Trends in the Infrastructure of Computing

CSCE 190: Computing in the Modern World

Dr. Jason D. Bakos

UNIVERSITY OF SOUTH CAROLINA

My Questions

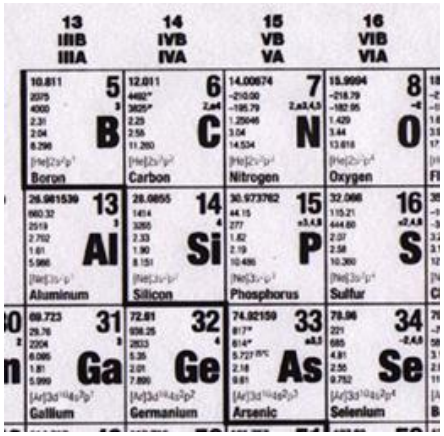
- How do computer processors work?
- Why do computer processors get faster over time?
- What makes one processor faster than another?
  - What type of tradeoffs are involved in processor design?
- What is the relationship between processor performance and how programs are written?
- Is it possible for one processor to achieve higher performance than all other processors, regardless of the type of programs for which it's used?

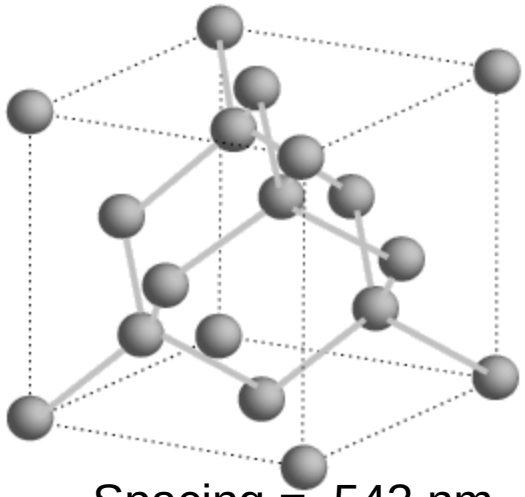
Talk Outline

1. Quick introduction to how computer processors work
2. The role of computer architects
3. CPU design philosophy and survey of state-of-the art CPU technology
4. Coprocessor design philosophy and survey of state-of-art coprocessor technology
5. Reconfigurable computing
6. Heterogeneous computing
7. Brief overview of my research

Semiconductors

- Silicon is a group IV element
- Forms covalent bonds with four neighbor atoms (3D cubic crystal lattice)
  - Si is a poor conductor, but *conduction* characteristics may be altered
  - Add impurities/dopants (replaces silicon atom in lattice):
    - Group V element (phosphorus/arsenic) => 5 valence electrons
      - Leaves an electron free => n-type semiconductor (electrons, negative carriers)
    - Group III element (boron) => 3 valence electrons
      - Borrows an electron from neighbor => p-type semiconductor (holes, positive carriers)

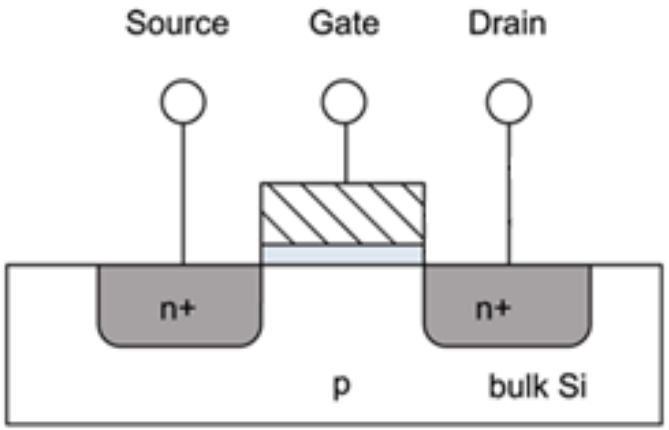




Spacing = .543 nm

MOSFETs

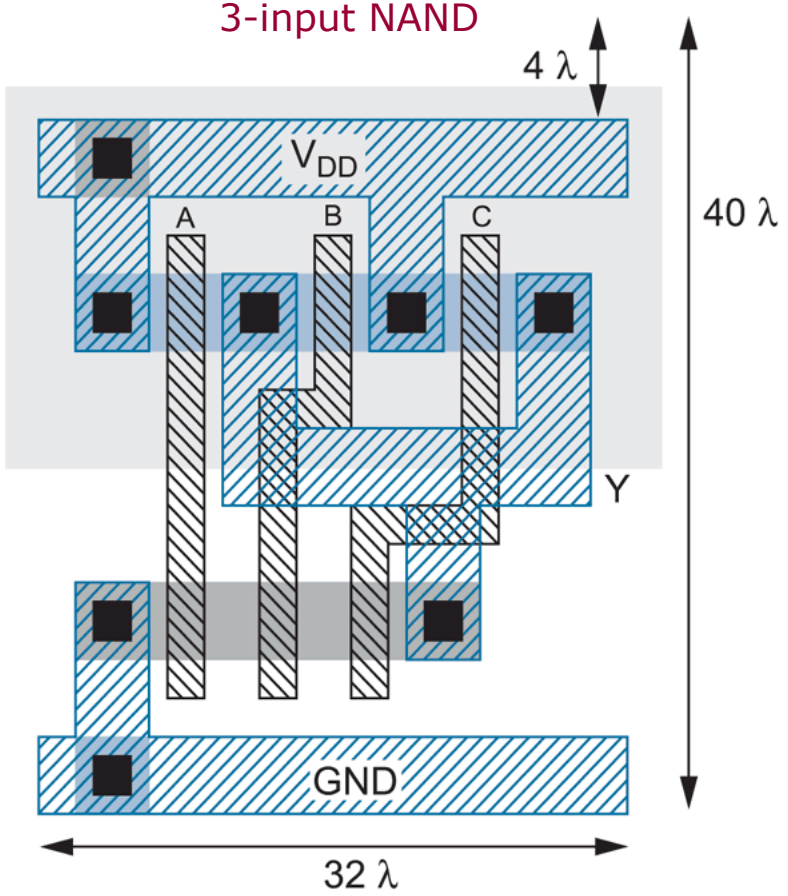
Cut away side view

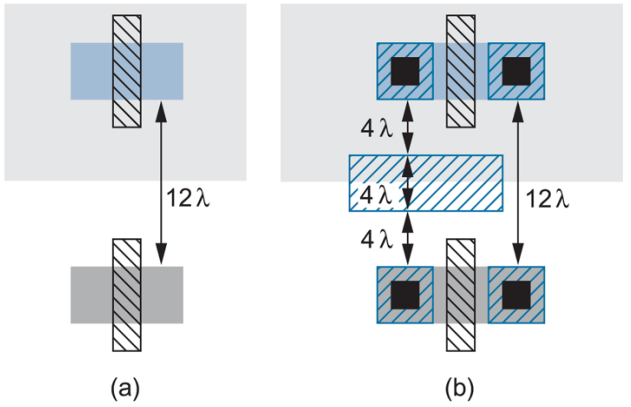


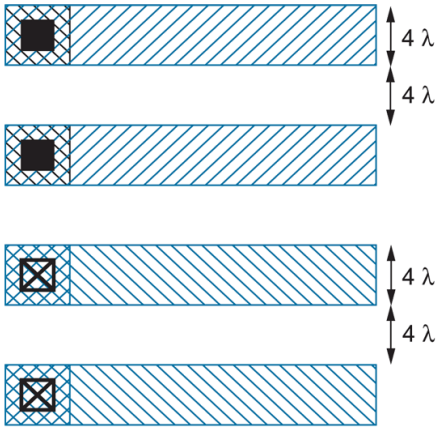
- Metal-*poly*-Oxide-Semiconductor structures built onto substrate
  - *Diffusion*: Inject dopants into substrate
  - *Oxidation*: Form layer of SiO2 (glass)
  - *Deposition* and *etching*: Add aluminum/copper wires

Layout

3-input NAND



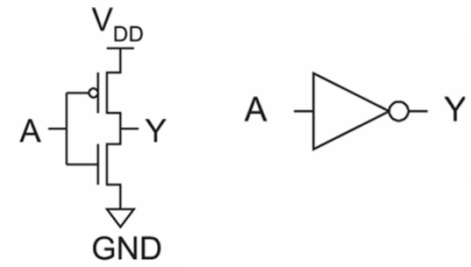




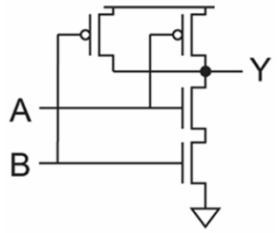
Logic Gates

inv

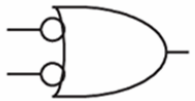
$$Y = \overline{A}$$



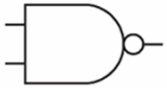
NAND2



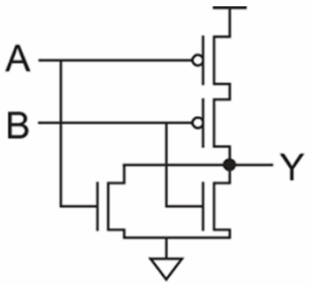
$$Y = \overline{A} + \overline{B}$$



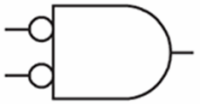
$$Y = \overline{A \bullet B}$$



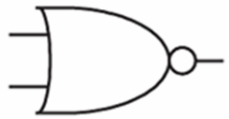
NOR2



$$Y = \overline{A} \bullet \overline{B}$$



$$Y = \overline{A + B}$$



Logic Synthesis

- Behavior:
  - S = A + B
  - Assume A is 2 bits, B is 2 bits, C is 3 bits

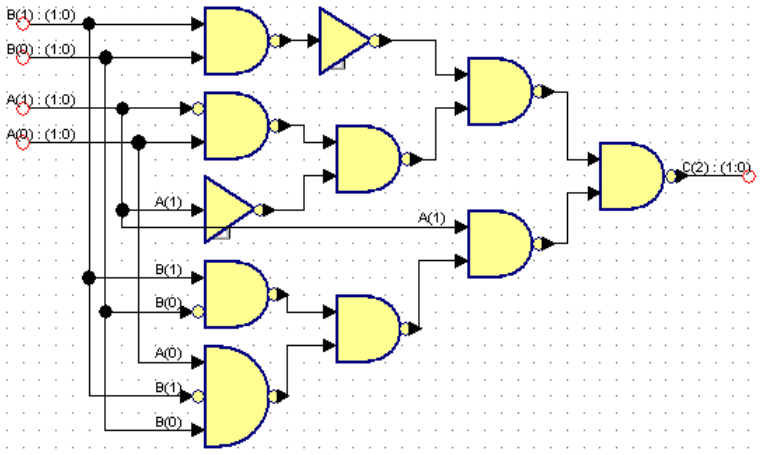
A	B	C
00 (0)	00 (0)	000 (0)
00 (0)	01 (1)	001 (1)
00 (0)	10 (2)	010 (2)
00 (0)	11 (3)	011 (3)
01 (1)	00 (0)	001 (1)
01 (1)	01 (1)	010 (2)
01 (1)	10 (2)	011 (3)
01 (1)	11 (3)	100 (4)
10 (2)	00 (0)	010 (2)
10 (2)	01 (1)	011 (3)
10 (2)	10 (2)	100 (4)
10 (2)	11 (3)	101 (5)
11 (3)	00 (0)	011 (3)
11 (3)	01 (1)	100 (4)
11 (3)	10 (2)	101 (5)
11 (3)	11 (3)	110 (6)

$$C_2 = \overline{A_1}A_0B_1B_0 + A_1\overline{A_0}B_1\overline{B_0} + A_1\overline{A_0}B_1B_0 + A_1A_0\overline{B_1}B_0 + A_1A_0B_1\overline{B_0} + A_1A_0B_1B_0$$

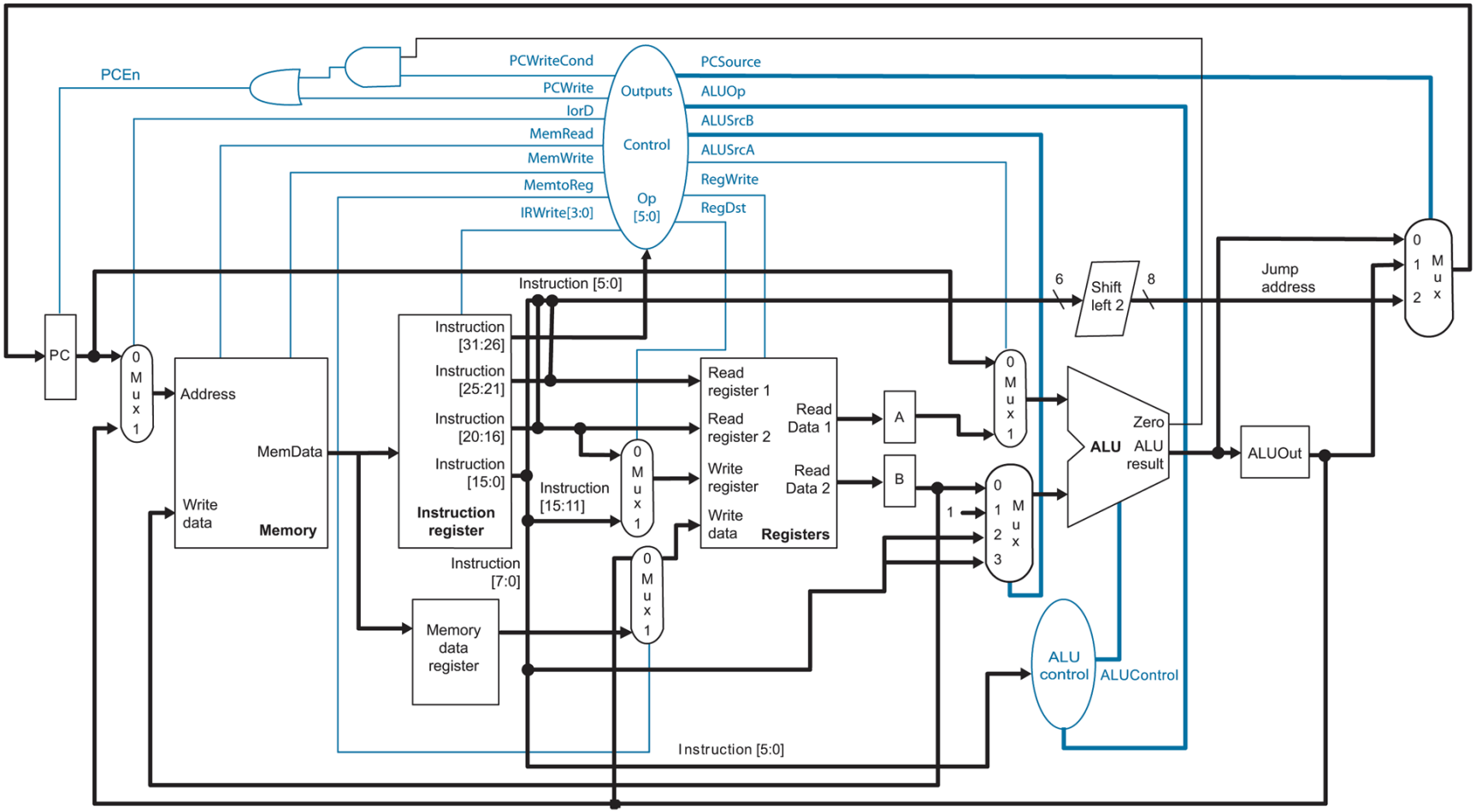
$$C_2 = B_1B_0(\overline{A_1}A_0 + A_1\overline{A_0} + A_1A_0) + A_1B_1\overline{B_0}(\overline{A_0} + A_0) + A_1A_0\overline{B_1}B_0$$

$$C_2 = B_1B_0(\overline{A_1}A_0 + A_1(\overline{A_0} + A_0)) + A_1B_1\overline{B_0} + A_1A_0\overline{B_1}B_0$$

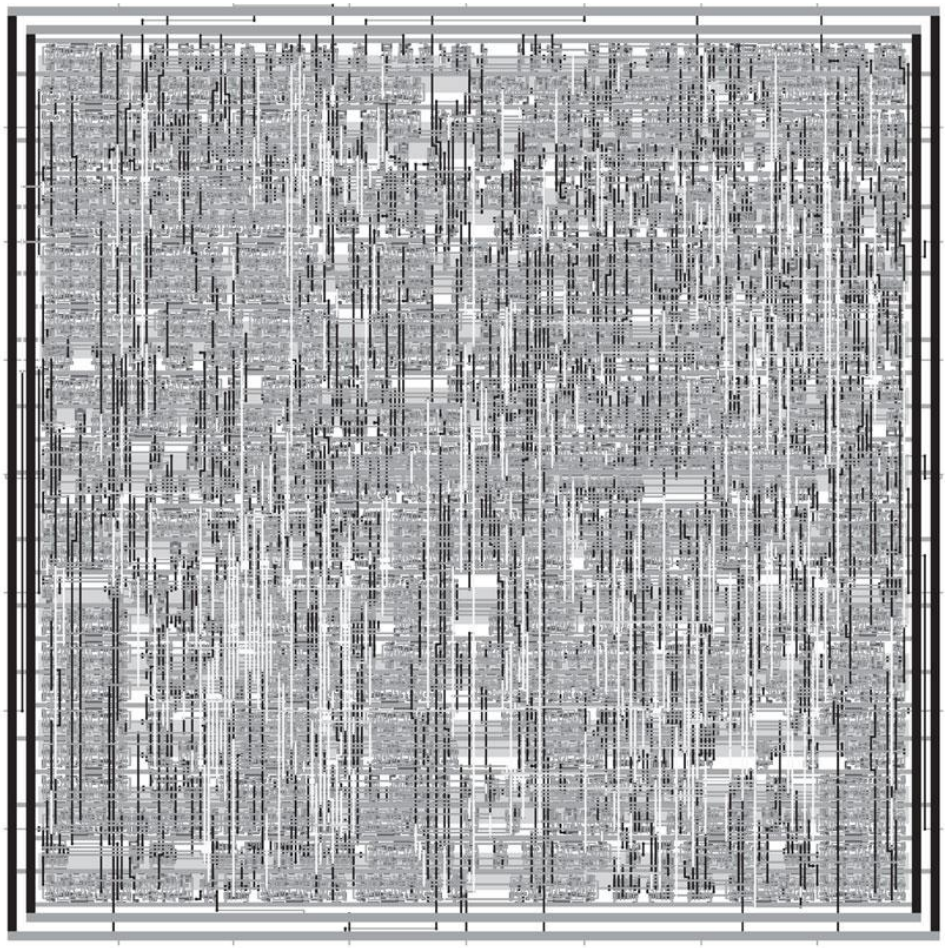
$$C_2 = B_1B_0(\overline{A_1}A_0 + A_1) + A_1(B_1\overline{B_0} + A_0\overline{B_1}B_0)$$



Microarchitecture

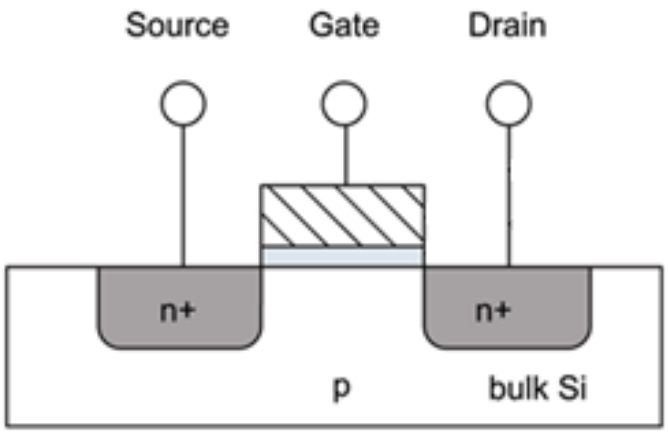


Synthesized and P&R'ed MIPS Architecture



Feature Size

- Shrink minimum feature size…
  - Smaller L decreases carrier time and increases current
  - Therefore, W may also be reduced for fixed current
  - C_g, C_s, and C_d are reduced
  - Transistor switches faster (~*linear* relationship)



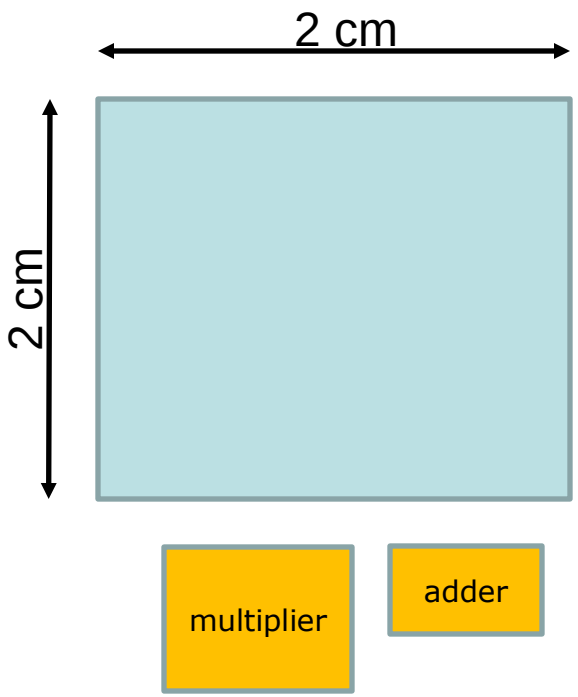
Minimum Feature Size

Year	Processor	Speed	Transistor Size	Transistors
1982	i286	6 - 25 MHz	1.5 μm	~134,000
1986	i386	16 – 40 MHz	1 μm	~270,000
1989	i486	16 - 133 MHz	.8 μm	~1 million
1993	Pentium	60 - 300 MHz	.6 μm	~3 million
1995	Pentium Pro	150 - 200 MHz	.5 μm	~4 million
1997	Pentium II	233 - 450 MHz	.35 μm	~5 million
1999	Pentium III	450 – 1400 MHz	.25 μm	~10 million
2000	Pentium 4	1.3 – 3.8 GHz	.18 μm	~50 million
2005	Pentium D	2 threads/package	.09 μm	~200 million
2006	Core 2	2 threads/die	.065 μm	~300 million
2008	Core i7 "Nehalem"	8 threads/die	.045 μm	~800 million
2011	"Sandy Bridge"	16 threads/die	.032 μm	~1.2 billion

Year	Processor	Speed	Transistor Size	Transistors
2008	NVIDIA Tesla	240 threads/die	.065 μm	1.4 billion
2010	NVIDIA Fermi	512 threads/die	.040 μm	3 billion

The Role of Computer Architects

- Given a blank slate (silicon substrate)
- Budget: 2 billion transistors:
  - Transistors ⇔ Area



- Choose your components:

Component	Cost
Control Logic and Cache	
Cache	50K transistors/1KB + 10K transistors/port
Out-of-order instruction scheduler and dispatch	200K transistors/core
Speculative execution	400K transistors/core
Branch predictor	200K transistors/core
Functional Units	
Integer and load/store units	100K transistors/unit/core
Floating-point unit	1M transistors/unit/core
Vector/SIMD floating-point unit	100M transistors/64b width/unit/core

The Role of Computer Architects

- Problem:
  - Cost of fabricating one state-of-the-art (32 nm) 4 cm^2 chip:
    - Hundreds of millions of dollars
  - Additional cost of fabricating hundreds of thousands of copies of same chip:
    - FREE
  - Strategy for staying in business:
    - Sell LOTS of chips
  - How to sell LOTS of chips:
    - Make sure it works well for a wide range of applications
    - Make sure it's easy to program

- Most modern CPUs are designed in a similar way
  - Maximize performance of each thread, target small number of threads

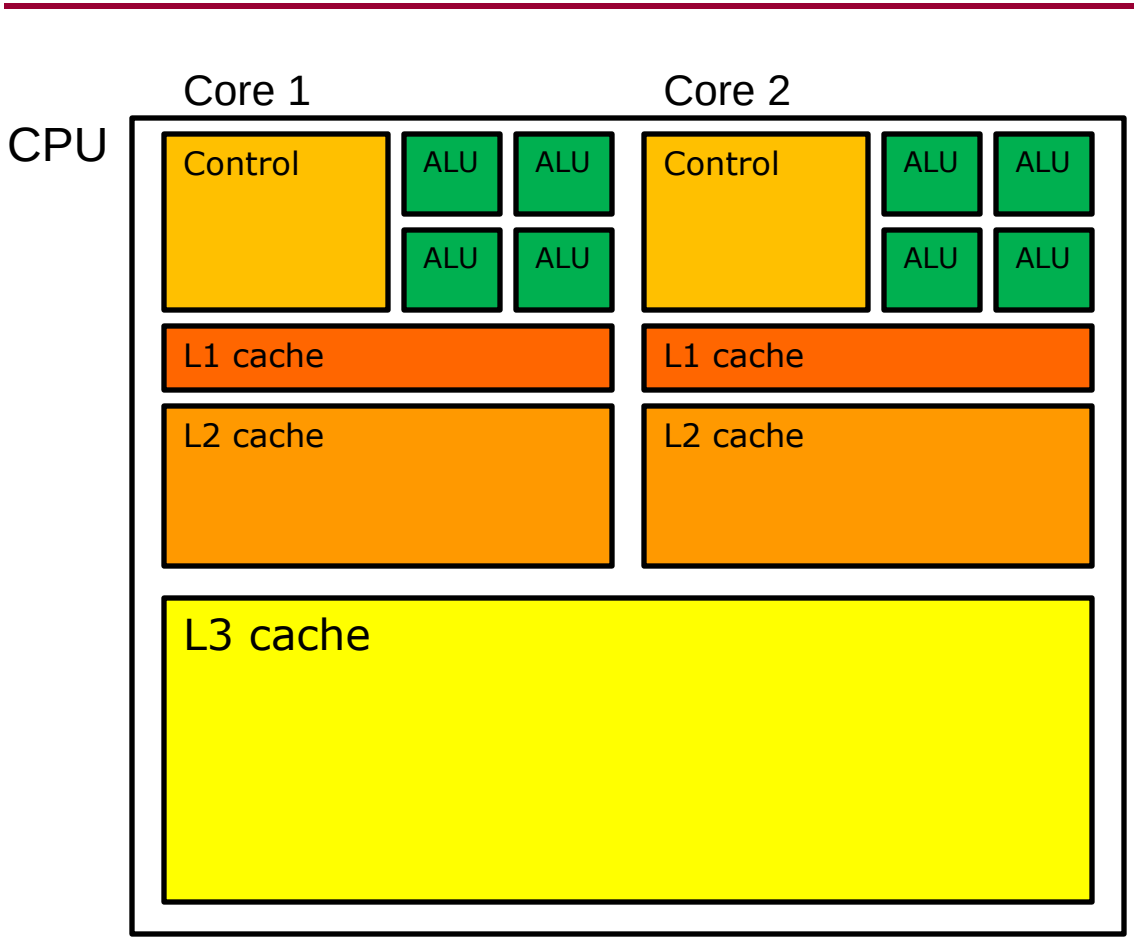
CPU Design Philosophy

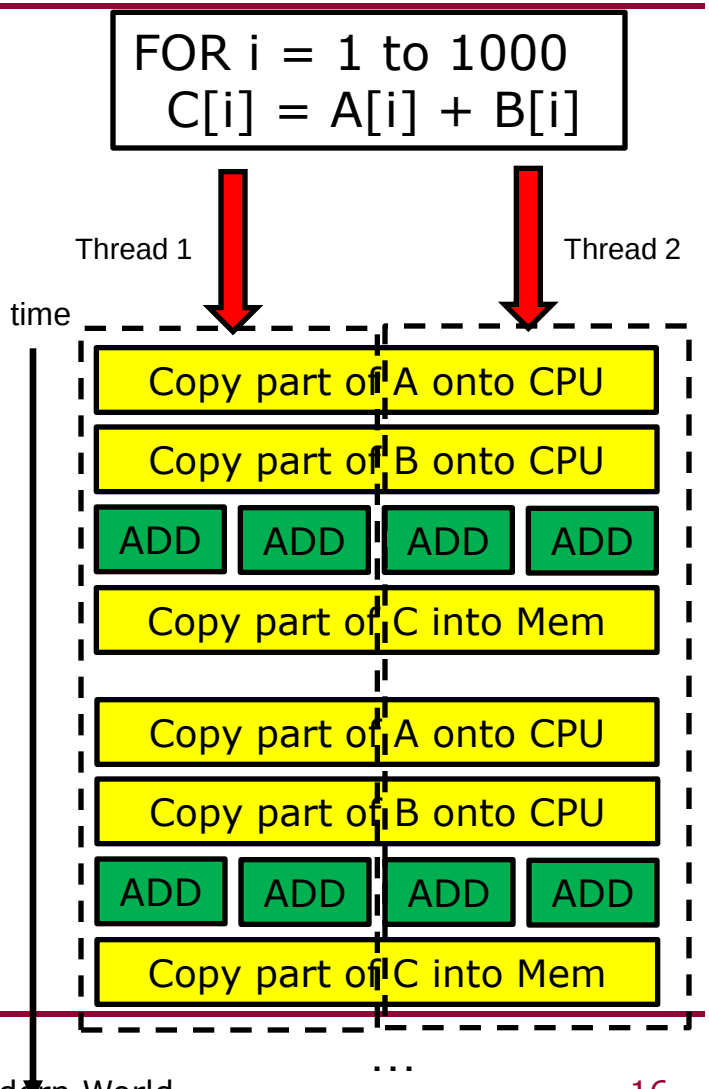
- Processors consist of three main componets:
  - Control logic
  - Cache
  - Functional units

- Premise:
  - Most code (non scientific) isn't written to take advantage of multiple cores
  - Devote real estate budget to maximizing performance of each thread
    - Control logic: reorders operations within a thread, speculatively execution
    - Cache: reduces memory delays for different access patterns

- CPUs do achieve higher performance when code is written to be multi-threaded
  - But CPUs quickly run out of functional units

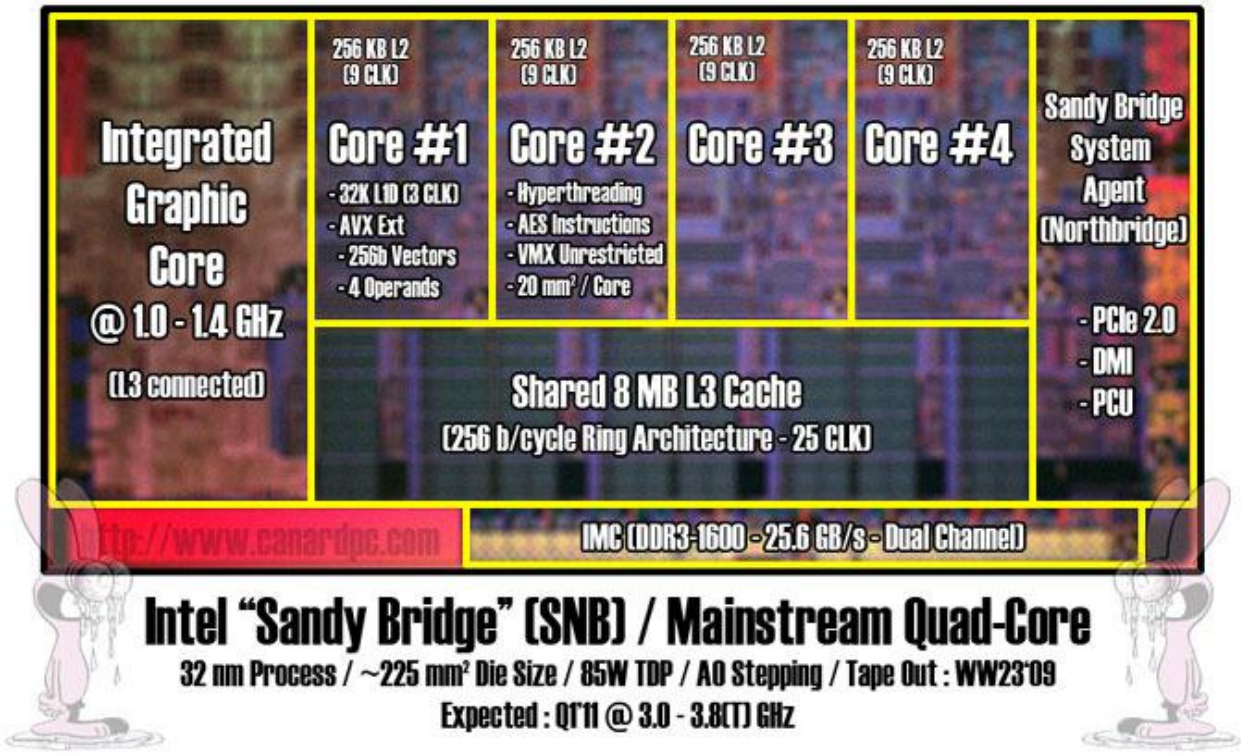
CPU Design


CPU
Core 1
Core 2
Control
ALU
ALU
ALU
ALU
Control
ALU
ALU
ALU
ALU
L1 cache
L1 cache
L2 cache
L2 cache
L3 cache


FOR i = 1 to 1000
C[i] = A[i] + B[i]
Thread 1
Thread 2
time
Copy part of A onto CPU
Copy part of B onto CPU
ADD
ADD
ADD
ADD
Copy part of C into Mem
Copy part of A onto CPU
Copy part of B onto CPU
ADD
ADD
ADD
ADD
Copy part of C into Mem
...

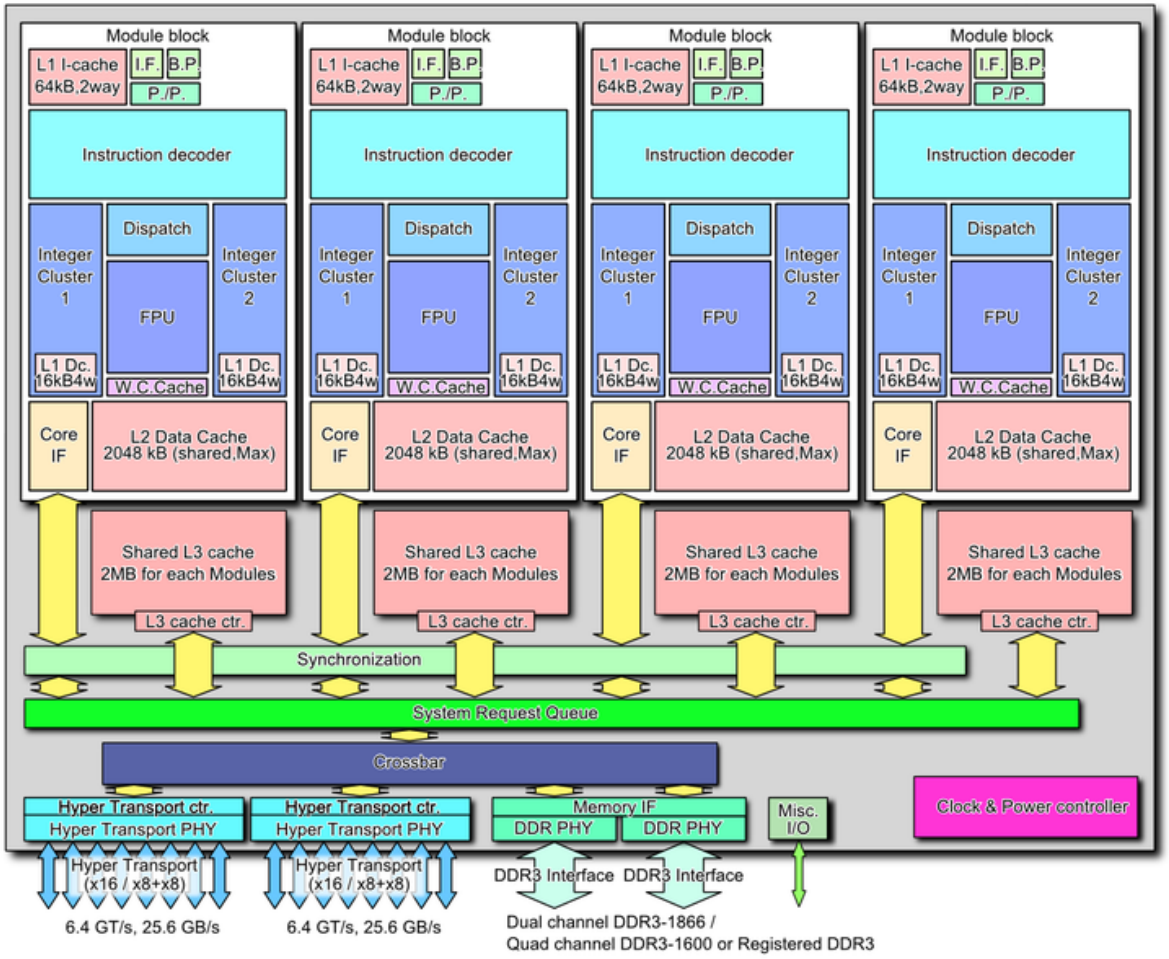
Intel Sandy Bridge Architecture

- Four cores, each core has 256-bit SIMD unit
- Integrated GPU (IGU) with 12 execution units



AMD Bulldozer Architecture

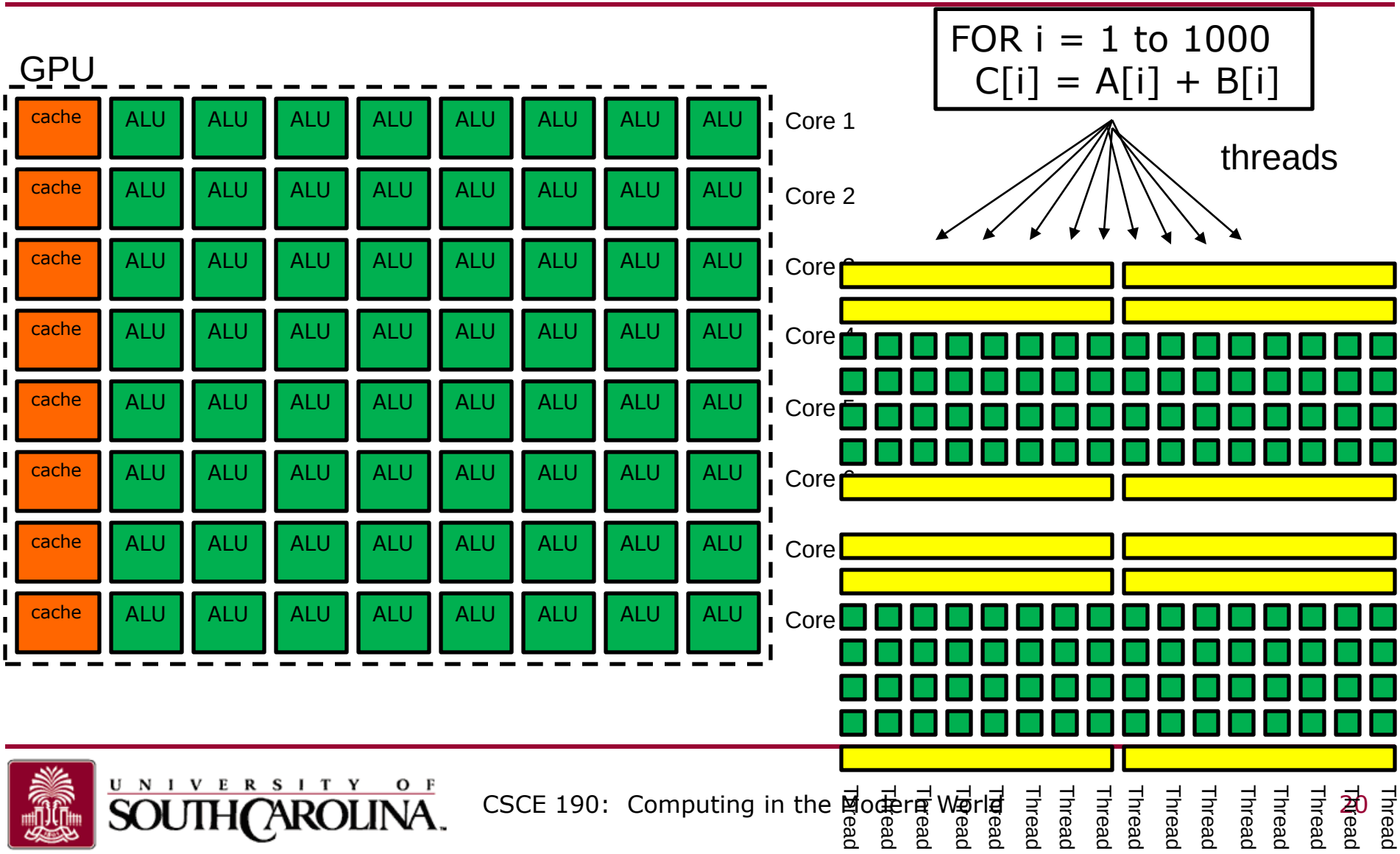
- Next generation AMD architecture
- Designed from scratch
- Four cores, each core:
  - Executes 2 threads with dedicated logic
  - Contains two 128-bit FP multiply-add units
- Decoupled L3 caches



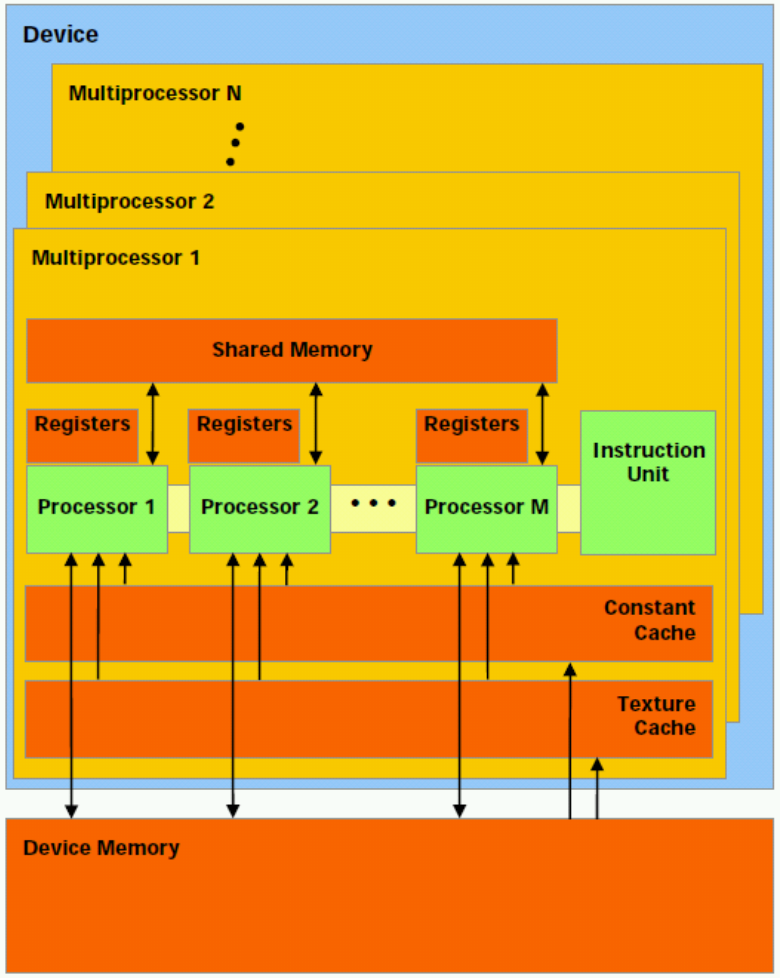
Graphical Processor Units (GPUs)

- Basic idea:
  - Originally designed for 3D graphics rendering
  - Success in gaming market
  - Devote real estate to computational resources
    - As opposed to control and caches to make naïve and general-purpose code run fast
  - Achieves very high performance but:
    - Extremely difficult to program
      - Program must be split into 1000s of threads
    - Only works well for specific types of programs
    - Lacks functionality to manage computer system resources
  - Now widely used for High Performance Computing (HPC)

Co-Processor (GPU) Design

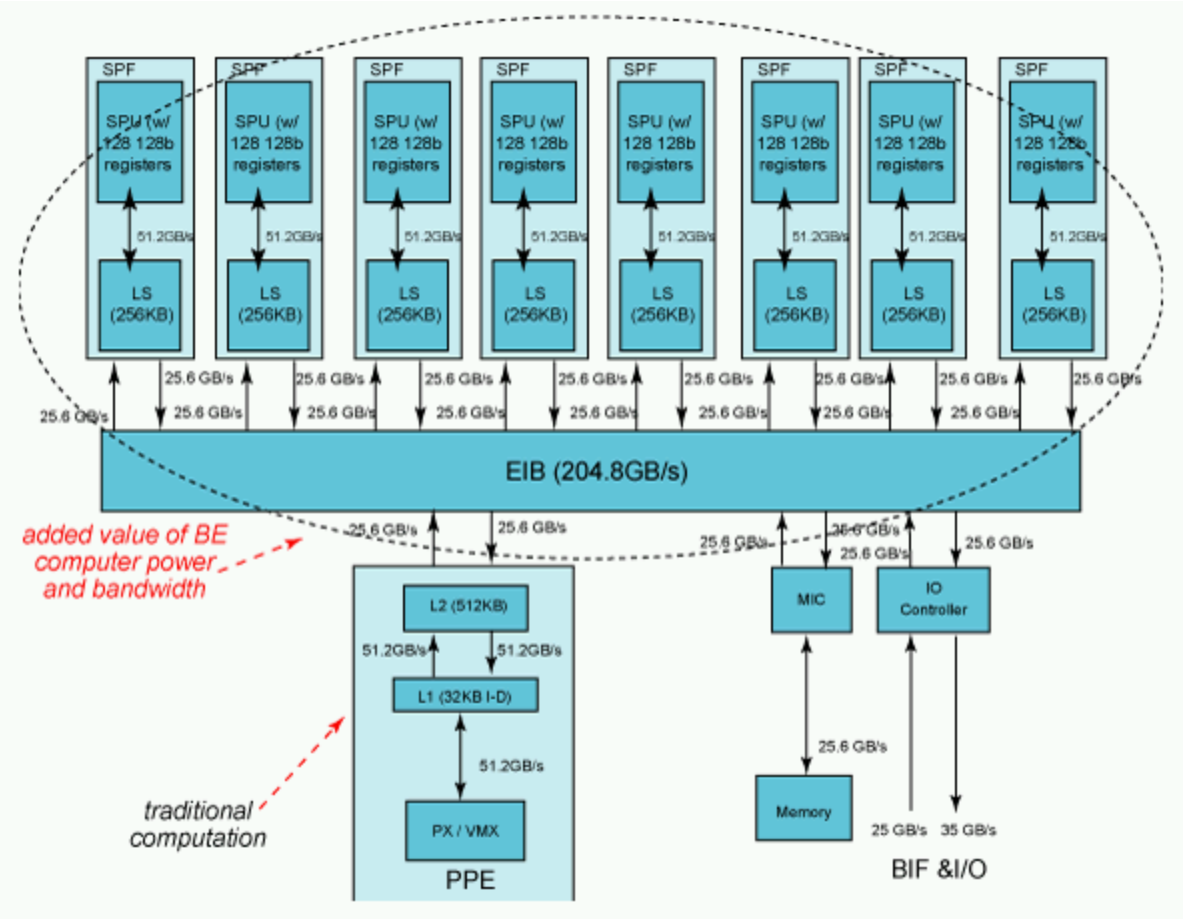


NVIDIA GPU Architecture



- Hundreds of simple processor cores
- Core design:
  - Executes each individual thread very slowly
  - Each thread can only perform one operation at a time
  - No operating system support
  - Able to execute 32 threads at the same time
  - Has 15 cores

IBM Cell/B.E. Architecture



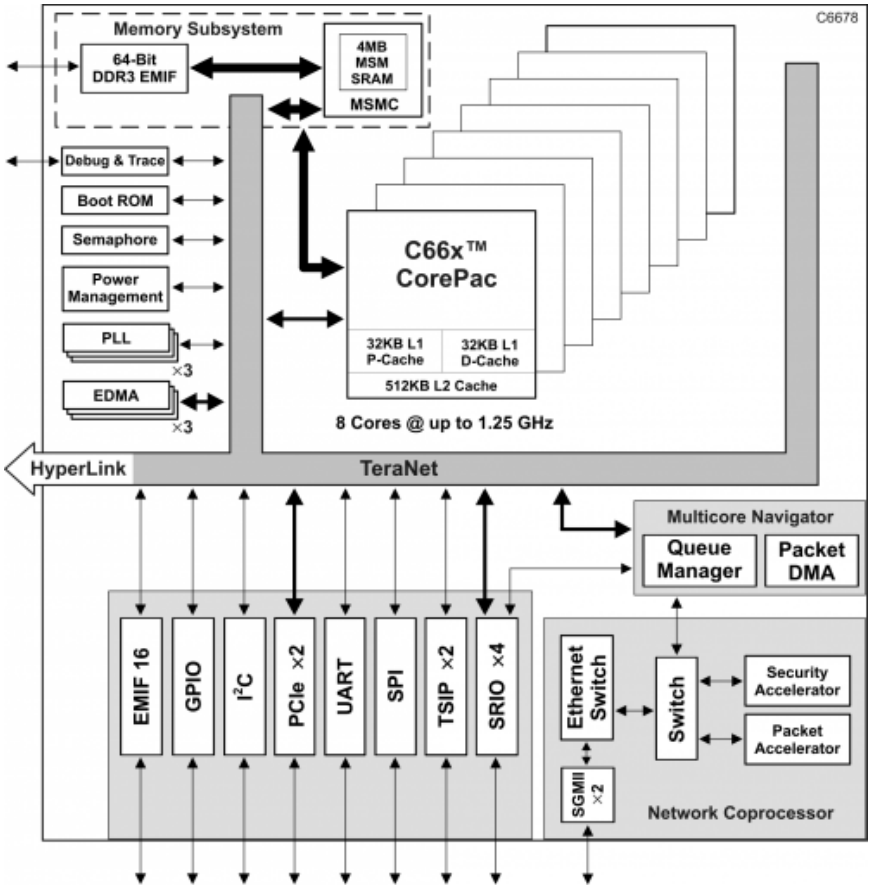
- 1 PPE, 8 SPEs
- Programmer must manually manage 256K memory and threads invocation on each SPE
- Each SPE includes a vector unit like the one on current Intel processors
  - 128 bits wide (4 ops)

Intel MIC Architecture

- PCIe coprocessor card
- Used like a GPU
- "Many Integrated Core"
  - 32 x86 cores, 128 threads
  - 512 bit SIMD units
  - Coherent cache among cores
  - 2 GB onboard memory
  - Uses Intel ISA
  - No operating system?

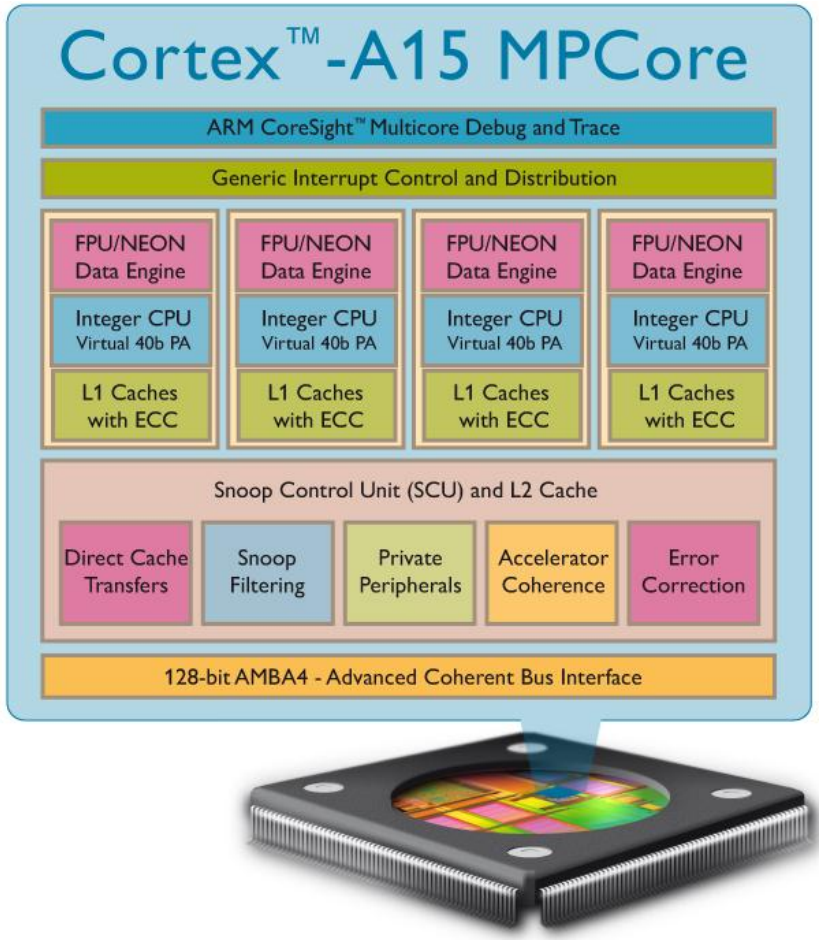
Texas Instruments C66x Architecture

- Originally designed for signal processing
- 8 cores
- Can do floating-point or fixed-point
  - Can do fixed-point much faster
- Possible coprocessor for HPC?



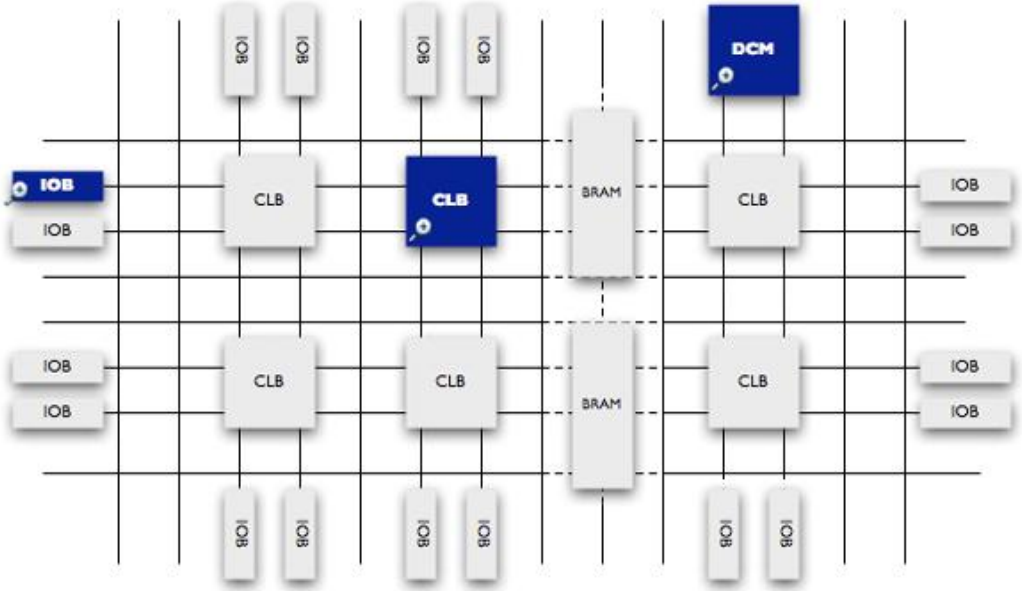
ARM Cortex-A15 Architecture

- Originally designed for embedded (cell phone) computing
- Out-of-order superscalar pipelines
- 4 cores per cluster, up to 2 clusters per chip
- Possible coprocessor for HPC?
  - LOW POWER
    - FLOPS/watt

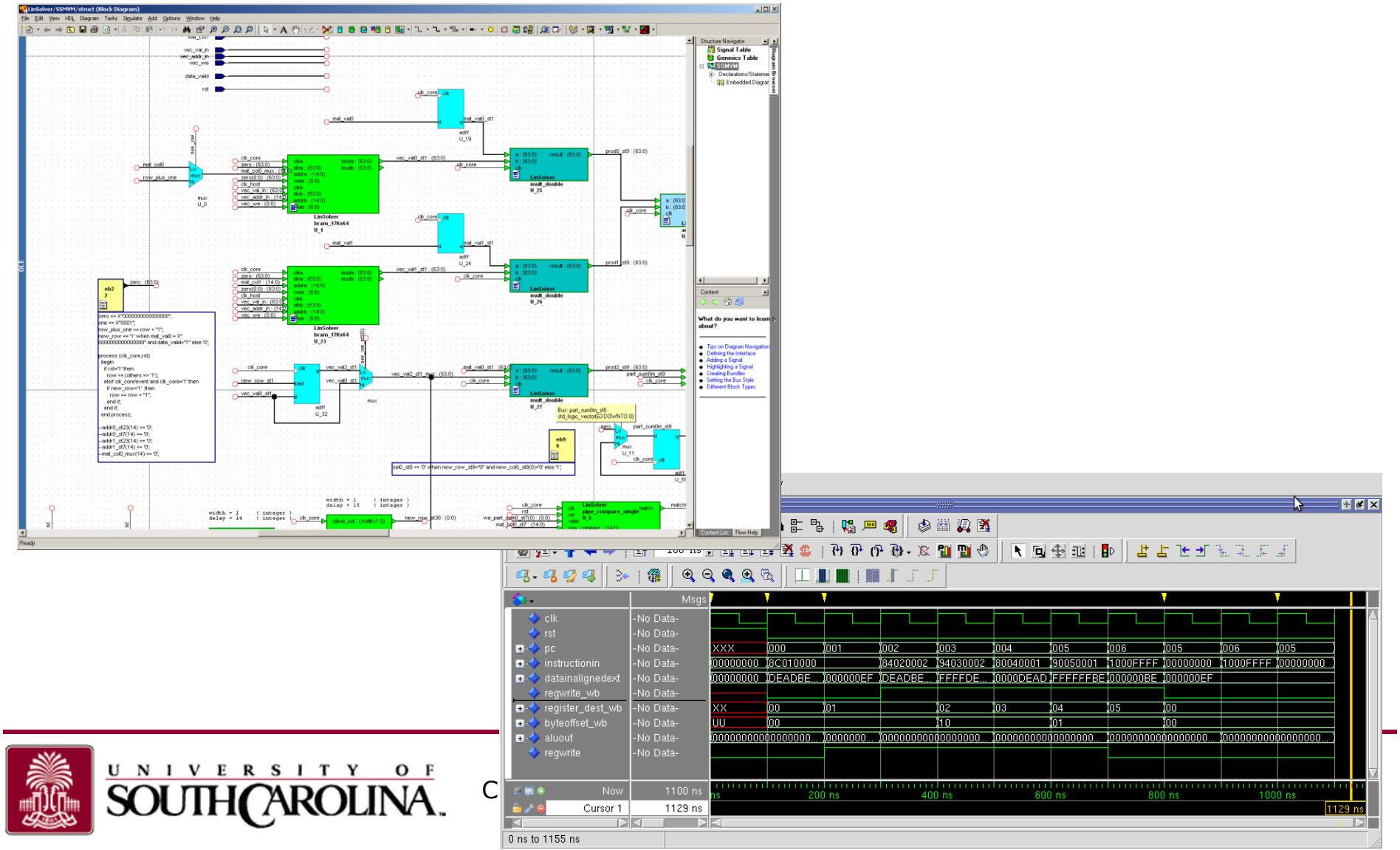


Field Programmable Gate Arrays

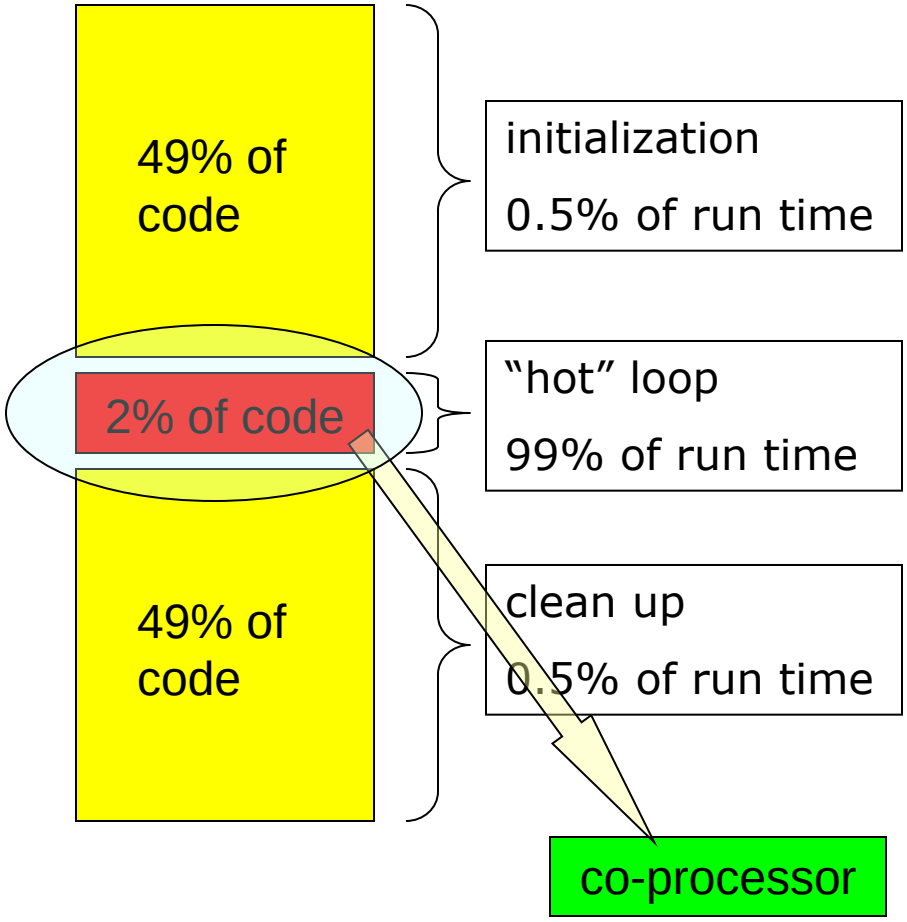
- FPGAs are blank slates that can be electronically reconfigured
- Allows for totally customized architectures
- Drawbacks:
  - More difficult to program than GPUs
  - 10X less logic density and clock speed



Programming FPGAs



Heterogeneous Computing

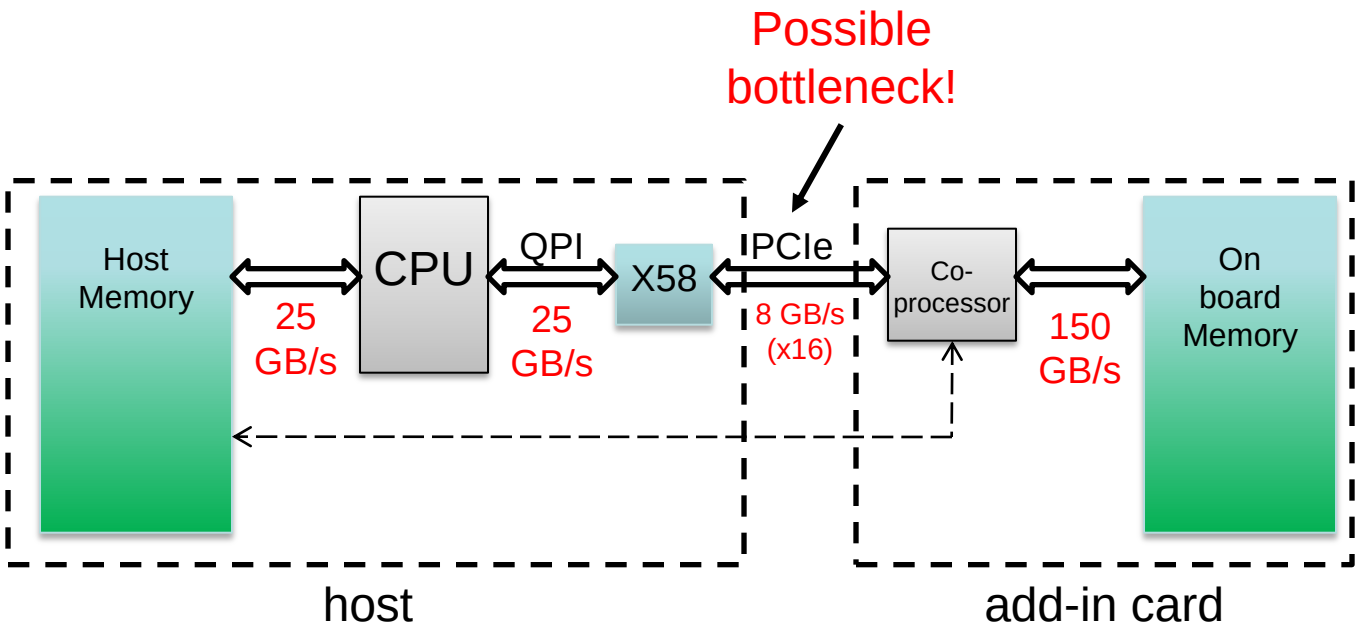


- Combine CPUs and coprocs
- Example:
  - Application requires a week of CPU time
  - Offload computation consumes 99% of execution time

Kernel speedup	Application speedup	Execution time
50	34	5.0 hours
100	50	3.3 hours
200	67	2.5 hours
500	83	2.0 hours
1000	91	1.8 hours

Heterogeneous Computing

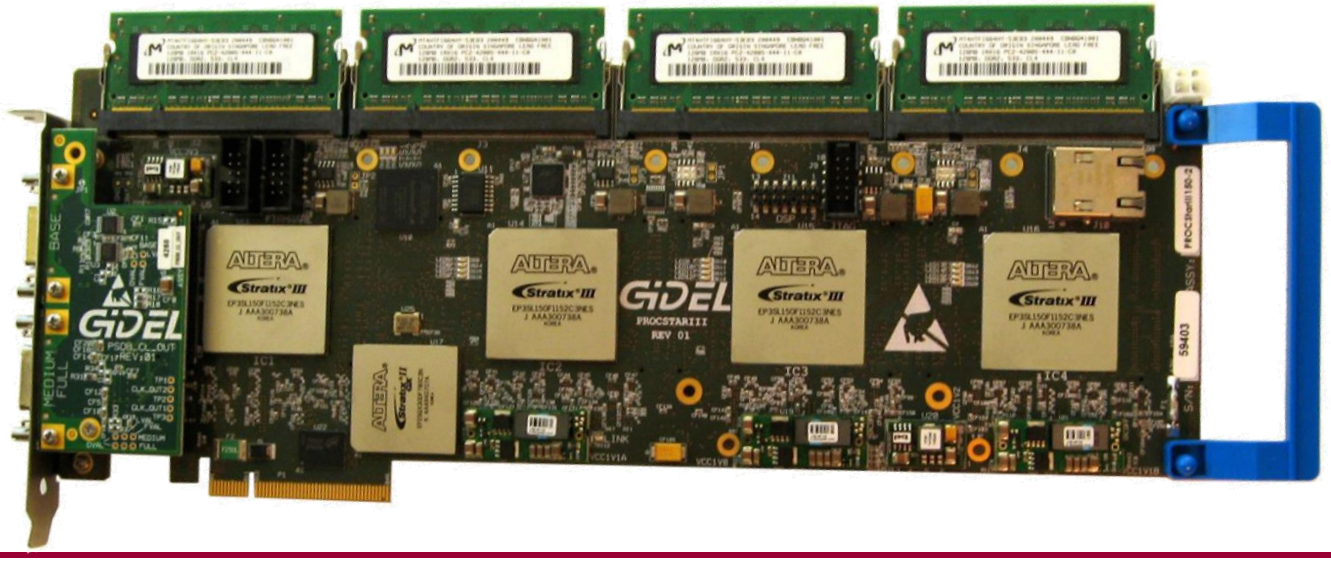
- General purpose CPU + special-purpose processors in one system
- Use coprocessors to execute code that they can execute fast!
  - Allow coprocessor to have its own high speed memory



Heterogeneous Computing with FPGAs

Annapolis Micro Systems WILDSTAR 2 PRO

GiDEL PROCSTAR III

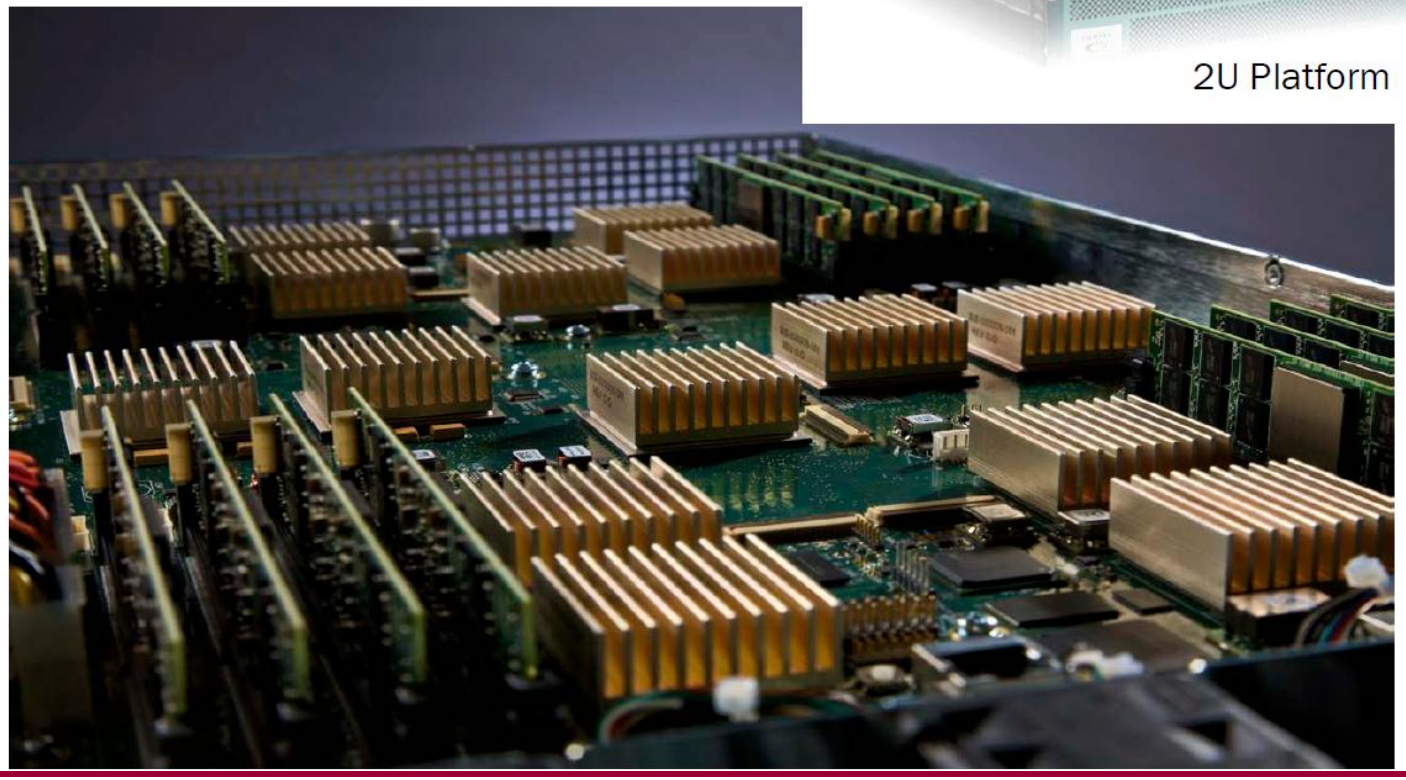


Heterogeneous Computing with FPGAs

Convey HC-1

- Top half of platform is the Coprocessor
- Bottom half is the Intel Motherboard

2U Platform



AMD Fusion Architecture

- Integrate GPU-type coprocessor onto CPU
  - Put a small RADEON GPU onto the GPU
- Allows to accelerate smaller programs than PCIe-connected coprocessor
- Targeted for embedded but AMD hopes to scale to servers

My Research

- Developed custom FPGA coprocessor architectures for:
  - Computational biology
  - Sparse linear algebra
  - Data mining
- Written GPU optimized implementations for:
  - Computational biology
  - Logic synthesis
  - Data mining
  - General purpose graph traversals
- Generally achieve 50X – 100X speedup over general-purpose CPUs

Current Research Goals

- Develop high level synthesis (compilers) for specific types of FPGA-based computers
  - Based on:
    - Custom pipeline-based architectures
    - Multiprocessor soft-core systems on reconfigurable chips (MPSoC)

- Develop code tuning tools for GPU code based on runtime profiling analysis