Special Topics in Advanced Computing Technology: Programming and System Design

by

Dr. Lesley Shannon

Email: lshannon@ensc.sfu.ca

Course Website: http://www.ensc.sfu.ca/~lshannon/courses/ensc894

Simon Fraser University

Slide Set: 0
Date: January 19, 2010

Slide Set Overview

- Course Motivation
- Logistical Details
- Course Components (handouts review)
- Lecture Topics

Computing Technology is everywhere

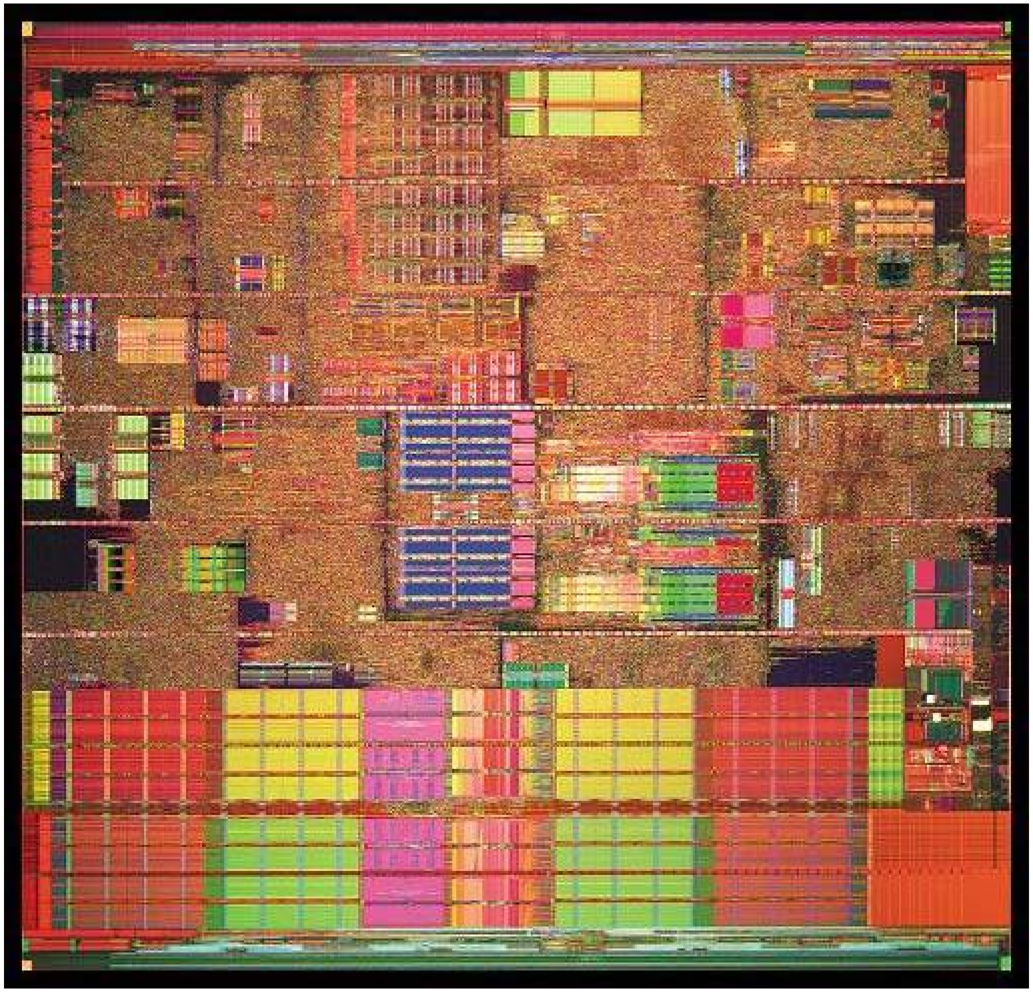
&

it’s one of the largest growing job markets

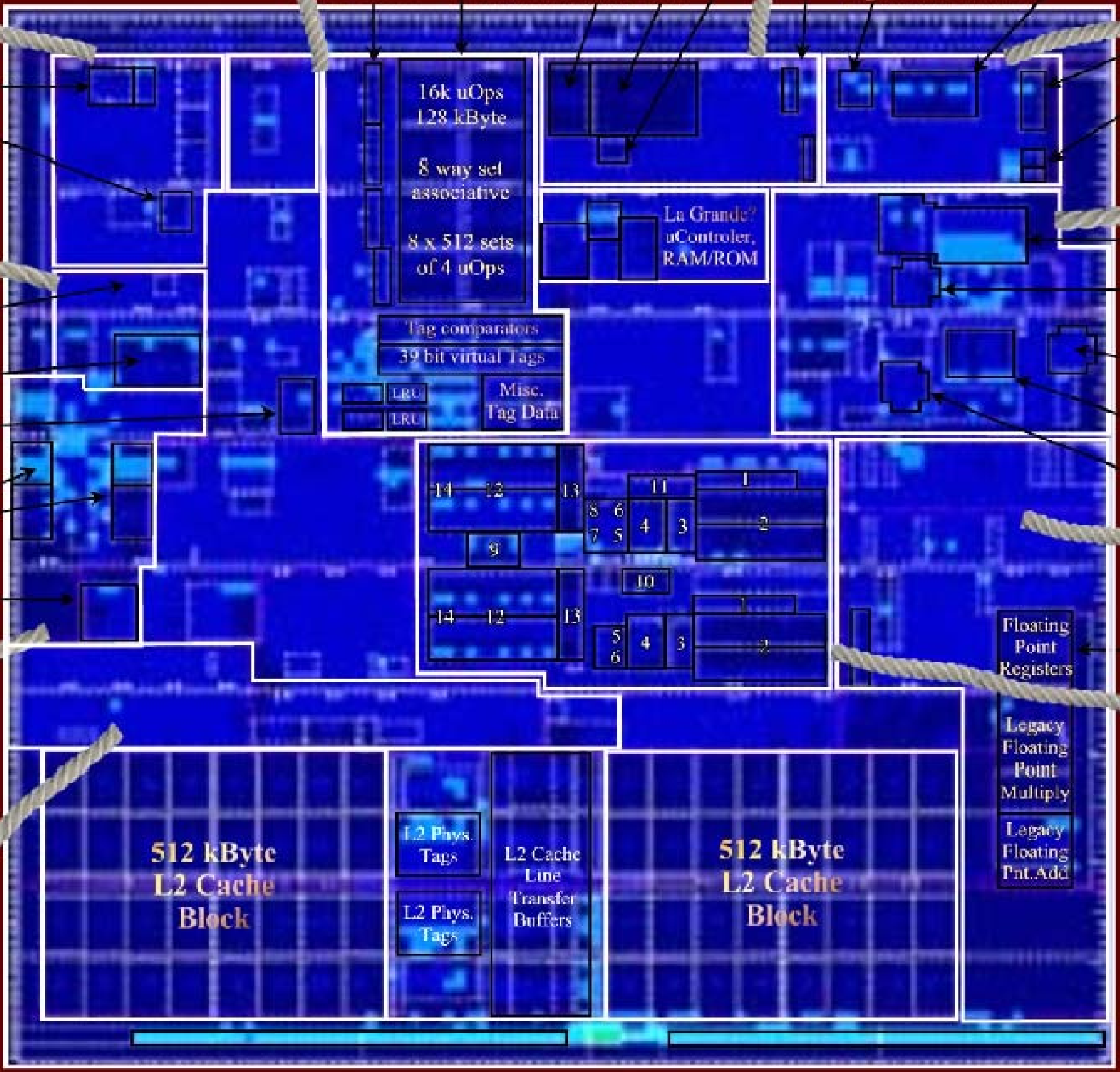
Personal Computers

Prescott Pentium 4

122 mm^2 with 125 million transistors



Intel Pentium 4 Prescott



Trace Cache Access, next Address Predict

Trace Cache Branch Prediction Table (BTB), 1024 entries.

Return Stacks (4 x 16 entries)

Trace Cache next IP's (4x)

Instruction Decoder

Up to 4 decoded uOps/cycle out (from max. one x86 instr/cycle)
Instructions with more than four are handled by Micro Sequencer

Raw Instruction Bytes in

Data TLB, 64 entry fully associative, between threads dual ported (for loads and stores)

Front End Branch Prediction Tables (BTB), shared, 4096 entries in total

Instruction TLB's 128 entry, fully associative for 4k and 4M pages. In: Virtual address [47:12] Out: Physical address [39:12] + 2 page level bits

Instruction Fetch from L2 cache and Branch Prediction

Front Side Bus Interface, 533..800 MHz

Instruction Trace Cache

Fill Buffers

Trace Cache

Micro code Flash & ROM

Micro code Sequencer

Execution Pipeline Start

uOp Queue

Register Alias History Tables (4x128)

Register Alias Tables

Buffer Allocation & Register Rename

Instruction Queue (for less critical fields of the uOps)

General Instruction Address Queue & Memory Instruction Address Queue (queues register entries and latency fields of the uOps for scheduling)

uOp Schedulers

Parallel (Matrix) Scheduler for the two double pumped ALU's

General Floating Point and Slow Integer Scheduler: (8x8 dependency matrix)

FP Move Scheduler: (8x8 dependency matrix)

Load / Store Linear Address Collision History Table

Load / Store uOp Scheduler: (8x8 dependency matrix)

FP, MMX, SSE1..3

Floating Point, MMX, SSE1..3 Renamed Register File 256 entries of 128 bit.

Integer Execution Core

(1) uOp Dispatch unit & Replay Buffer
Dispatches up to 6 uOps / cycle

(2) Integer Renamed Register File
256 entries of 32 bit (+ 6 status flags)
12 read ports and six write ports

(3) Databus switch & Bypasses to and from the Integer Register File.

(4) Flags, Write Back

(5) Double Pumped ALU 0

(6) Double Pumped ALU 1

(7) Load Address Generator Unit

(8) Store Address Generator Unit

(9) Load Buffer (96 entries)

(10) Store Buffer (48 entries)

(11) ROB Reorder Buffer 4x64 entries

(12) 16 kByte Level 1 Data cache
four way set associative. 1R/1W

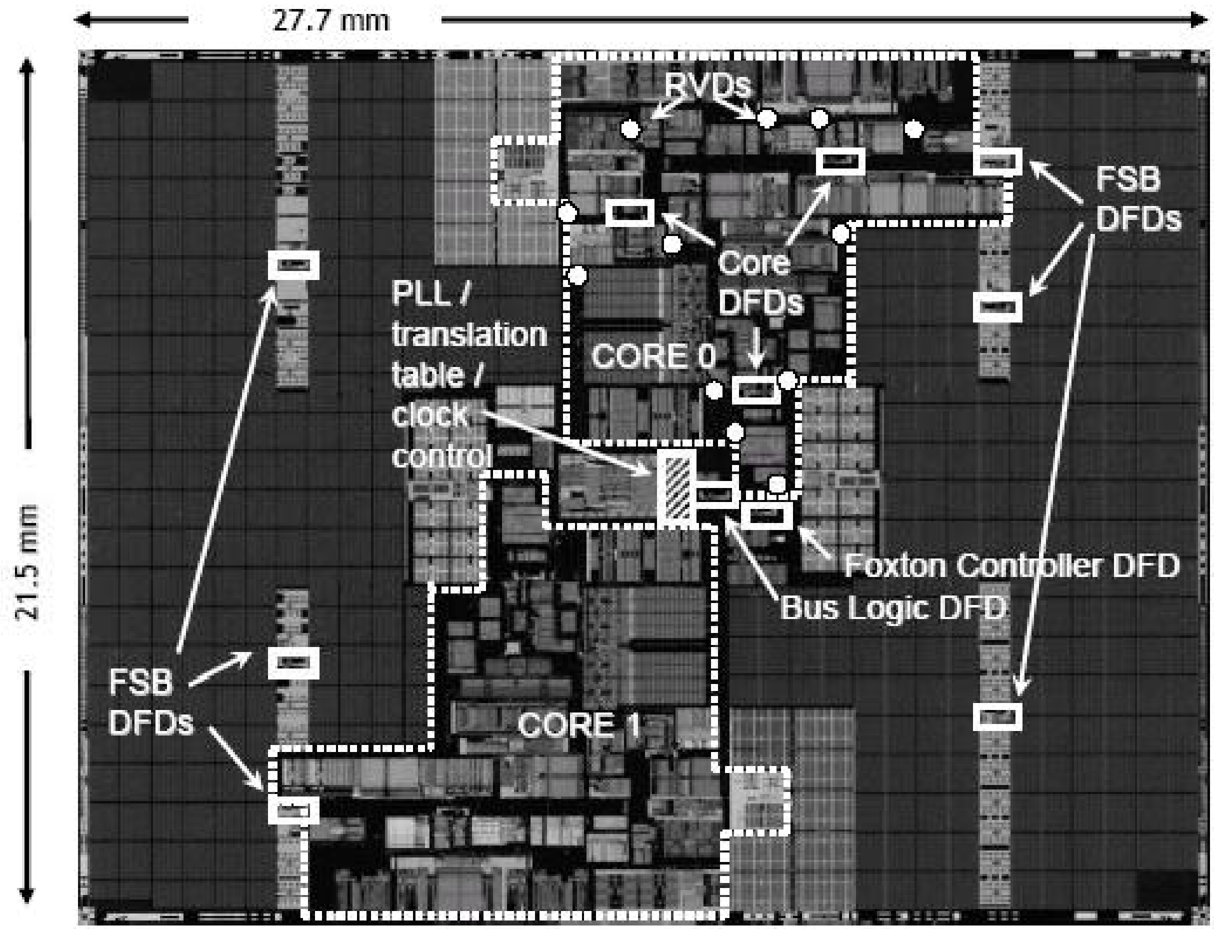
(13) Databus multiplexing

(14) Cache Line Read / Write Transferbuffers and
256 bit wide bus to and from L2 cache

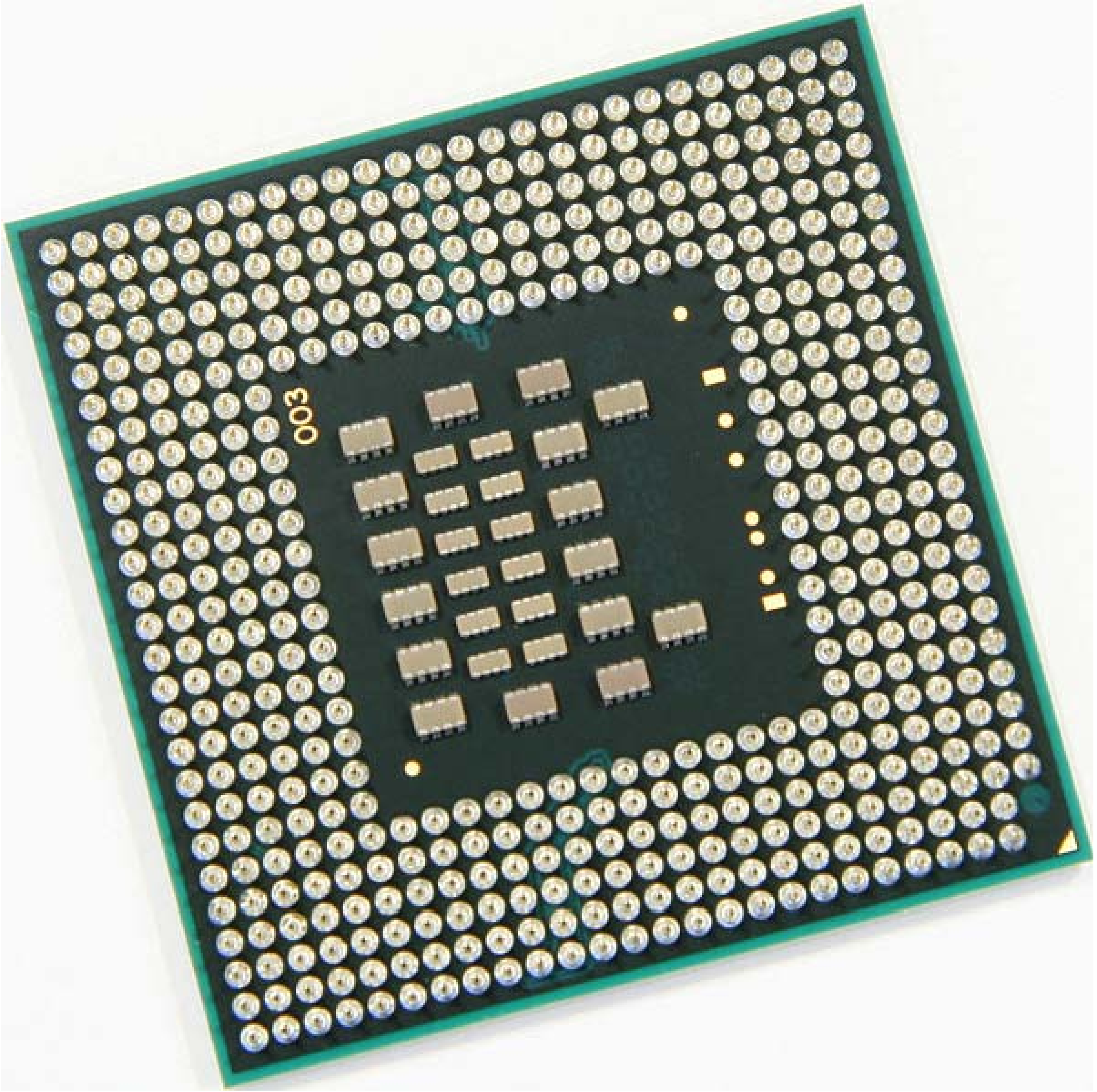
April 19, 2003 www.chip-architect.com

Dual-Core Intel Itanium 2 processor (Montecito)

1.72 billion transistors, 27.72 mm x 21.5 mm



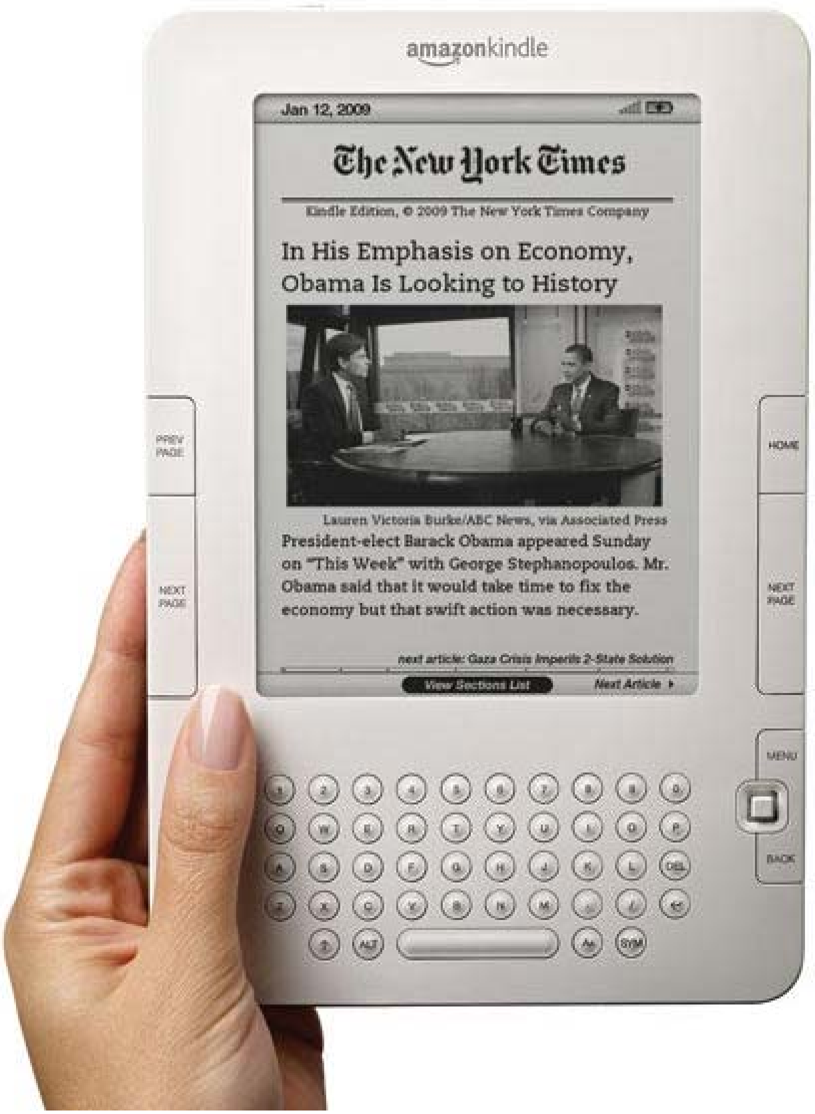
Core Duo



But what about …

Microsoft Kinect vs the ipad &iphone

We don’t even read the same ...



Jan 12, 2009

The New York Times

Kindle Edition, © 2009 The New York Times Company

In His Emphasis on Economy, Obama Is Looking to History

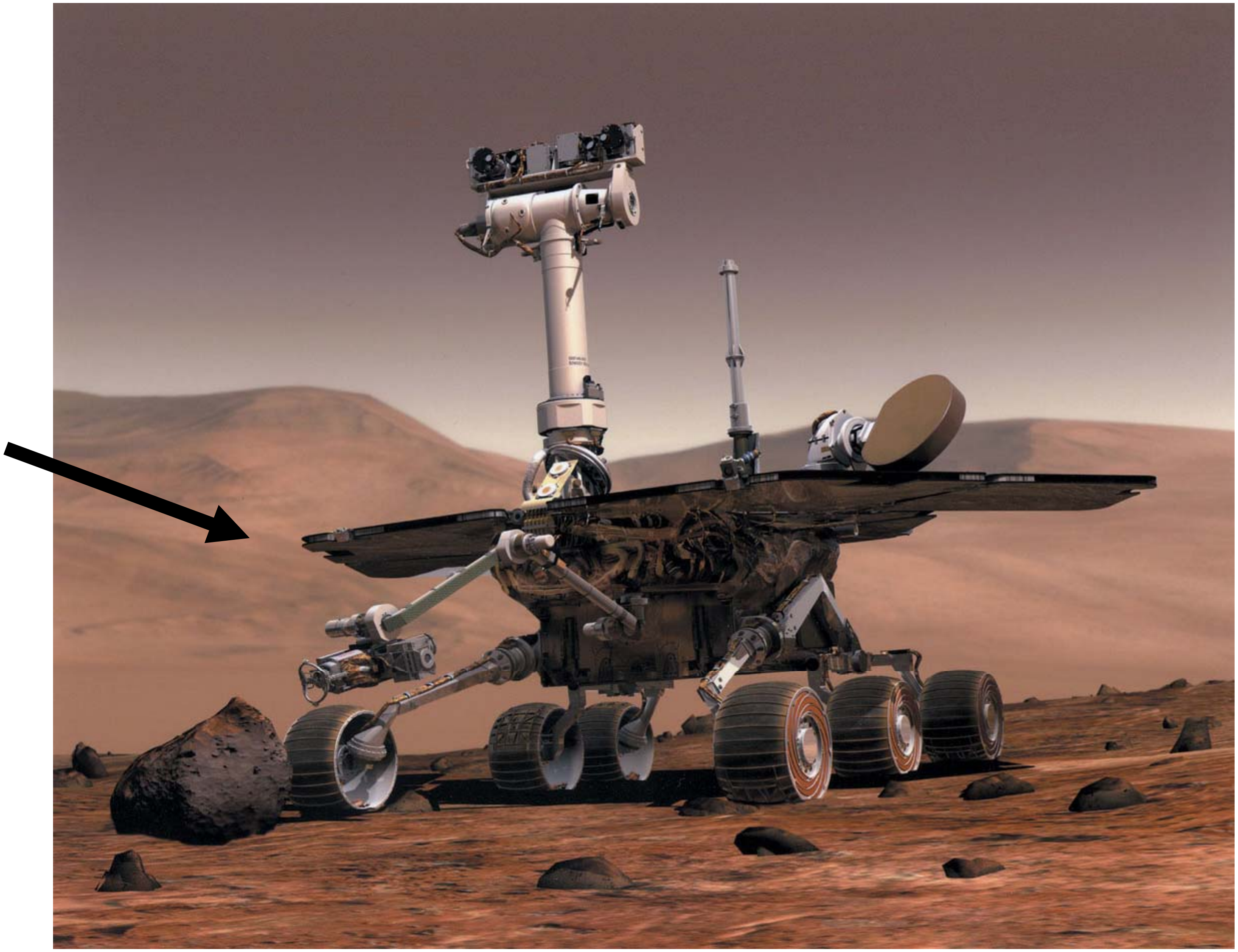
Lauren Victoria Burke/ABC News, via Associated Press

President-elect Barack Obama appeared Sunday on “This Week” with George Stephanopoulos. Mr. Obama said that it would take time to fix the economy but that swift action was necessary.

next article: Gaza Crisis Imperils 2-State Solution

Outer-Space

Mars Rover



Xilinx

FPGAs

Here

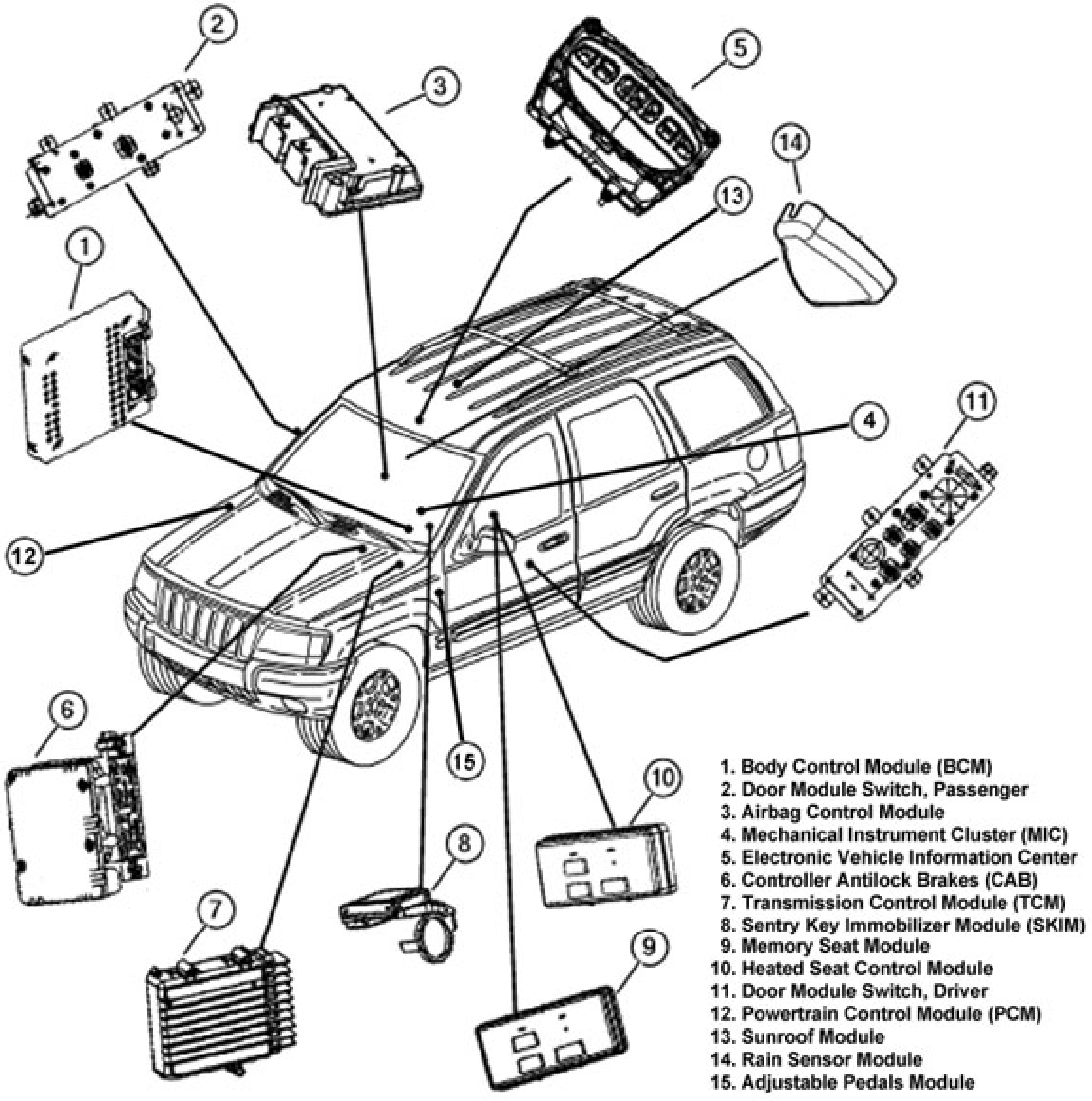
Orbital Space Plane

Orbital Space Plane

Cars & Boats

Boats

Automotive Electronics



1. Body Control Module (BCM)
2. Door Module Switch, Passenger
3. Airbag Control Module
4. Mechanical Instrument Cluster (MIC)
5. Electronic Vehicle Information Center
6. Controller Antilock Brakes (CAB)
7. Transmission Control Module (TCM)
8. Sentry Key Immobilizer Module (SKIM)
9. Memory Seat Module
10. Heated Seat Control Module
11. Door Module Switch, Driver
12. Powertrain Control Module (PCM)
13. Sunroof Module
14. Rain Sensor Module
15. Adjustable Pedals Module

7-Series BMW:
63 Embedded Processors

Mercedes S-Class
65 Embedded Processors

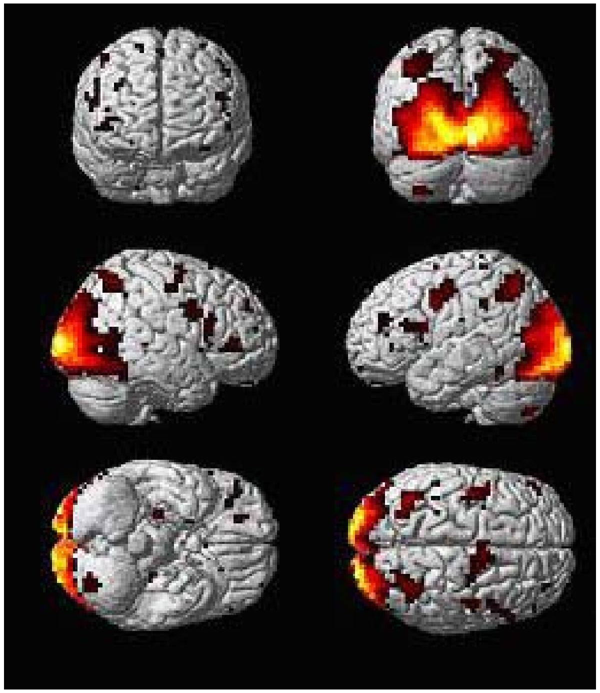
More than 80% of the innovation in autos is from innovations in electronics
- Daimler-Chrysler

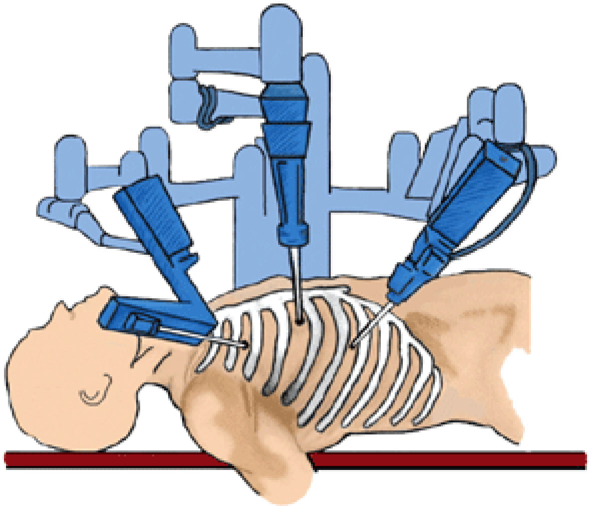
Automotive Semiconductor Market: US$13.1 billion / year

Inside you

Biomedical Applications

Medical Applications





Diagnostic Applications: MRI, PET, FD-OCT…

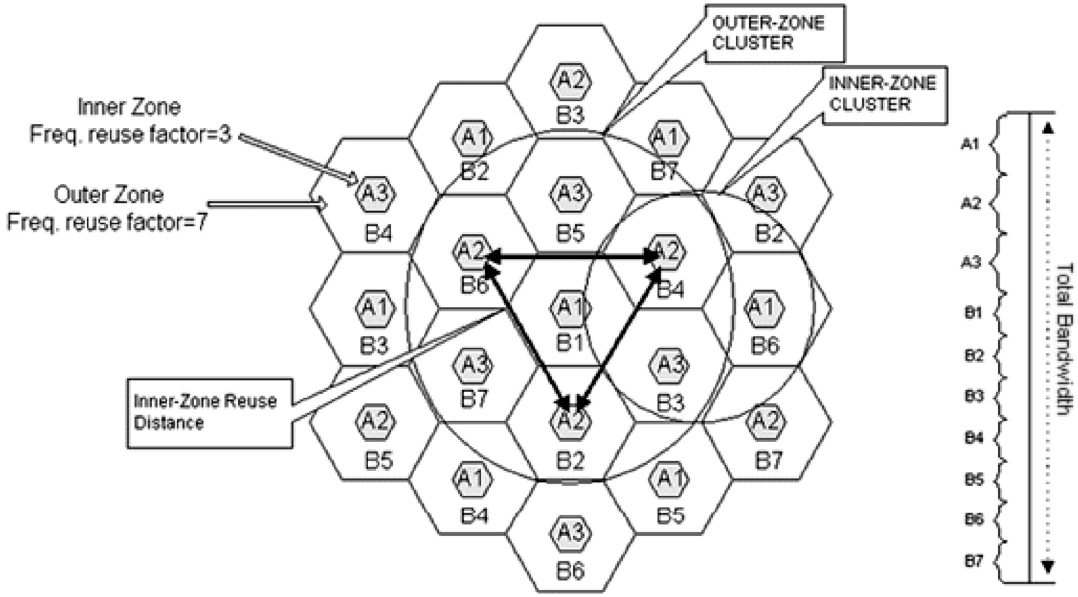
Tele-health (diagnostics and surgery)

Cellular Applications

Cellular Applications

Cellphones

&

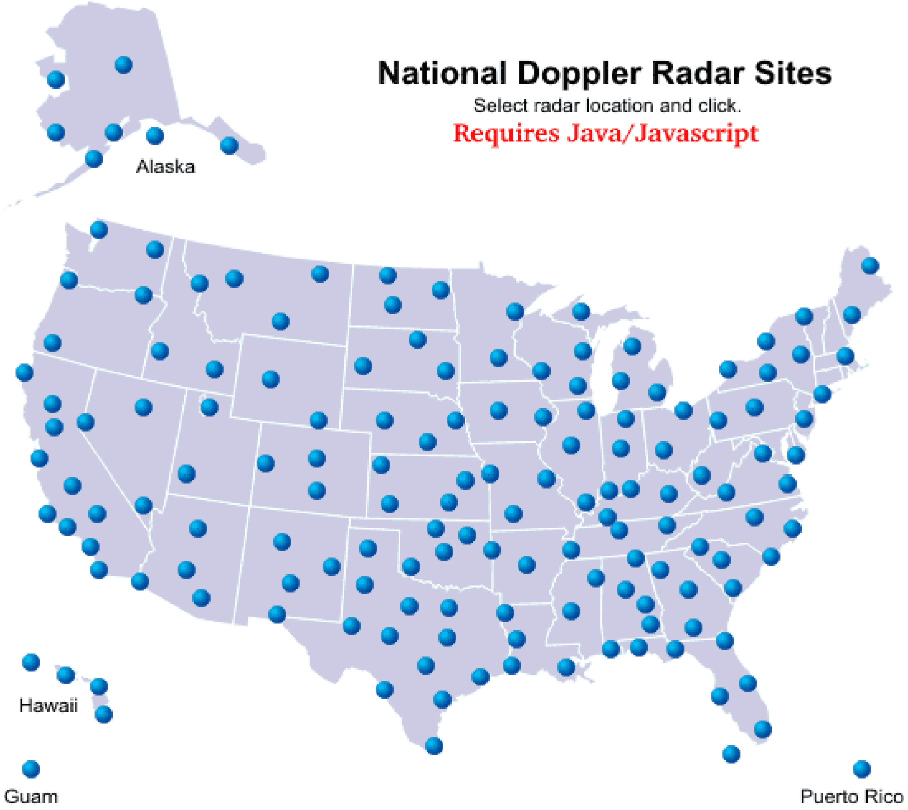


Cellphone Networks

Predicting the weather

Meteorological Applications

Doppler Weather Radar



National Doppler Radar Sites

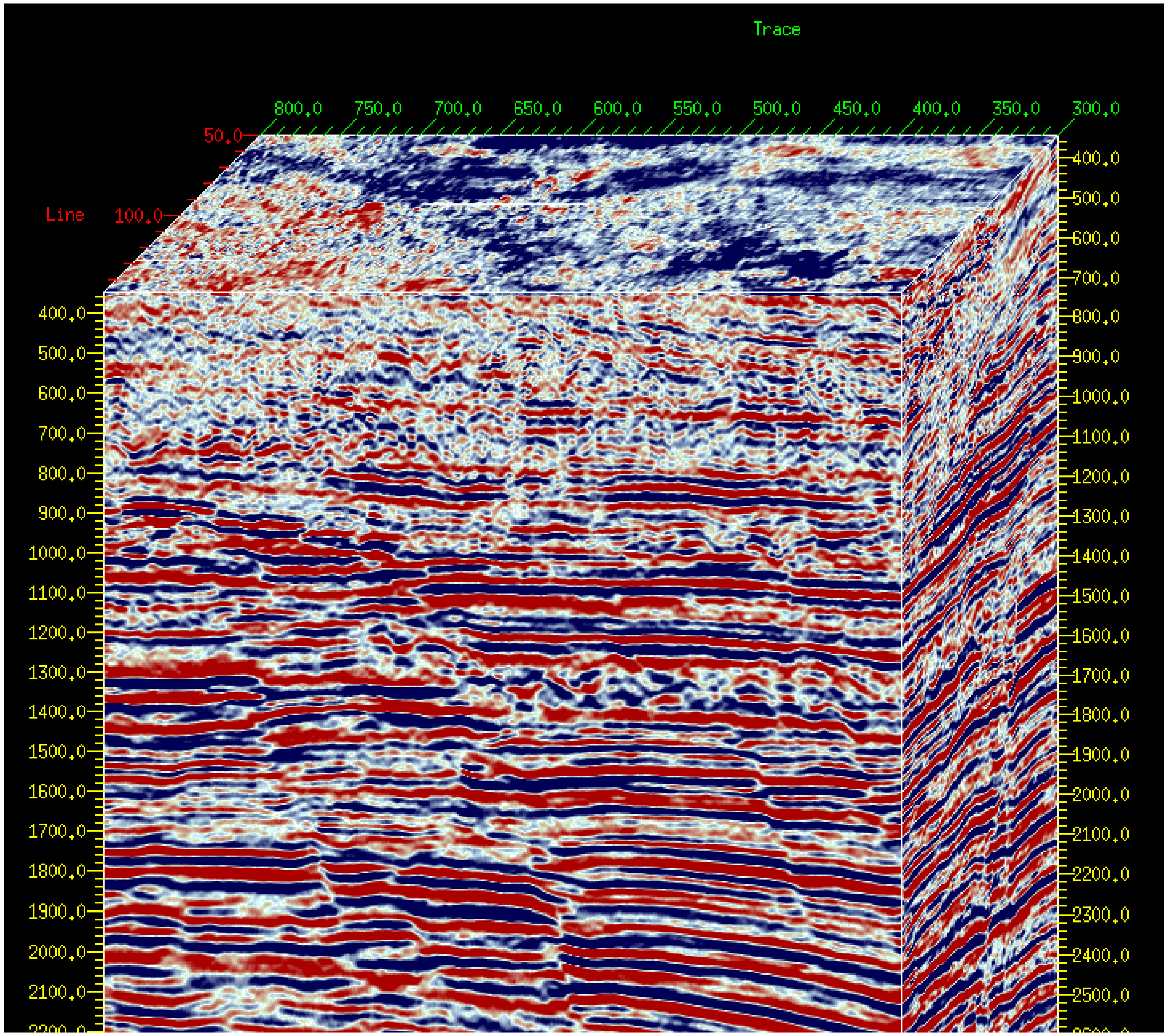
Select radar location and click.

Requires Java/Javascript

Finding Oil

Seismic Imaging Applications

Let's play: Find the petroleum!



Making movies

Special Effects and Computer Animated Films

Avatar: Petabytes worth of data

Lots of movie special effects done here in Vancouver

PLUS:

Music Mixing/Sound Effects

Environmental Monitoring

Production/Manufacturing Plants

Power Grid

Telecommunications Grid

Banking and Stock Markets

Hadron Collider

Quick Review: what should you know already?

- Von Neumann Architecture
- A software programming language (C/C++/java)
- Basic digital logic components and Synchronous Logic
  - Logic Gates and Registers
  - Counters, state machines
- Pipelining
  - Not for processors per se, but to make logic faster

Quick Review (cont'd)

- Numbers and Arithmetic
  - Base 2 and 16, floating point/fixed point, masking, etc
- I/O
  - Accessing via memory mapping, interrupts, buses, point-to-point, DMA
- Memory System
  - Distributed vs Shared

Quick Review (cont'd)

- Parallel Programming Concepts
  - Mutexes, semaphores, locks, barriers, message passing
- Assembly Language (what is it)?
- Context Switching
- Transistors (what the word means)

Quick Review (cont'd)

- CAD Flow or Compilation Flow
  - Basic understanding
- Operating Systems Basics

Why take this course?

- It’s cool! ☺

- Gives you a great overview of how computing systems work from top to bottom (good introductory course)

- Gives you a practical understanding of how the different layers interact and impact performance

- Highlights the kind of considerations that need to be evaluated for Application-Specific Designs (HUGE MARKET)

*Note: Two in-class tests.

What will you learn?

- What’s the cool research in computing system design!?!
- Basic interactions between hierarchy layers
  - How they impact your design and performance
- Reading and analysing research papers:
  - Critical Analysis skills

About Me

- Microelectronics Group
- Computer Engineer:
  - I teach Digital and Embedded Systems Design courses
  - My research group is the Reconfigurable Computing Lab
- My interests include:
  - Reconfigurable Computing, Application-Specific Architectures, Design Methodologies, FPGAs, Embedded Systems, Computing System Design, CAD, Operating Systems

Logistics

My Contact Info

- Office:
  - ASB8819
- Email:
  - lshannon@ensc.sfu.ca
- Homepage:
  - http://www.ensc.sfu.ca/~lshannon/
- Office Hours:
  - Tuesdays 10:30-12:30pm & Wednesdays 2:30-3:30pm

Lectures

- Wednesdays 5:30-8:30pm
  - 5:30-7pm; 15 minute break; 7:15-:815pm

- May schedule a Guest Lecture
  - ***If so ATTENDANCE IS MANDATORY***

- Guest lecture at UBC recommended:
  - Dr. Philip Leong (more to follow; check the bulletin board)

Course Webpage

- Course Webpage:
  - http://www.ensc.sfu.ca/~lshannon/courses/ensc894/
- Course Handouts will be posted here.
- Lecture Slides will also be posted before each class.
- We will be using WebCT for bulletin board postings only
- A Course Mailing List has been created
  - It will only be used to post last minute notices.
  - All other information will be posted to WebCT

Course Bulletin Board

Please observe appropriate bulletin board etiquette (***be respectful***).

Course Bulletin Board Disclaimer

Please do not use these forums to post any material that is knowingly false and/or defamatory, abusive, vulgar, hateful, obscene, threatening, invasive of a person's privacy, or otherwise in violation of any law. Do not post any copyrighted material unless the copyright is owned by you. I reserve the right to remove any messages posted and to reveal your identity (or whatever information is known about you) in the event of a complaint or legal action arising from any message posted by you.

By posting your message, you agree to indemnify me, my employees, agents and representatives, and to hold them harmless from any and all claims and liabilities (including lawyers' fees) resulting from any material posted to these forums, or from any acts resulting from participants' use of these forums .

Class Surveys

- This is the FIRST offering of the course:
  - I'm developing/revising the lecture material
  - For this, I really would appreciate your honest input
  - I'm interested to find out what you think of the
    - Course Material
    - Pace and Depth of Presentation
    - Selected Readings
    - Approach
  - I'll be giving a survey near the end of the course
    - However, if people have concerns/suggestions during the term, please let me know

Grade Breakdown

- 10 % Reading Group Participation
- 15 % 3 Paper Critiques*
- 25 % Seminar Presentation
- 25 % Research Review*
- 25 % Application Analysis*

*Although there is no official course project, at the instructor's discretion, students can choose to do a project in lieu of one more components of the course work (see handout)

Handouts

Remember: Pictures say a thousand words!

Lecture Topics

Lecture Topics

- January 26th: Programming Models – how we picture the problem.
  - Parallel Models: Dataflow, Queuing Models, Streams; Shared vs Distributed
- February 2nd: Programming Languages
  - Language Choice: VHDL, CUDA, OpenCL, LabView, Java, …
  - Algorithms and Data Structures
- February 9th: Analysing Application Performance, Debugging and Testing
  - Hardware Performance Counters, Gprof, vtune, Simulators, Testbeds, Code coverage, software engineering (program Design) …

Lecture Topics

- February 16th: Synthesis and Compilation
  - Just in time, Dynamic Recompiling
  - Optimization choices (area, power, speed)
- February 23rd: Operating Systems
  - Key functionalities; Customization (Keeping what you need)
  - Scheduling
- March 2nd: CLASS CANCELLED

Lecture Topics

- March 9th/16th: Computing Architectures (SoC vs the Cloud)**
  - Heterogeneous, NoC, Embedded Computing, reconfigurable computing
  - Using the cloud, managing data in the cloud
- March 16th/23rd: Computing System Design
  - Where's the bottleneck? Memory hierarchy and memory Technology
- March 23rd/30th: Reliability and DFT, Silicon Debug, Verification
  - Critical Systems

Lecture Topics

- March 30th/April 6th: Security
  - Is your application/data safe? (Drop boxes, …)
- April 6th/13th: Computing Technology
  - CMOS (how far can we go); molecular; quantum; ???

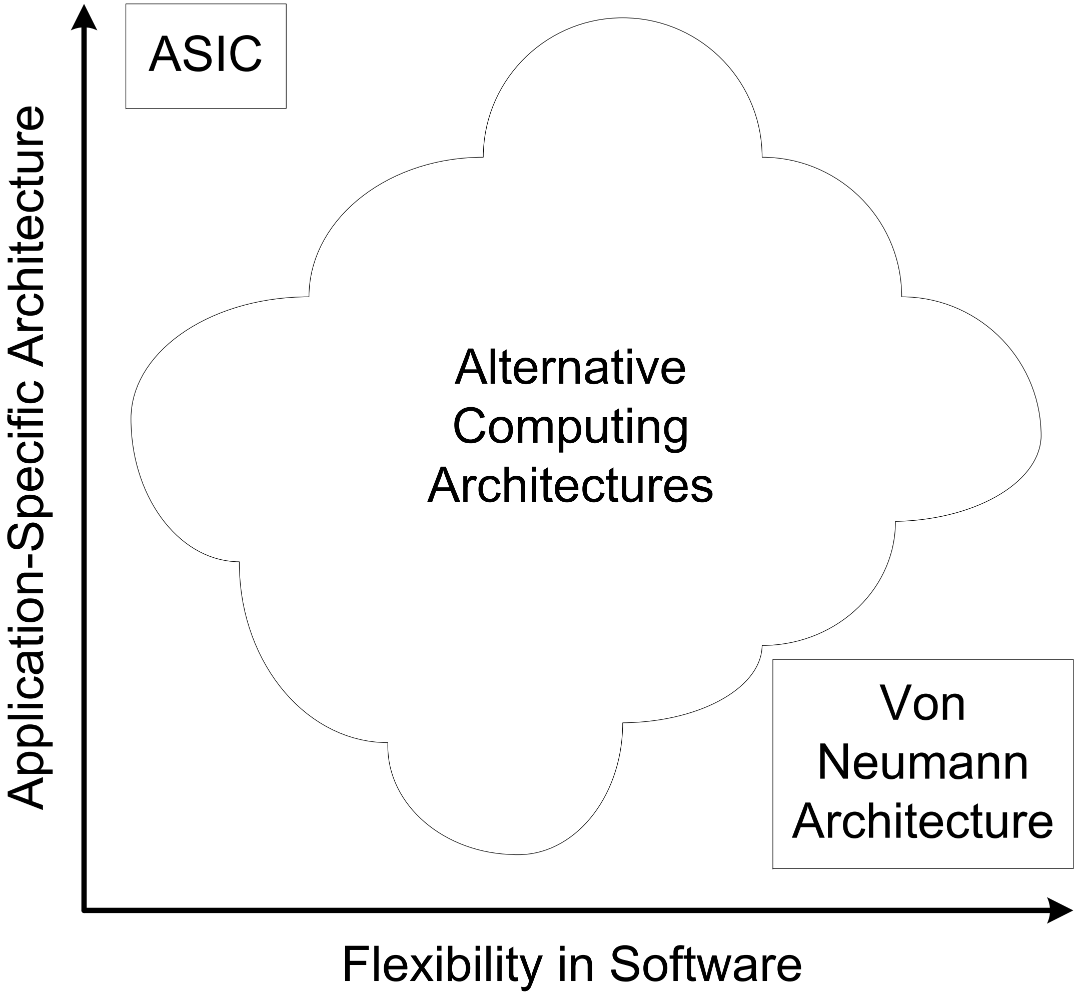
Lecture Topics: What does this look like?

Other Possible Lecture Topics

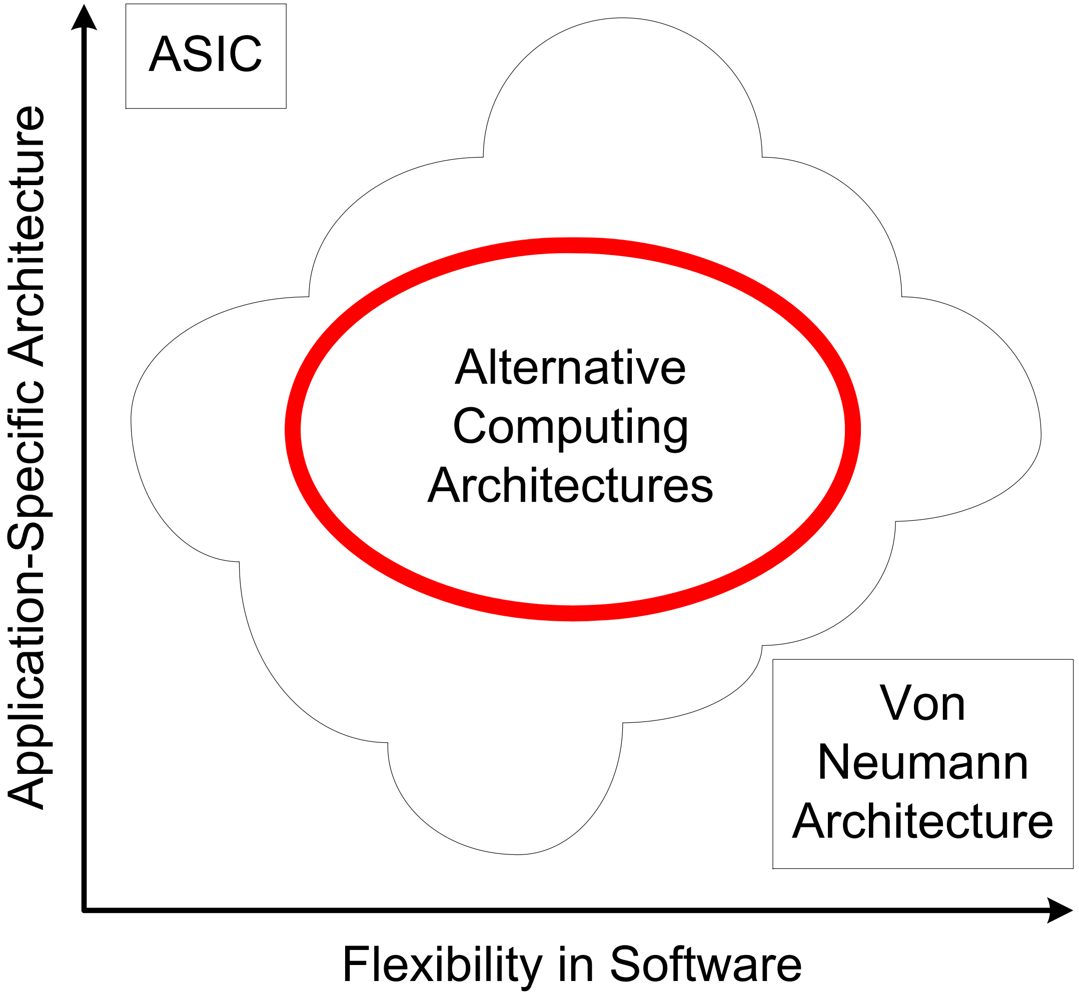
- Asynchronous Design
  - Data Synchronization Issues
  - Meta-stability

For Reference

Computing System Implementations


ASIC
Alternative Computing Architectures
Von Neumann Architecture
Application-Specific Architecture
Flexibility in Software

Computing System Implementations

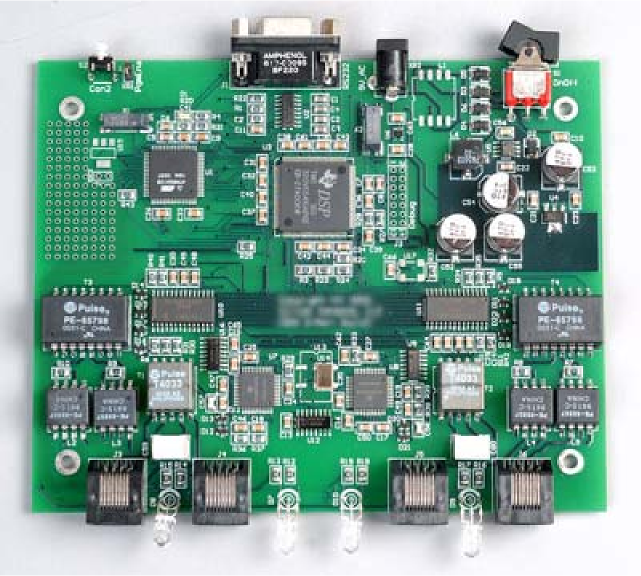

ASIC
Alternative Computing Architectures
Von Neumann Architecture
Application-Specific Architecture
Flexibility in Software

Alternative Computing Architectures

- Systems-on-Chip
- Many-core heterogenous Systems-on-Chip
- Networks-on-Chip
- Embedded and Real Time Computing Systems
- Reconfigurable Computing
- Meta-Computing/the cloud
- …. and more

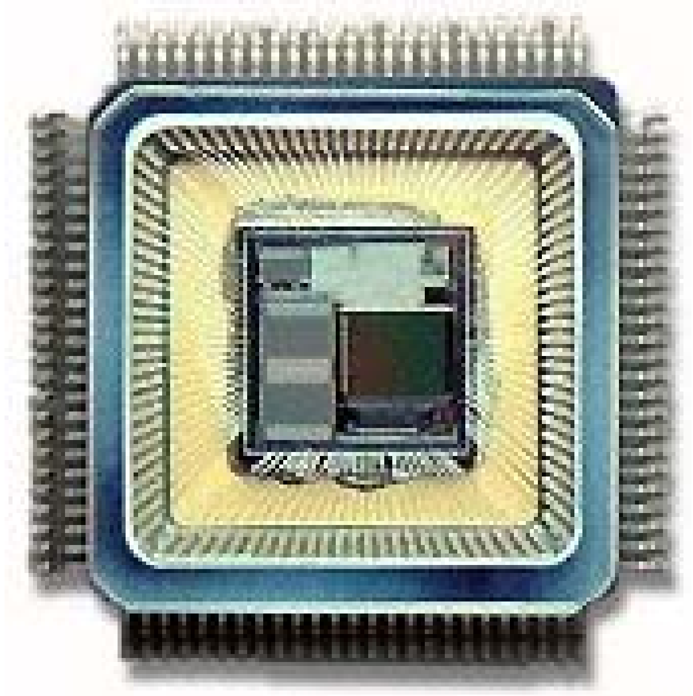
Recall: System-on-Chip

Old



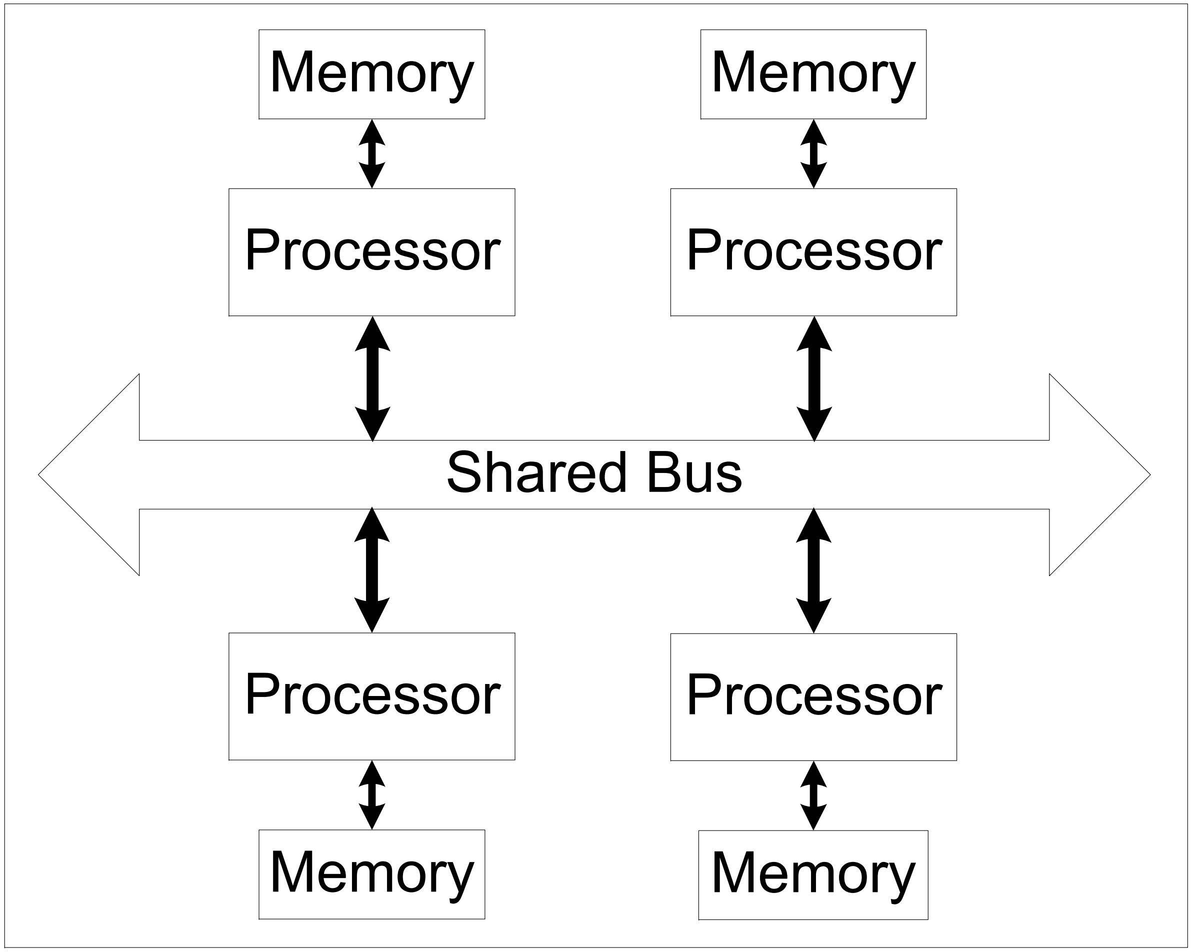
- System on Printed Circuit Board

New

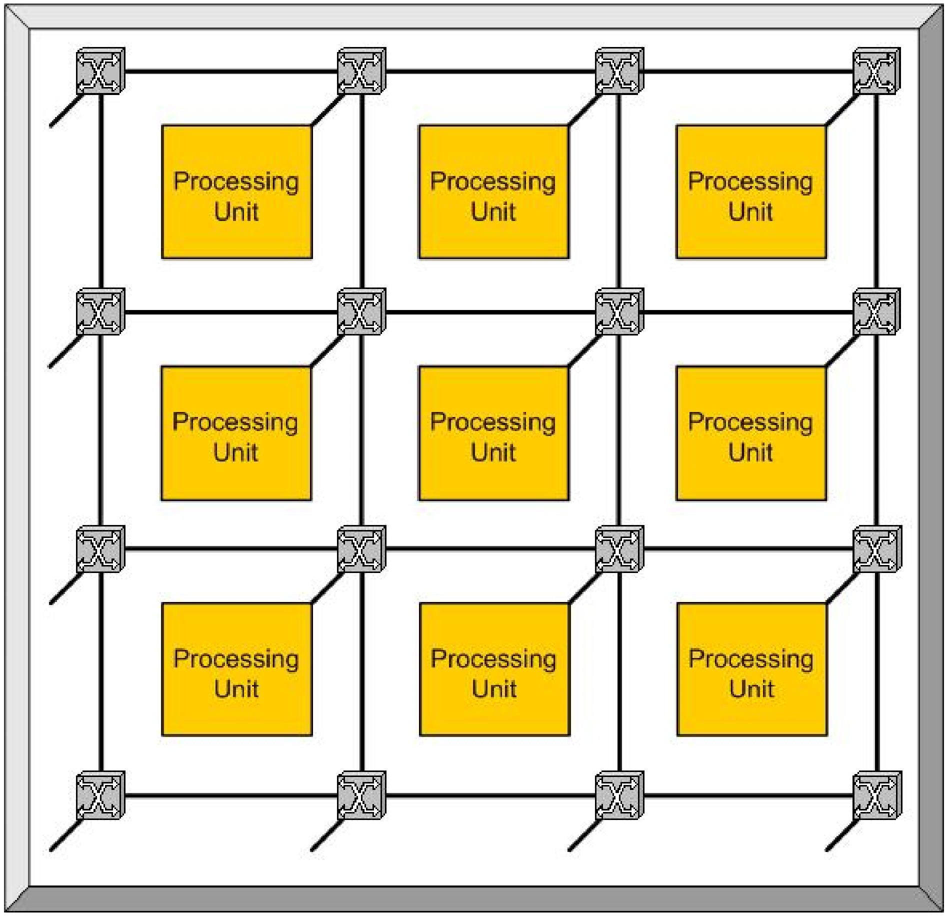


- System-on-Chip (SoC)

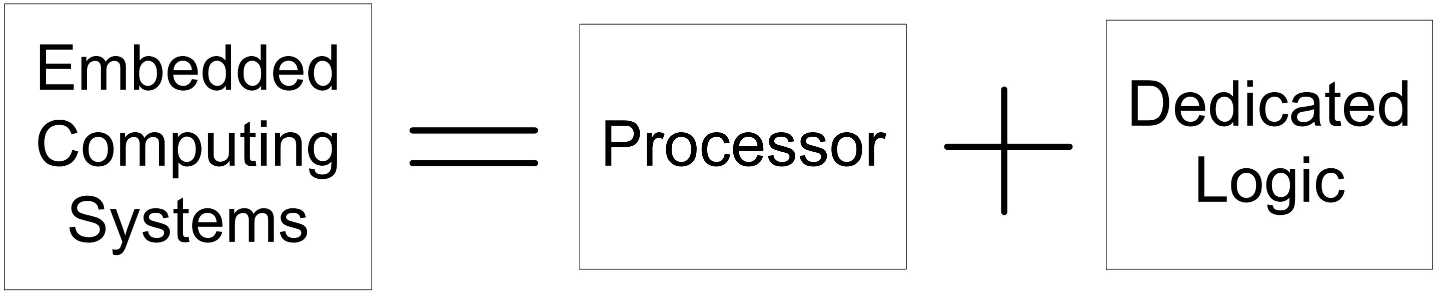
Multi-Processor System-on-Chip



Network-on-Chip



Embedded Computing Systems

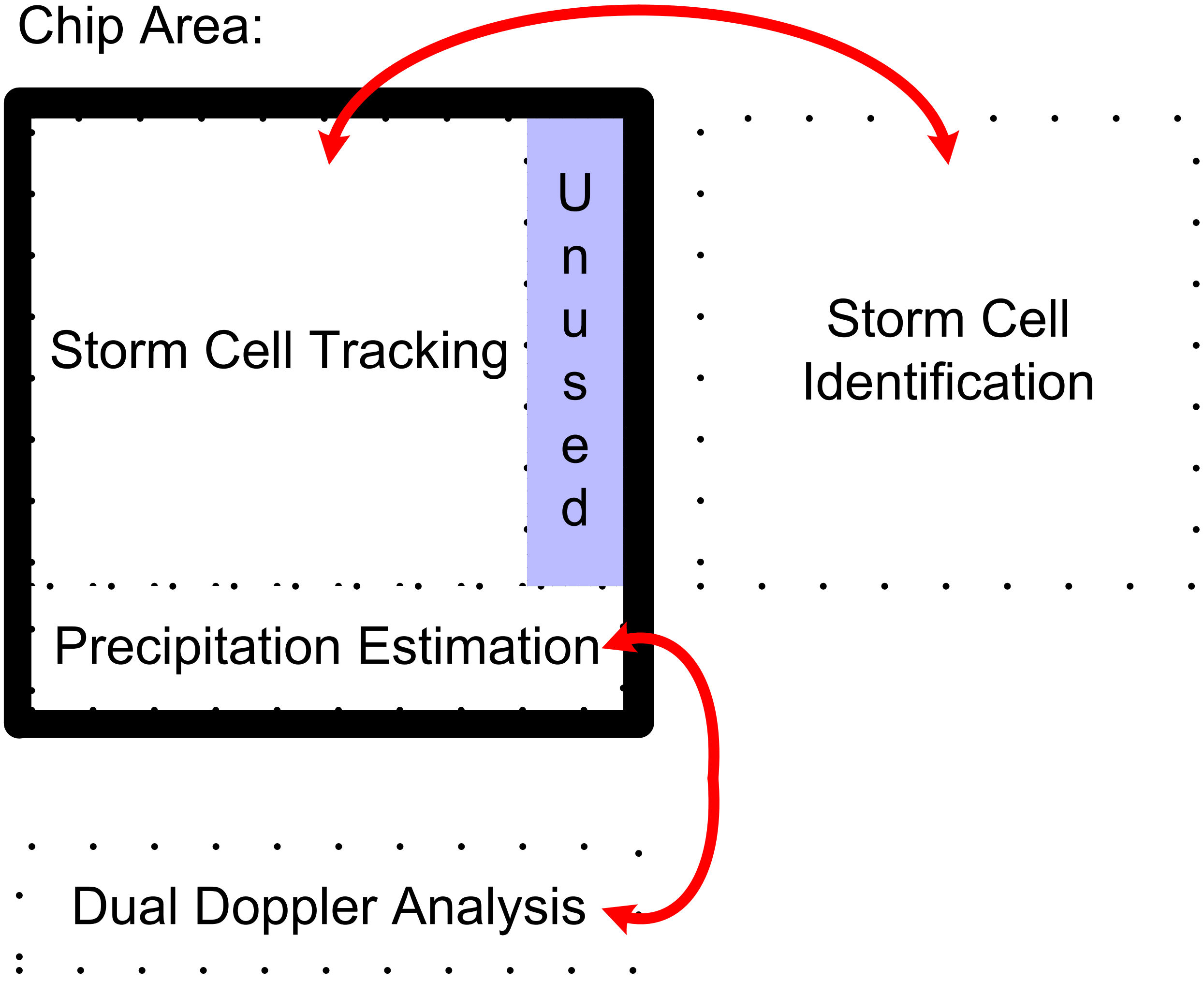


Embedded Computing Systems = Processor + Dedicated Logic

- Design Constraints:
  - Area
  - Power
  - Performance
  - Cost

Reconfigurable Computing

Chip Area:



Storm Cell Tracking

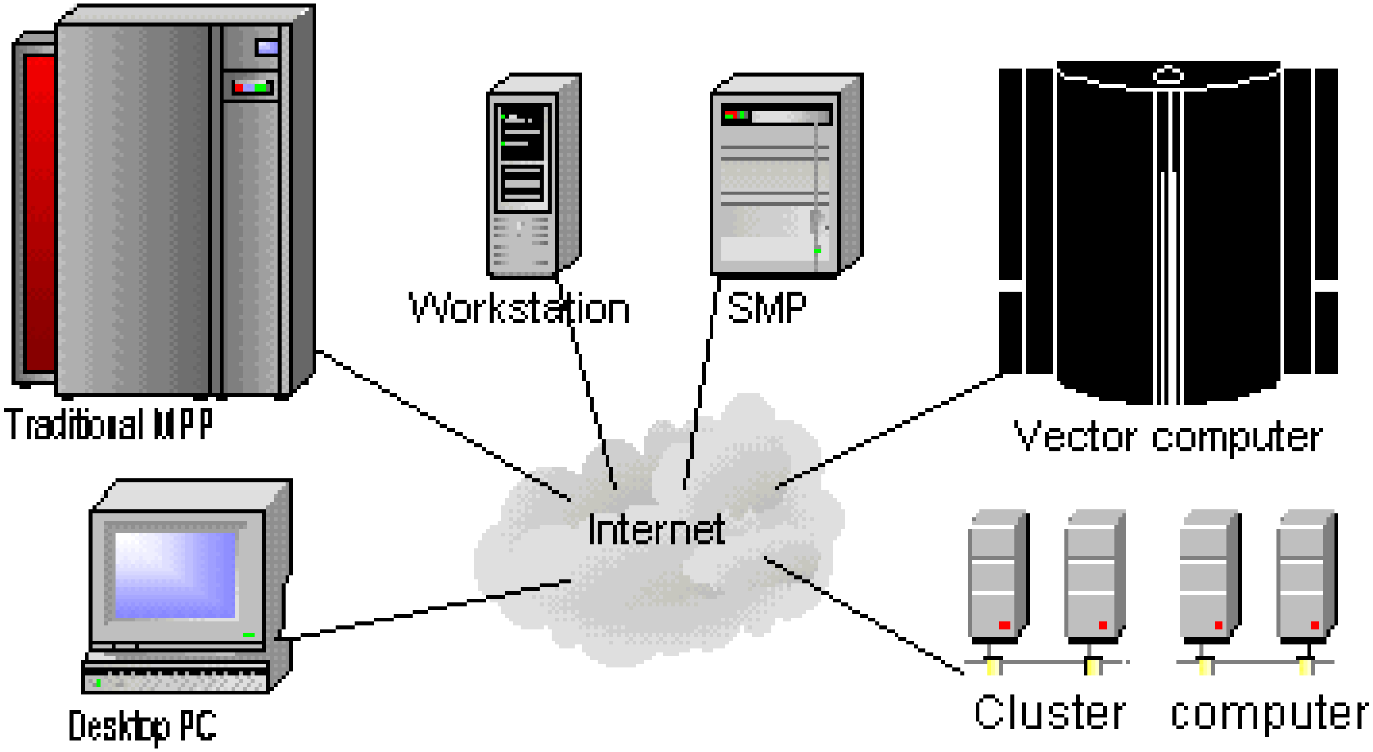
Unused

Storm Cell Identification

Precipitation Estimation

Dual Doppler Analysis

Meta-Computing/the Cloud


Traditional MPP
Workstation
SMP
Vector computer
Internet
Desktop PC
Cluster computer