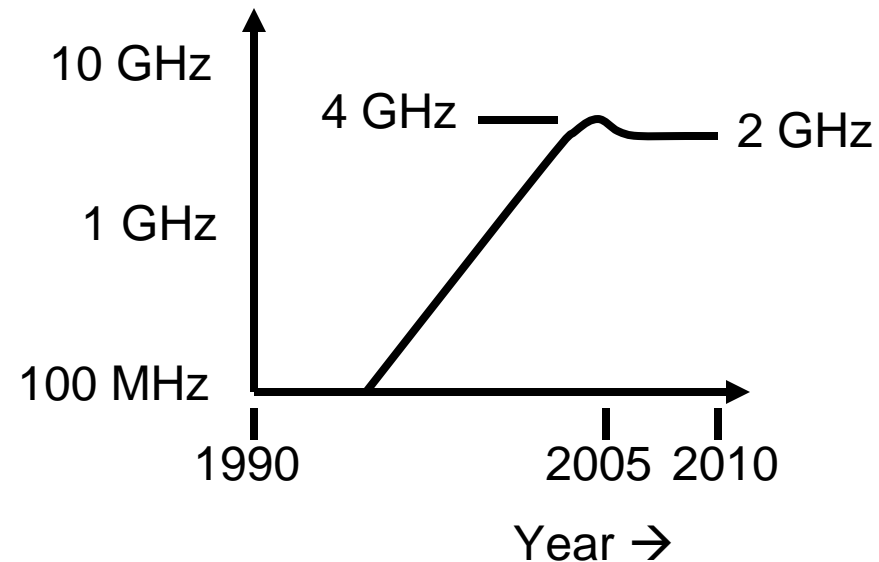


Overview



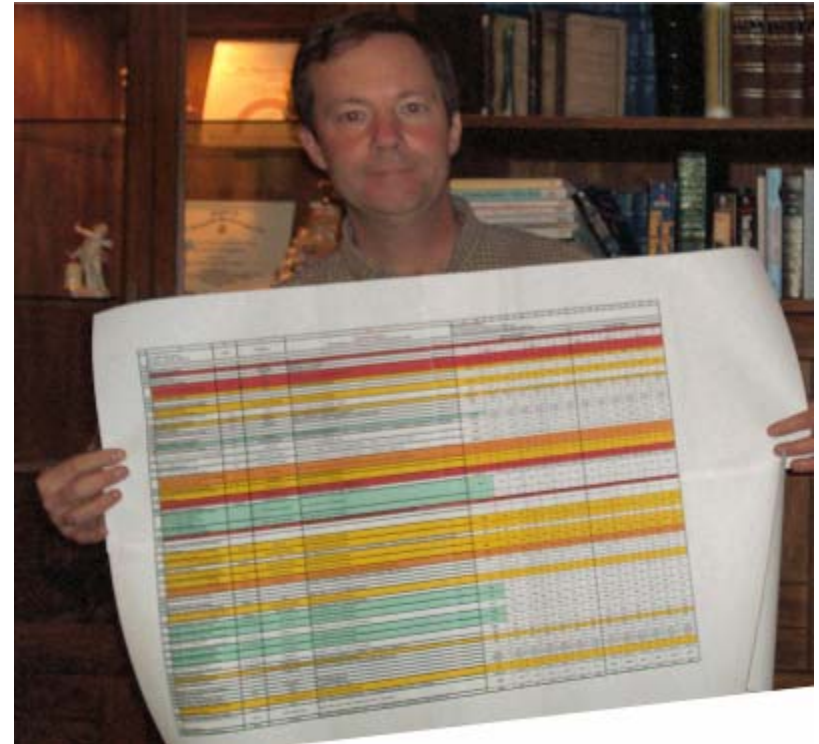
Clock Rate Flat Lined

- Clock rate flat lined a couple years ago, as vendors put excess resources into multiple cores
- This is a historical fact and evident to everybody, so there is little reason to comment on the cause
- However, it has profound architectural consequences (later slide)

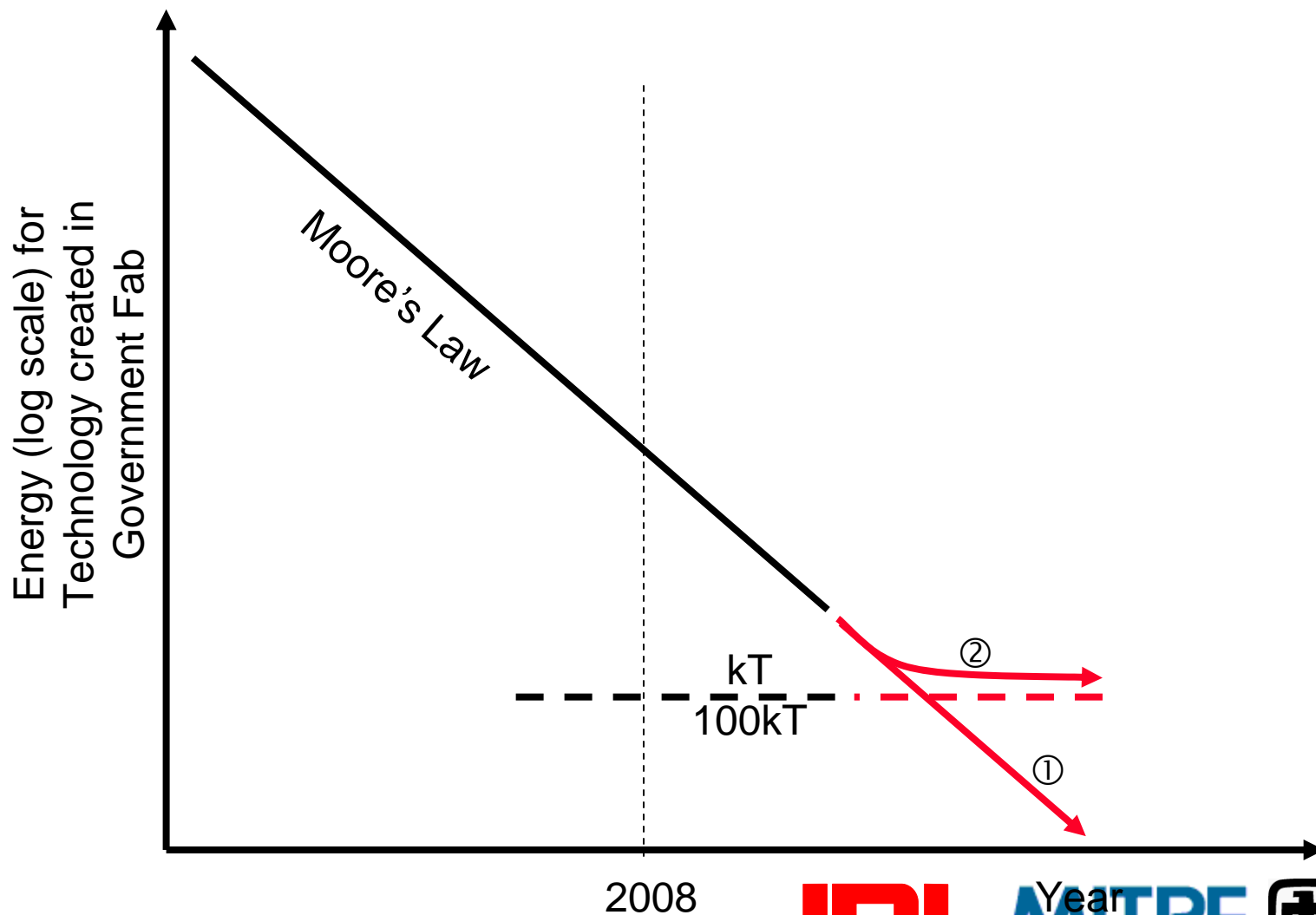


ITRS Process Integration Spreadsheet

- **Big Spreadsheet**
 - Columns are years
 - Rows are 100+ transistor parameters
 - Manual entry of process parameters by year
 - Excel computes operating parameters
 - Extra degrees of freedom go to making Moore's Law smooth – not the best computers



kT Limit Moderates Optimism for Perpetual Exponential Growth



Industry's Plans International Technology Roadmap for Semiconductors



2008 ITRS Update ORTC [Konigswinter Germany ITRS ITWG Plenary]

A.Allan, Rev 2,

[notes on IRC/CTSG More Moore, More than Moore, Beyond CMOS 04/04/08]

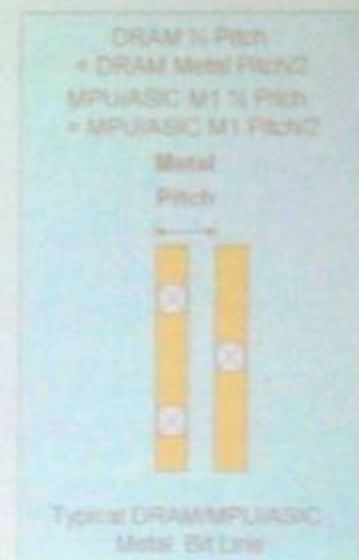


Industry's Plans

2008 - Unchanged

2008 Update Definition of the Half Pitch

[No single-product "node" designation; DRAM half-pitch still M1a driver; however, other product technology trends may be drivers on individual TWD tables]

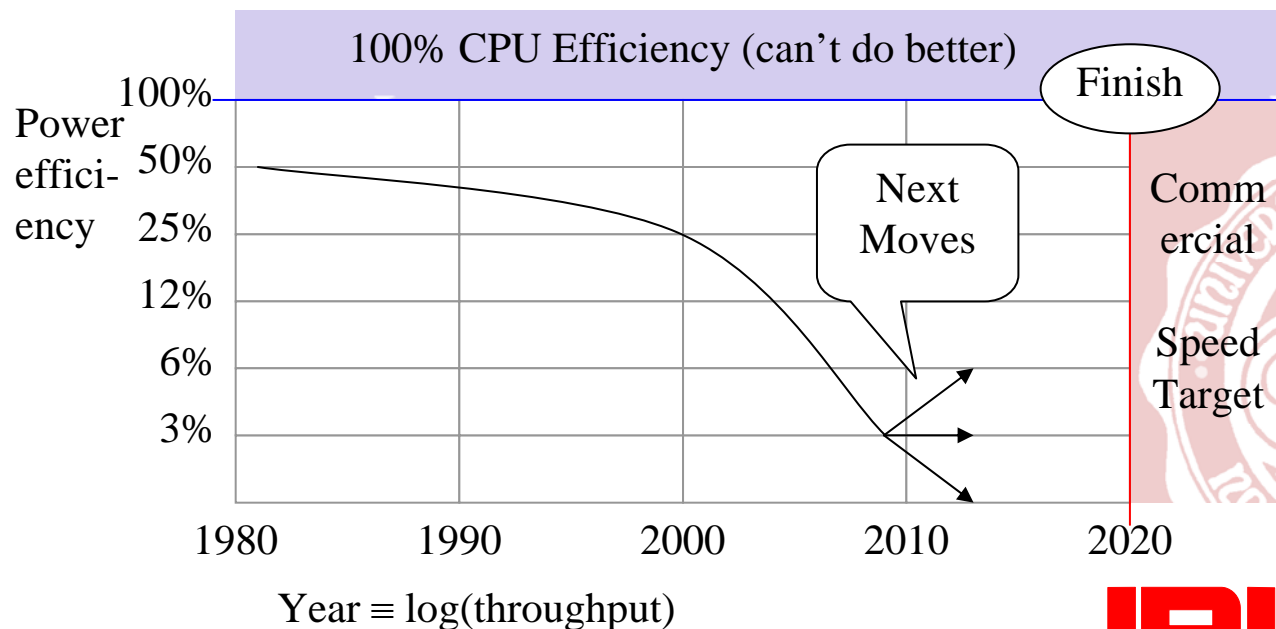


Source: 2007 ITRS - Equip. Technology Fig. 2

© 2007 SEMATECH. All rights reserved.

The Architecture Game

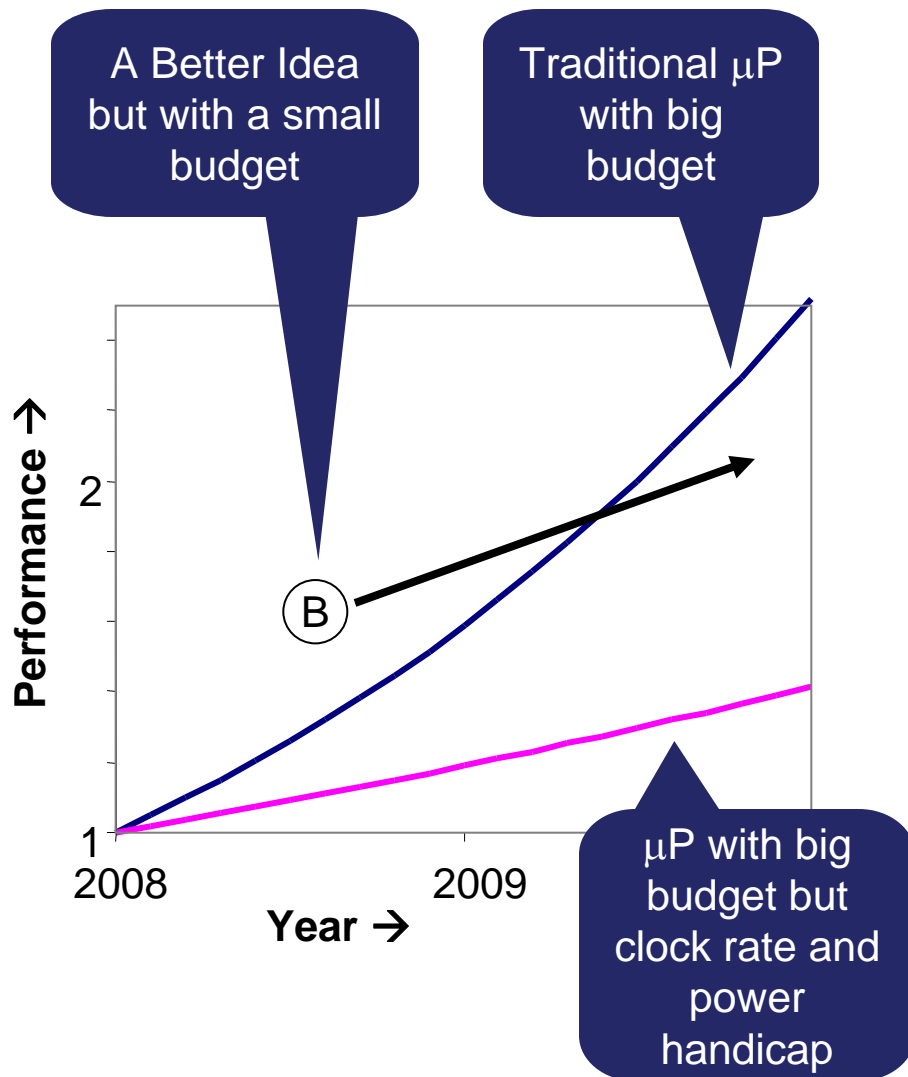
- This is my diagram from a paper to illustrate CMOS architecture in light of CMOS scaling limits
- [Discuss]



Next Moves:

- ↗ Switch to Vector Arch.
- ↗ Switch to SIMD Arch.
- ↗ Add Coprocessor
- ↗ Scale Linewidth
- Increase Parallelism
- ↘ Increase Cache
- ↘ More Superscalar
- ↘ Raise Vdd and Clk

Special Architectures Go Mainstream



• Conclusions

– Mainstream and embedded technology will become more similar

- Power
- Parallelism

– Architectures will become more special purpose

- General systems may be comprised of multiple special purpose sections



EXOCHI: Architecture and Programming Environment for A Heterogeneous Multi-core Multithreaded System

Perry H. Wang¹, Jamison D. Collins¹, Gautham N. Chinya¹,
Hong Jiang², Xinmin Tian³, Milind Girkar³, Nick Y. Yang²,
Guei-Yuan Lueh², and Hong Wang¹

Microarchitecture Research Lab, Microprocessor Technology Labs, Intel Corporation¹

Graphics Architecture, Chipset Group, Intel Corporation²

Intel Compiler Lab, Software Solutions Group, Intel Corporation³

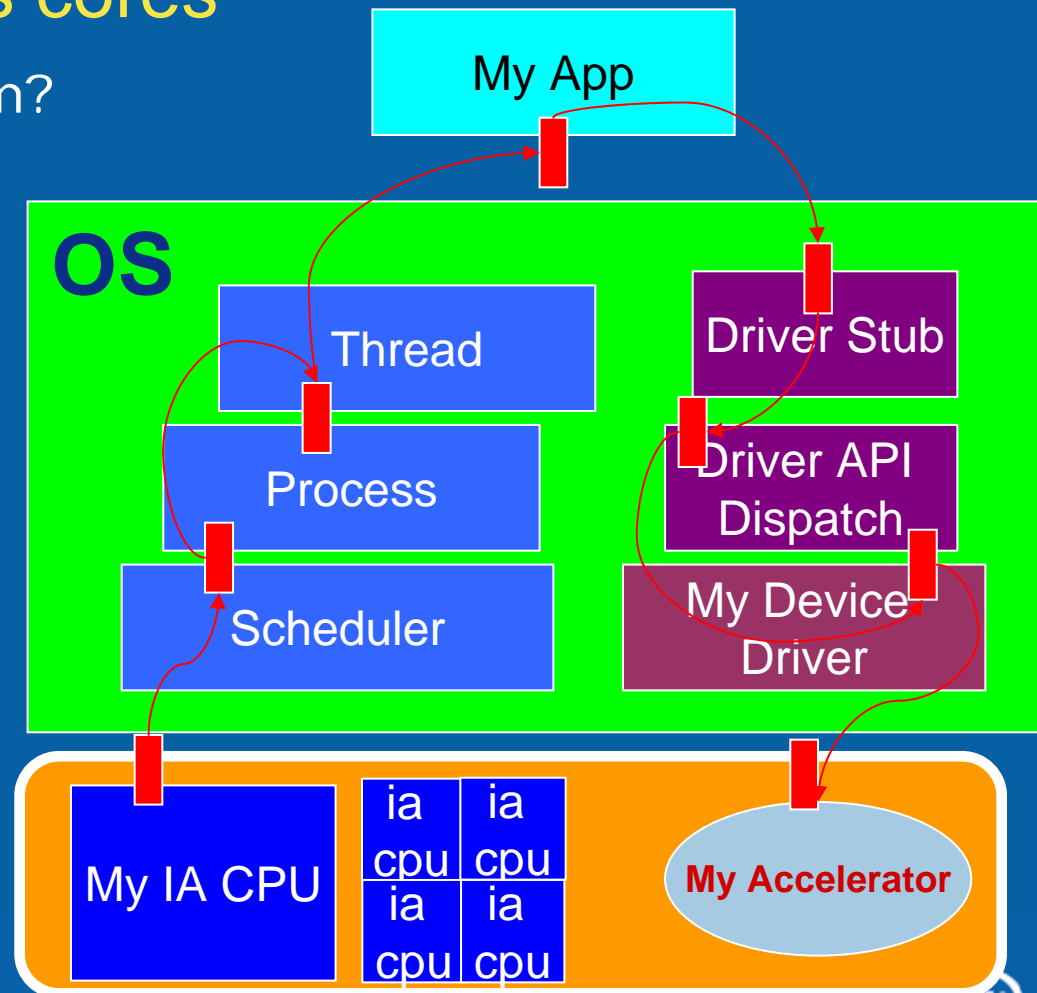


Motivation

The following 5 Viewgraphs sent by Jamison Collins with permission to post

Future mainstream microprocessors will likely integrate heterogeneous cores

- How will we *program* them?
- Map computation to driver / abstraction API
- Unfamiliar development / debugging flow
- OS / driver overheads
- Accelerator in distinct memory space



CHI Programming Environment

Compiler

- Modified front-end pragmas
 - Fork/join
 - Producer/consumer
- Generates fat binaries

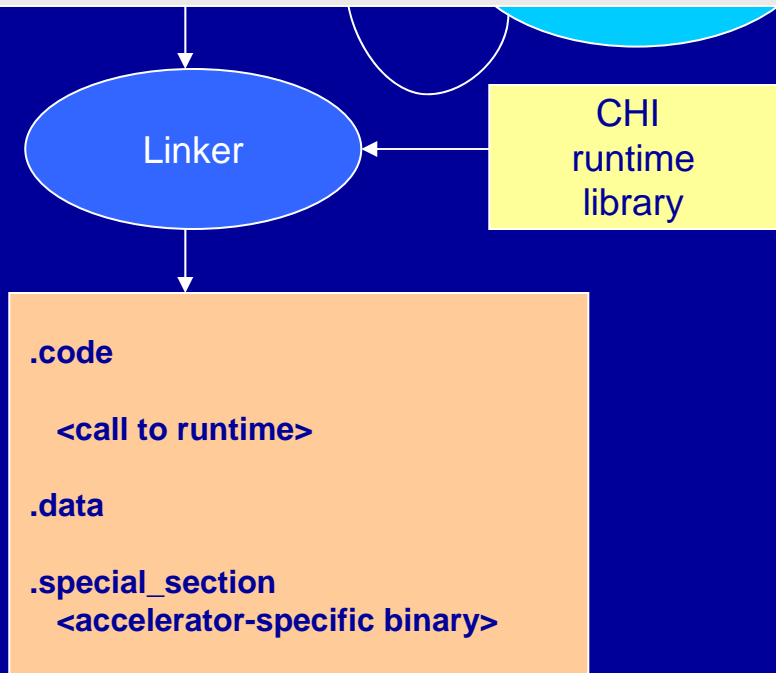
```
#pragma omp parallel target(targetISA) [clause[,]clause...]  
    structured-block
```

Where clause can be any of the following:

```
    firstprivate(variable-list)  
    private(variable-list)  
    shared(variable-ptr-list)  
    descriptor(descriptor-ptr-list)  
    num_threads(integer-expression)  
    master_nowait
```

CHI runtime

- Multi-shredding: User-level threading
- Extensible to multiple types of heterogeneous cores
 - E.g. Intel GMA X3000
 - E.g. A data streaming systolic array accelerator for communication



IA Look-n-Feel: Development and Debugging

The screenshot displays the Microsoft Visual Studio IDE with the 'Debug' window open. The 'Debug' window shows the following assembly code:

```
__asm
{
    shl.1.w vr1 = i, 3
    ld.8.dw [vr2..vr9] = (A_desc, vr1, 0)
    ld.8.dw [vr10..vr17] = (B_desc, vr1, 0)
    add.8.dw [vr18..vr25] = [vr2..vr9], [vr10..vr17]
    st.8.dw (C_desc, vr1, 0) = [vr18..vr25]
}
```

The 'GRF Register View' window is also visible, showing the 'Address' field and the 'Evaluate' button. The status bar at the bottom indicates 'Building workspace: (100%)'.

IA Look-n-Feel: Compilation and Execution

```
Visual Studio 2005 Command Prompt

C:\accelerator_exoskeleton\linear>compile
C:\accelerator_exoskeleton\linear>icl /Qopenmp /mGLOB_use_msasm ShredLibX.lib /I..\include linearfilter.c
Intel(R) C++ Compiler for applications running on IA-32, Version 10.0.0.1033, November 14, 2008.  Copyright (C) 1985-2006 Intel Corporation.  All rights reserved.
linearfilter.c
Begin compiling X3000 assembly
End compiling X3000 assembly
Microsoft (R) Incremental Linker Version 6.00.50727.42
Copyright (C) Microsoft Corporation.  All rights reserved.
-out:linearfilter.exe
-nodefaultlib:libguide_stats.lib
-nodefaultlib:libguide40_stats.lib
-defaultlib:libguide.lib
ShredLibX.lib
linearfilter.obj
C:\accelerator_exoskeleton\linear>run
C:\accelerator_exoskeleton\linear>linearfilter.exe ..\input\holly.bmp out.bmp
Linearfilter processed!
C:\accelerator_exoskeleton\linear>
```

Spaceborne Computing with Emerging Technologies

- Motivation
 - Greater quantities of data: perform more onboard computing, reduce communications requirements
- Vision
 - **Multiple** computing technologies each used to best advantage
 - Harness advances in semiconductors and nanotech
 - Need hardware interoperability
 - Need software tools to support heterogeneous hardware
- Workshop
 - Target date May 28-30, 2008
 - At Sandia, in and out
 - Immediate target: Inventory resources and set plans for coordination and standards
 - Rad hard processing

**Archival,
Maintainable,
Source Code**

**Fault-Tolerant High-Capability
Computational Subsystem**

**Spacecraft
Control Subsystem**

