CSCE 4114
Embedded Systems
Class website:

hthreads.github.io/classes/embedded-systems/

David Andrews

dandrews@uark.edu

Embedded Systems

1. What is an Embedded System ?

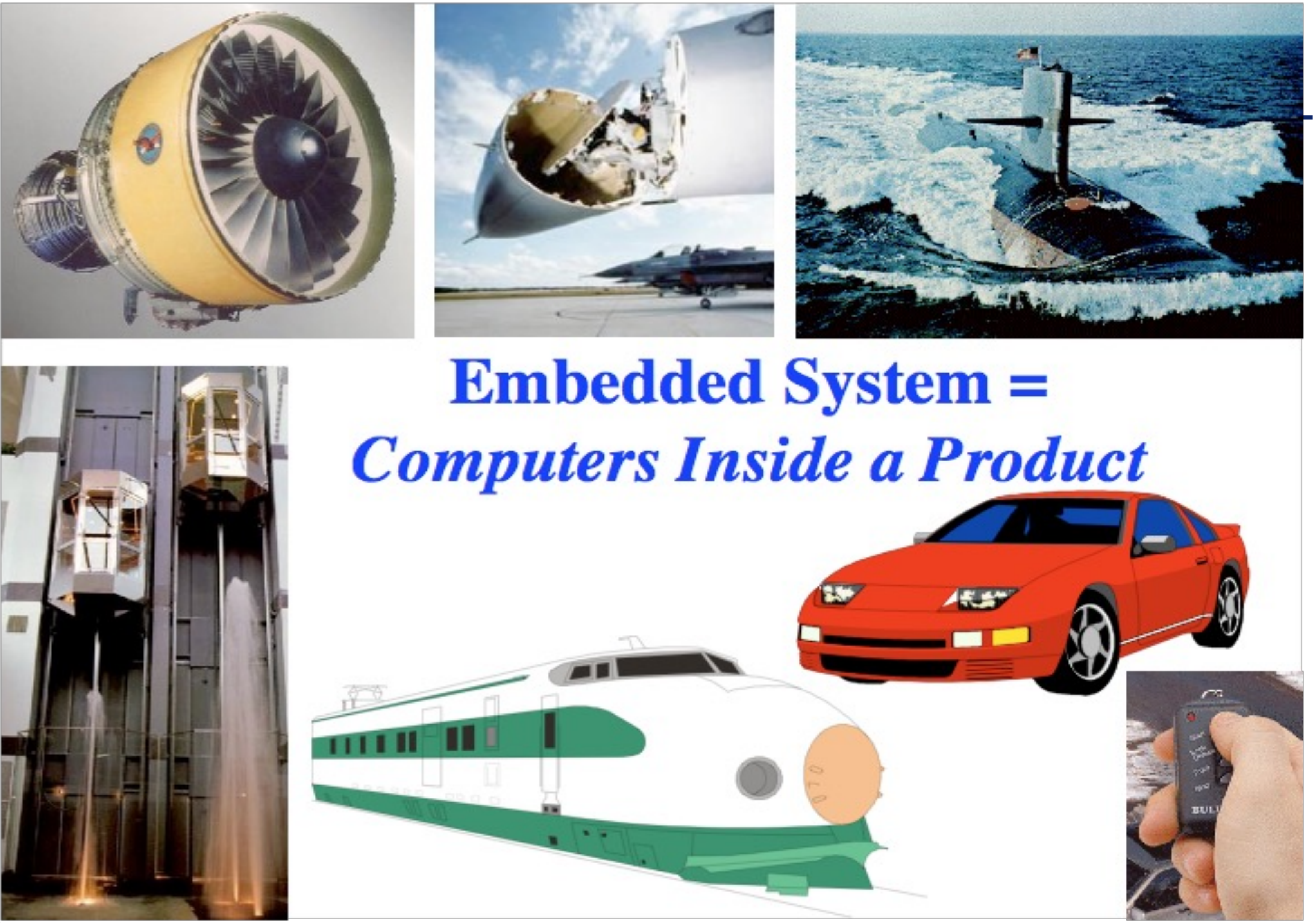
What is an Embedded System ?

- An **embedded system** is a computer system designed to do one or a few dedicated and/or specific functions[1][2] often with real-time computing constraints.

- It is *embedded* as part of a complete device often including hardware and mechanical parts.

- By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs.

Wikipedia

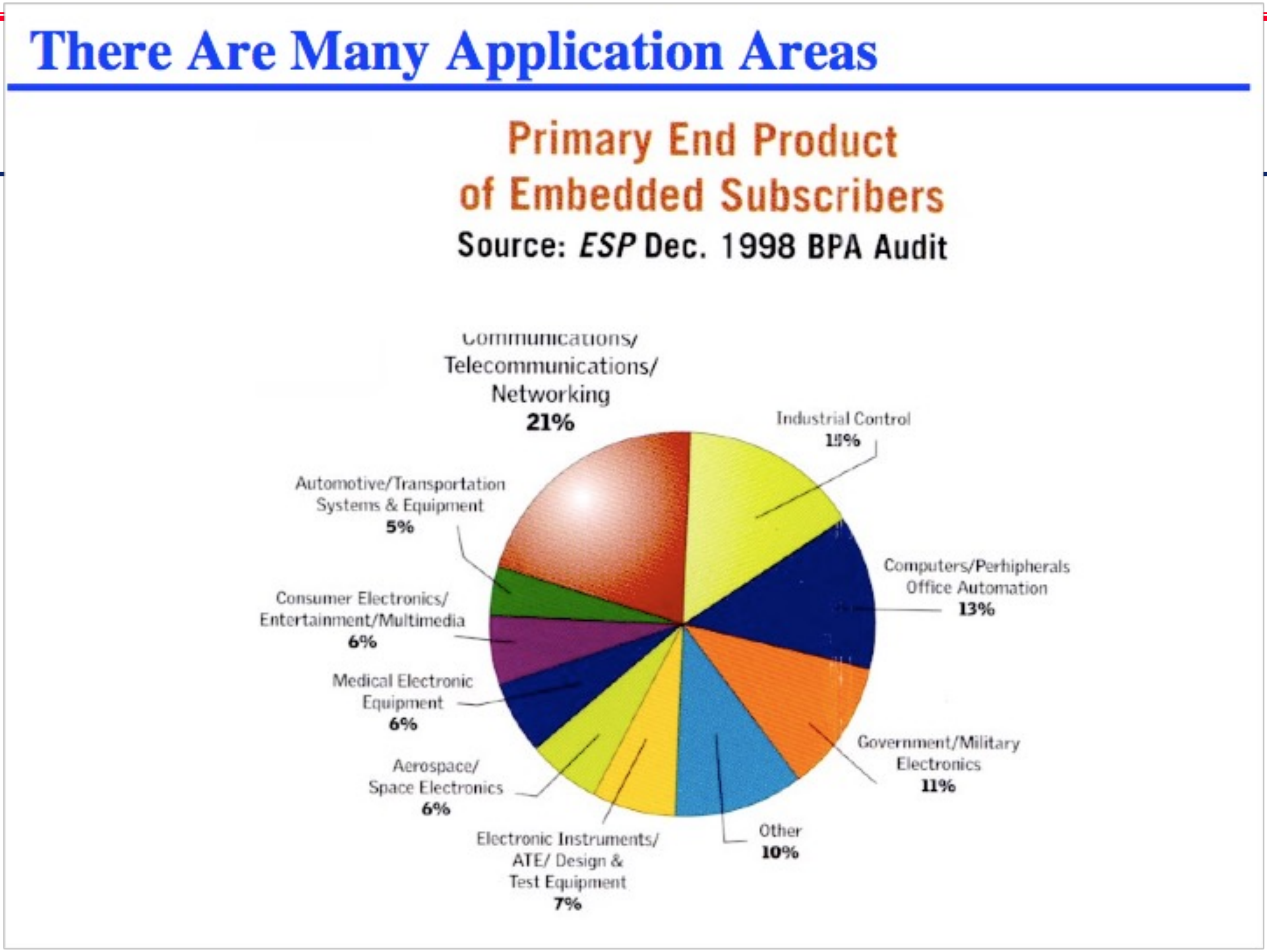


Embedded System =

Computers Inside a Product

From Koopman http://www.ece.cmu.edu/~ece649/lectures/01_intro.pdf

From Koopman http://www.ece.cmu.edu/~ece649/lectures/01_intro.pdf



There Are Many Application Areas

Primary End Product of Embedded Subscribers

Source: *ESP* Dec. 1998 BPA Audit

Which one will you be working in ?

From Koopman http://www.ece.cmu.edu/~ece649/lectures/01_intro.pdf

Embedded Versus Desktop

- Environmental Constraints

Embedded Versus Desktop

- Environmental Constraints
  - Size, Form Factor: must fit inside something else
    - Desktop can take up much more room

Embedded Versus Desktop

- Environmental Constraints
  - Size, Form Factor: must fit inside something else
    - Desktop can take up much more room
  - Weight: Must exist within a system that has weight limitations
    - Desktop isn't portable and weight not a factor

Embedded Versus Desktop

- Environmental Constraints
  - Size, Form Factor: must fit inside something else
    - Desktop can take up much more room
  - Weight: Must exist within a system that has weight limitations
    - Desktop isn't portable and weight not a factor
  - Energy: Usually battery operated
    - Desktop's plug into walls, Laptops recharged nightly

Embedded Versus Desktop

- Environmental Constraints
  - Size, Form Factor: must fit inside something else
    - Desktop can take up much more room
  - Weight: Must exist within a system that has weight limitations
    - Desktop isn't portable and weight not a factor
  - Energy: Usually battery operated
    - Desktop's plug into walls, Laptops recharged nightly
  - Harsh Environments: Vibration, Heat, Cold
    - Desktops operate in ambiant temperature range

Embedded Versus Desktop

- Environmental Constraints
  - Size, Form Factor: must fit inside something else
    - Desktop can take up much more room
  - Weight: Must exist within a system that has weight limitations
    - Desktop isn't portable and weight not a factor
  - Energy: Usually battery operated
    - Desktop's plug into walls, Laptops recharged nightly
  - Harsh Environments: Vibration, Heat, Cold
    - Desktops operate in ambiant temperature range
- Cost
  - Pennies on the Dollar. Must be Cheap
  - Desktop: You pay for the machine, not something else

Embedded Versus Desktop

- Functionality: Less Than General Purpose

Embedded Versus Desktop

- Functionality: Less Than General Purpose
  - Implements specific functions (Control loops)
  - Desktops Must Do Everything

Embedded Versus Desktop

- Functionality: Less Than General Purpose
  - Implements specific functions (Control loops)
  - Desktops Must Do Everything
- Performance

Embedded Versus Desktop

- Functionality: Less Than General Purpose
  - Implements specific functions (Control loops)
  - Desktops Must Do Everything
- Performance
  - Must meet timing constraints.
  - Desktop: A Little Delay is “OK”

Embedded Versus Desktop

- Functionality: Less Than General Purpose
  - Implements specific functions (Control loops)
  - Desktops Must Do Everything
- Performance
  - Must meet timing constraints.
  - Desktop: A Little Delay is "OK"
- Operational Mode

Embedded Versus Desktop

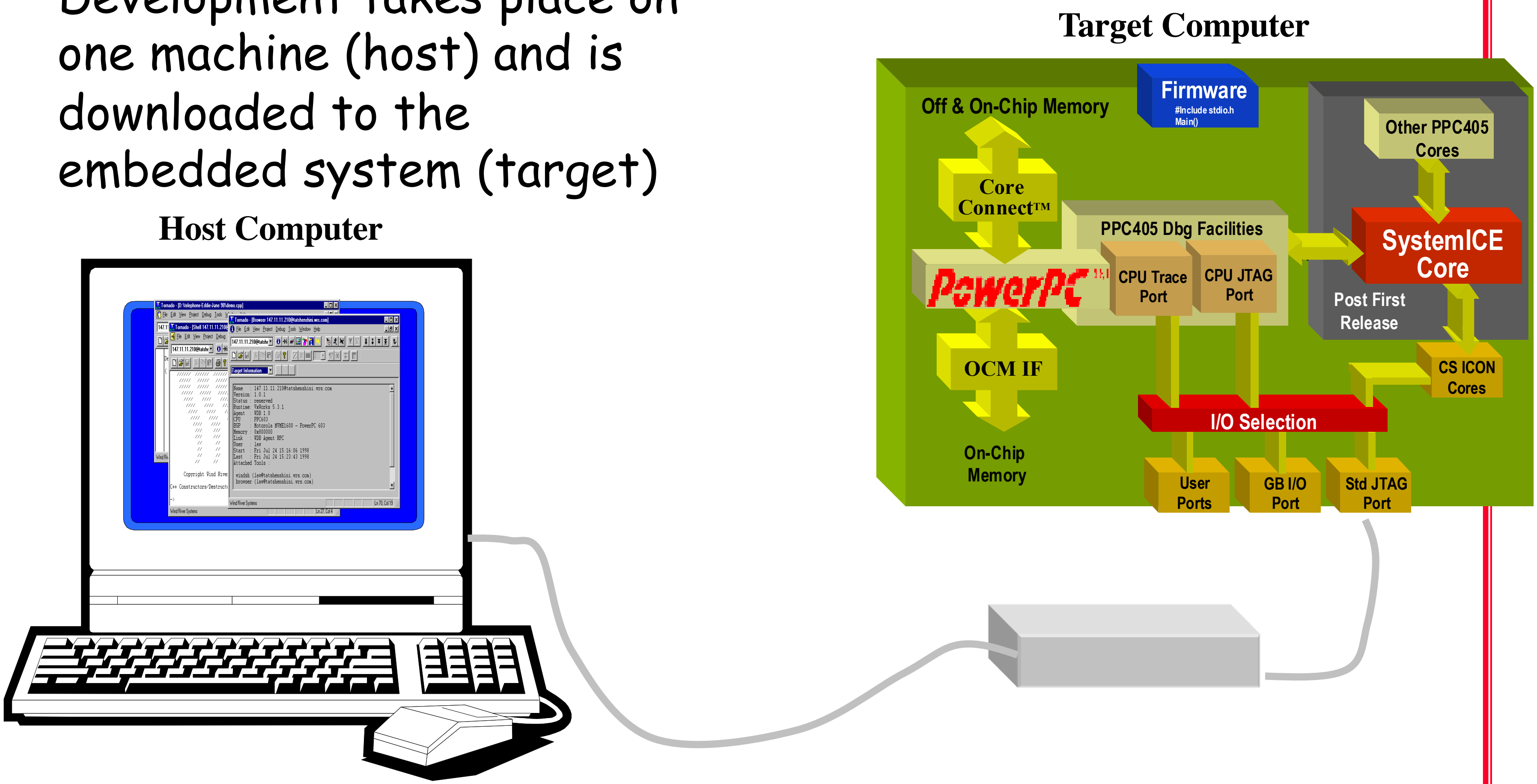
- Functionality: Less Than General Purpose
  - Implements specific functions (Control loops)
  - Desktops Must Do Everything
- Performance
  - Must meet timing constraints.
  - Desktop: A Little Delay is “OK”
- Operational Mode
  - Processing is “reactive” to stimulus (input)
  - Dekstop: Batch Processing

Embedded Versus Desktop

- Technology:
- Small Memory Capacity, No Hard Disk
  - Implications on size of Program & Data Storage
  - Desktop: Memory is cheap and abundant
- Processor Selection (somewhat historical)
  - Large use of older generation 8,16 bit processors
    - Cheaper, Smaller, Less Energy
  - Desktop: We want the latest and greatest !

Embedded versus Desktop Development Environments

- Development takes place on one machine (host) and is downloaded to the embedded system (target)



A cross-compiler is run on the host

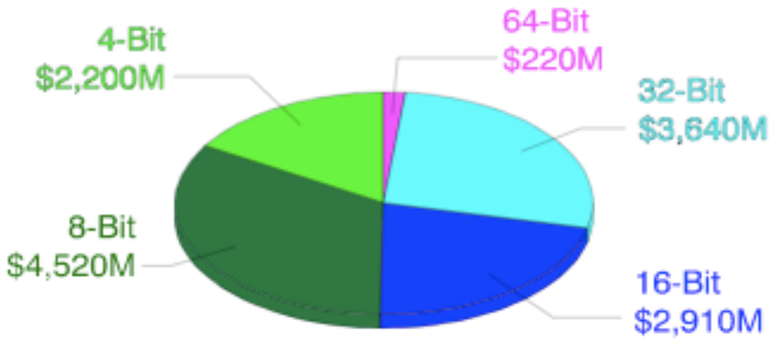
From Xilinx EDK Presentation

Is it Important ?

- >99% of CPU's sold are for Embedded Systems. Things like.....
- National Defense
- Power Grid
- Personal Electronics (TV's, phones etc)
- and importantly your set top box games…

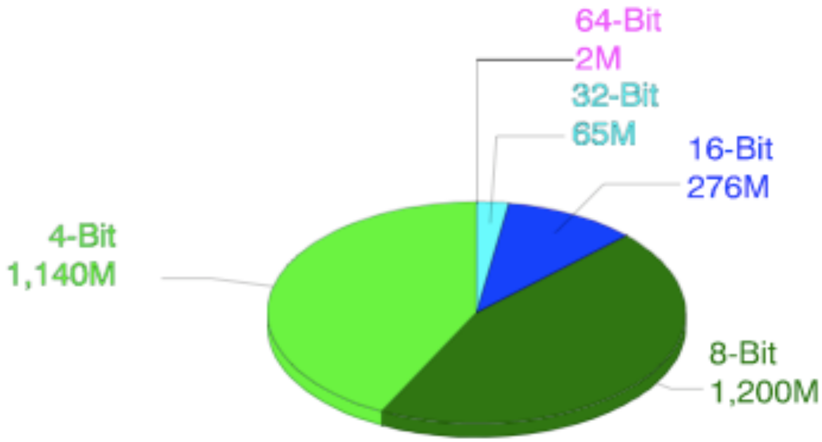
Small Computers Rule The Marketplace

- **~80 Million PCs vs. ~3 Billion Embedded CPUs Annually in 1995**
  - 150 Million PCs and 7.5 Billion embedded CPUs + in 2000



$13,490M Total

1994 Worldwide
Microcontroller Revenue
($Million U.S.)



2,683M Total

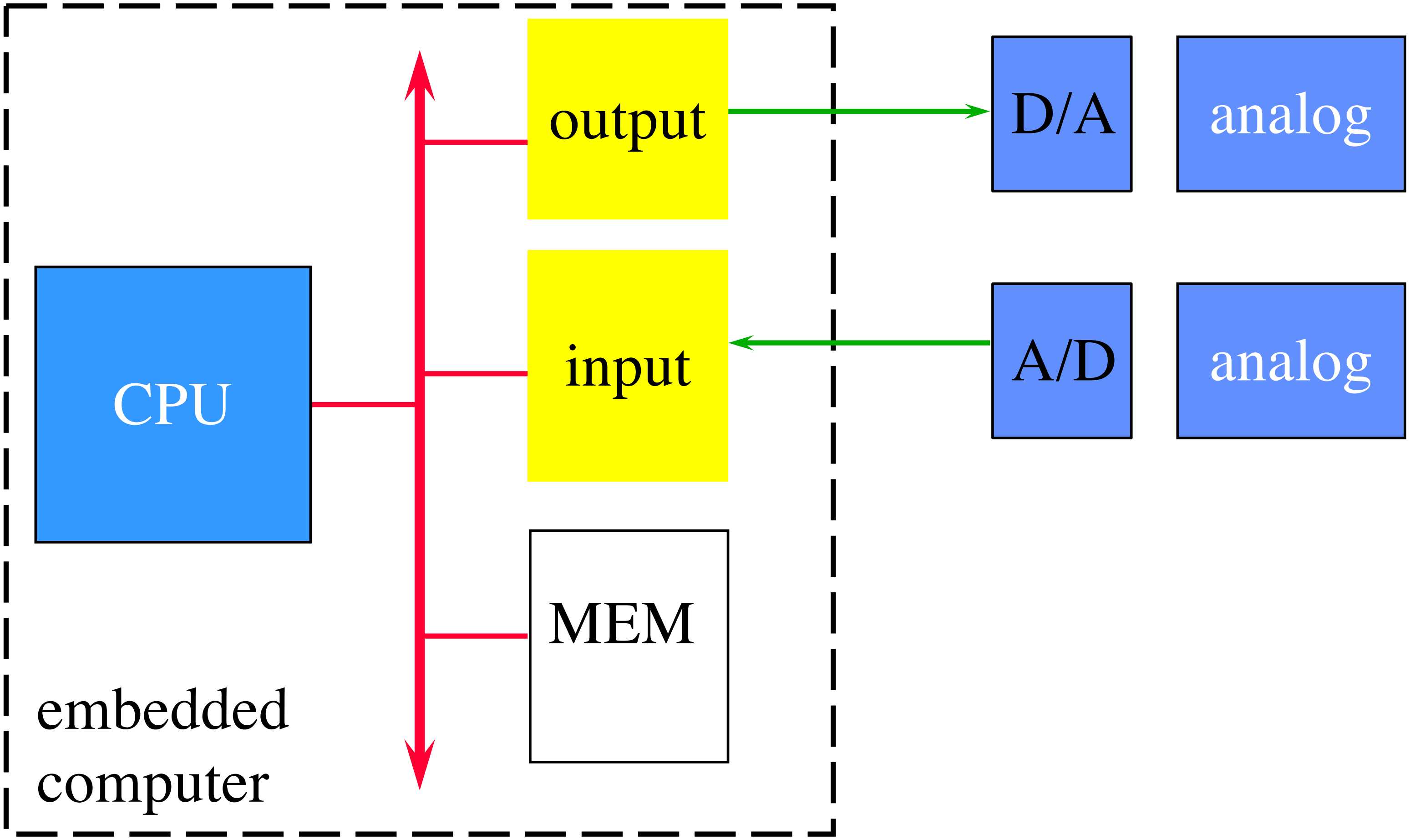
1994 Worldwide
Microcontroller Units
(Million Devices)

Approximated from EE Times, March 20, 1995
Source: The Information Architects

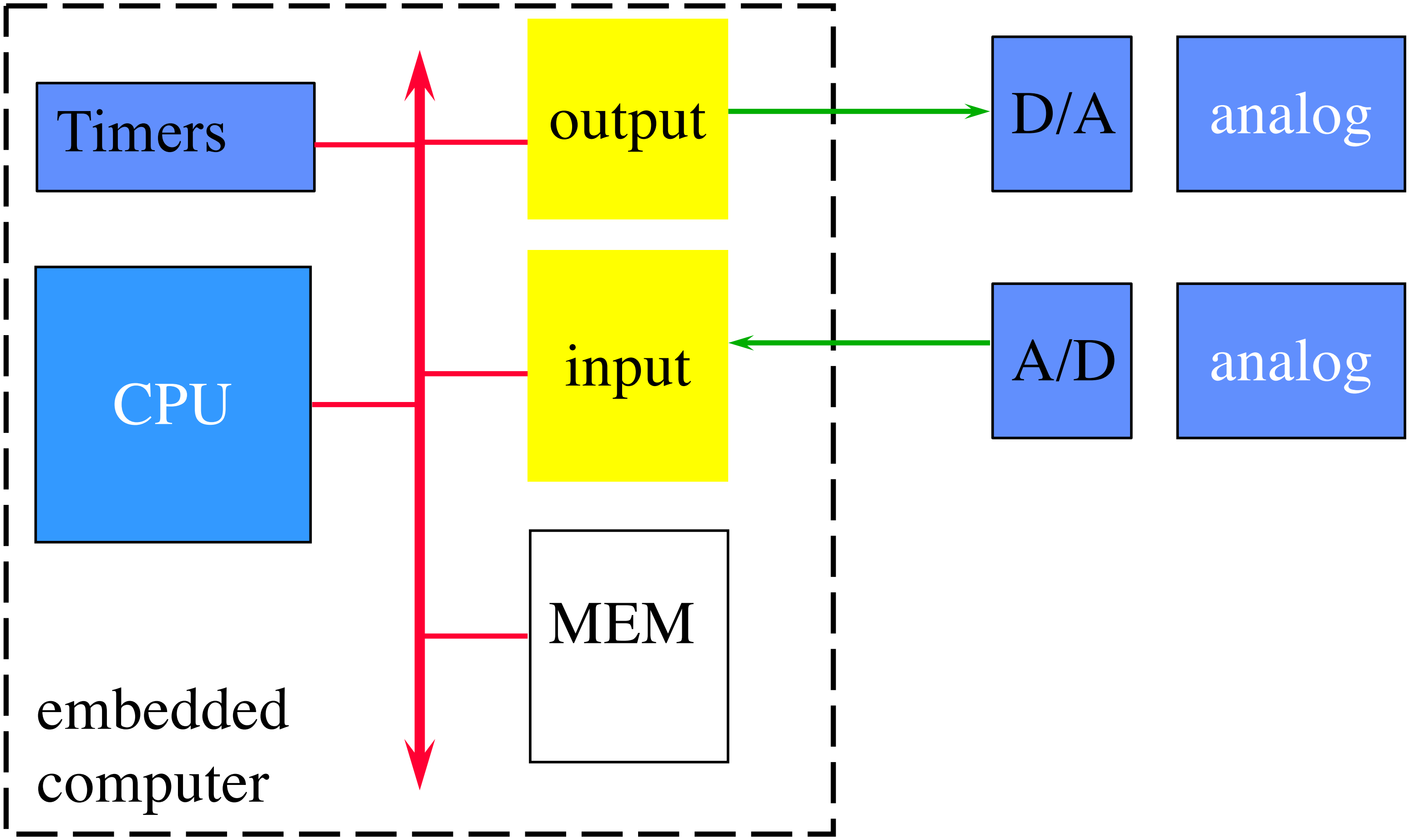
From Koopman http://www.ece.cmu.edu/~ece649/lectures/01_intro.pdf

> 99% of CPU's sold go into Embedded Systems

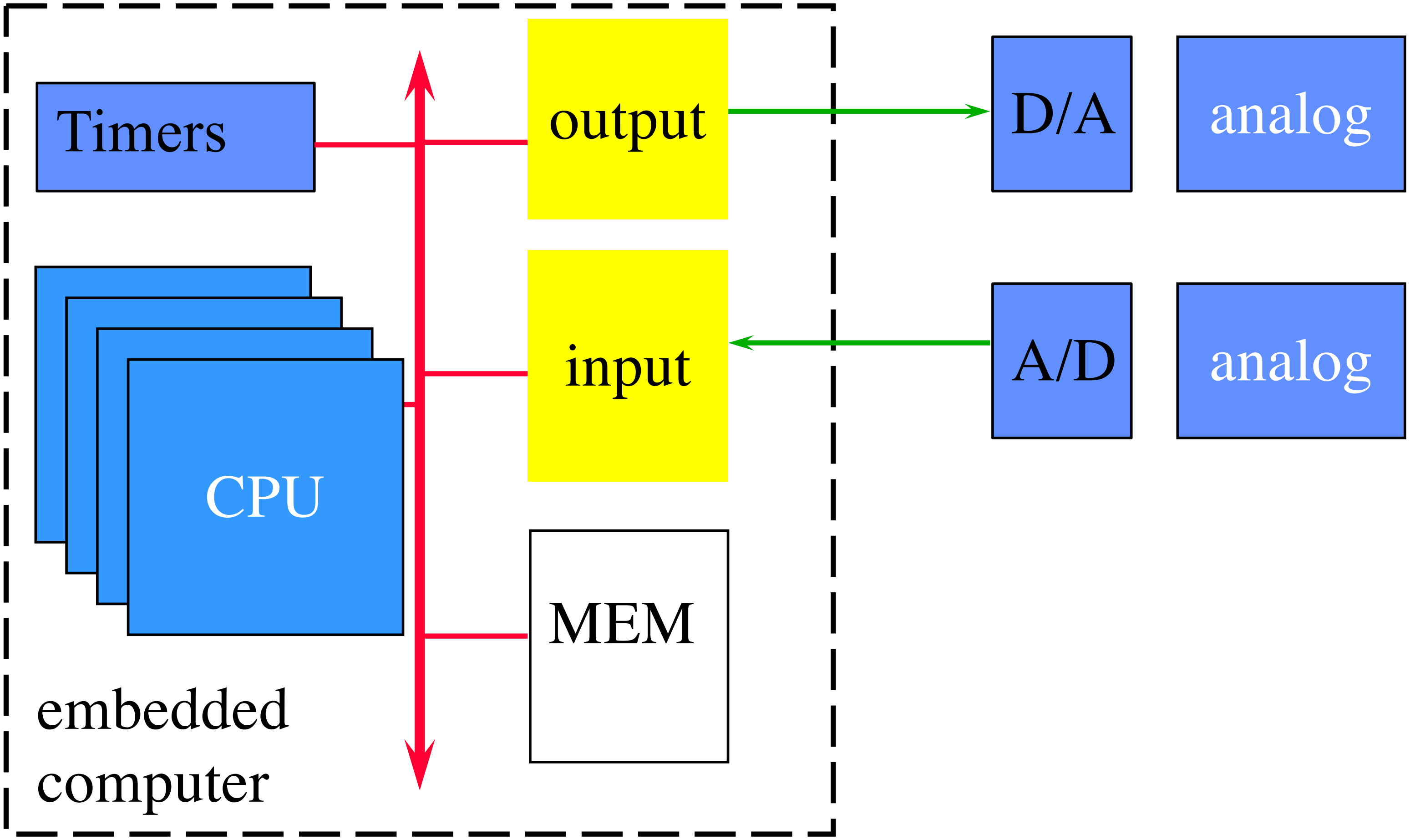
“Generic” Embedded Computer



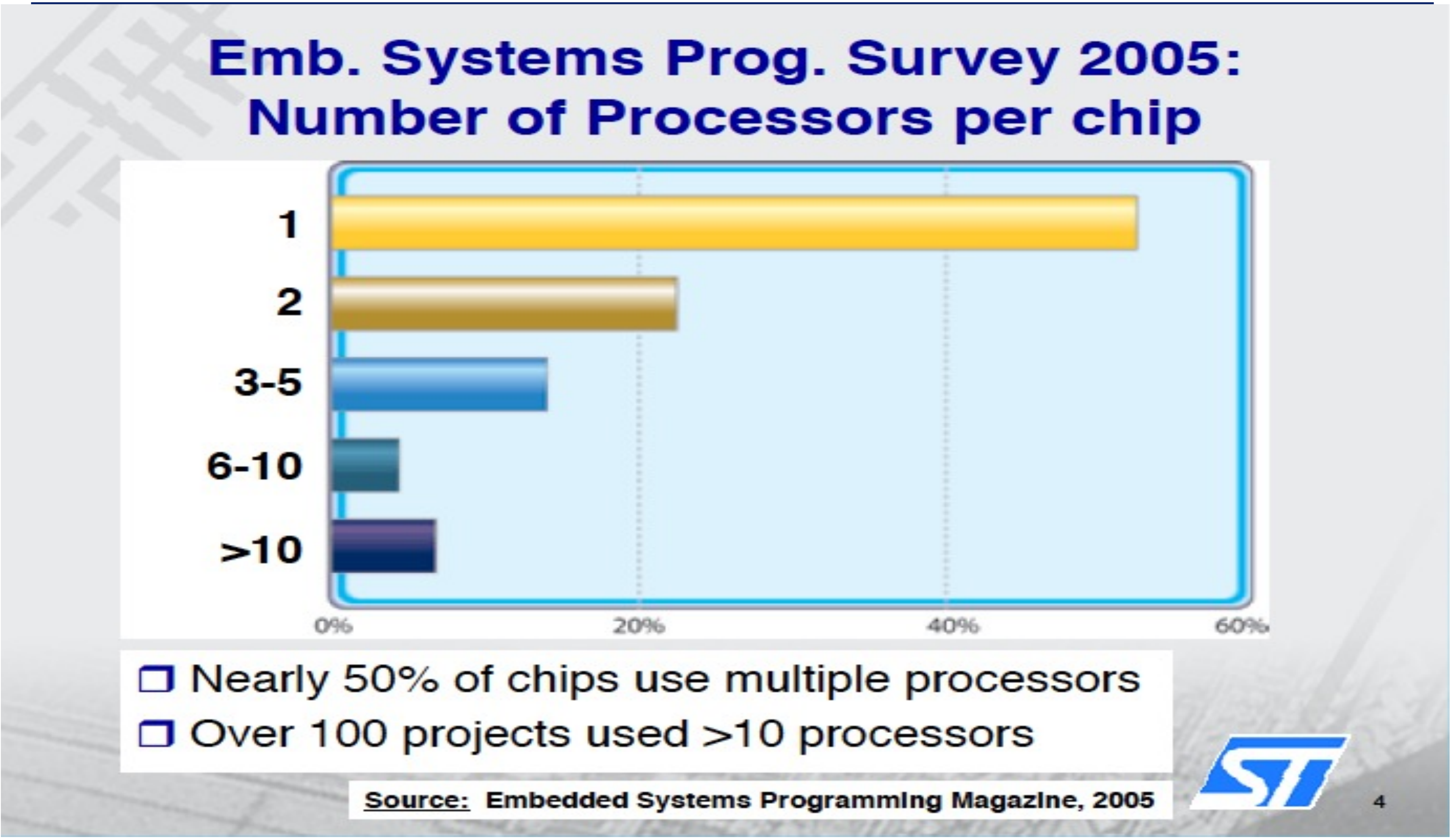
“Generic” Embedded Computer



"Generic" Embedded Computer



Meeting Performance



Emb. Systems Prog. Survey 2005:
Number of Processors per chip

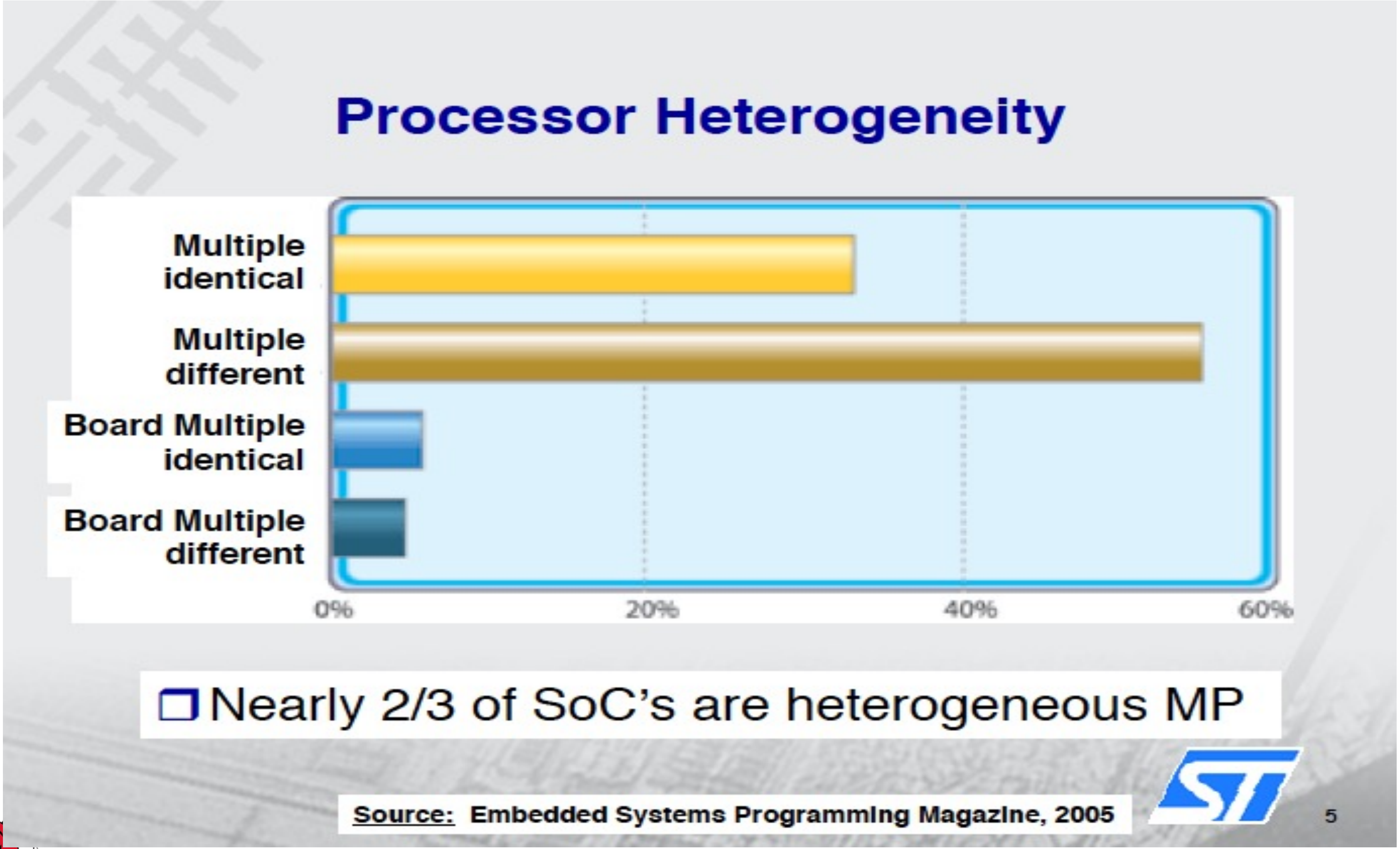
- ❒ Nearly 50% of chips use multiple processors
- ❒ Over 100 projects used >10 processors

Source: Embedded Systems Programming Magazine, 2005

4

From Pierre Paulin STMicroelectronics Presentation

Meeting Performance



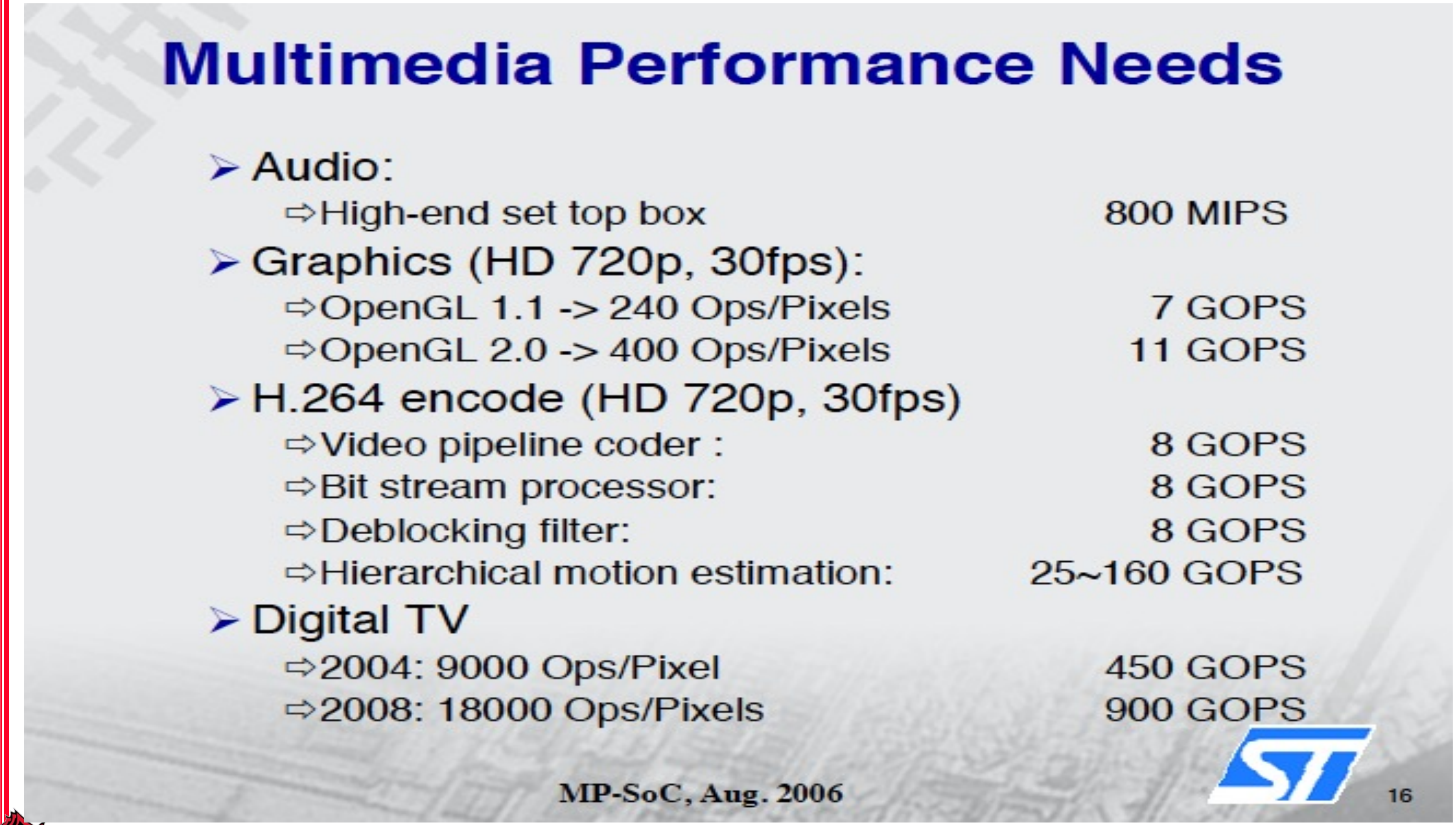
Processor Heterogeneity

❒ Nearly 2/3 of SoC's are heterogeneous MP

Source: Embedded Systems Programming Magazine, 2005

From Pierre Paulin STMicroelectronics Presentation

Performance Drivers



Multimedia Performance Needs

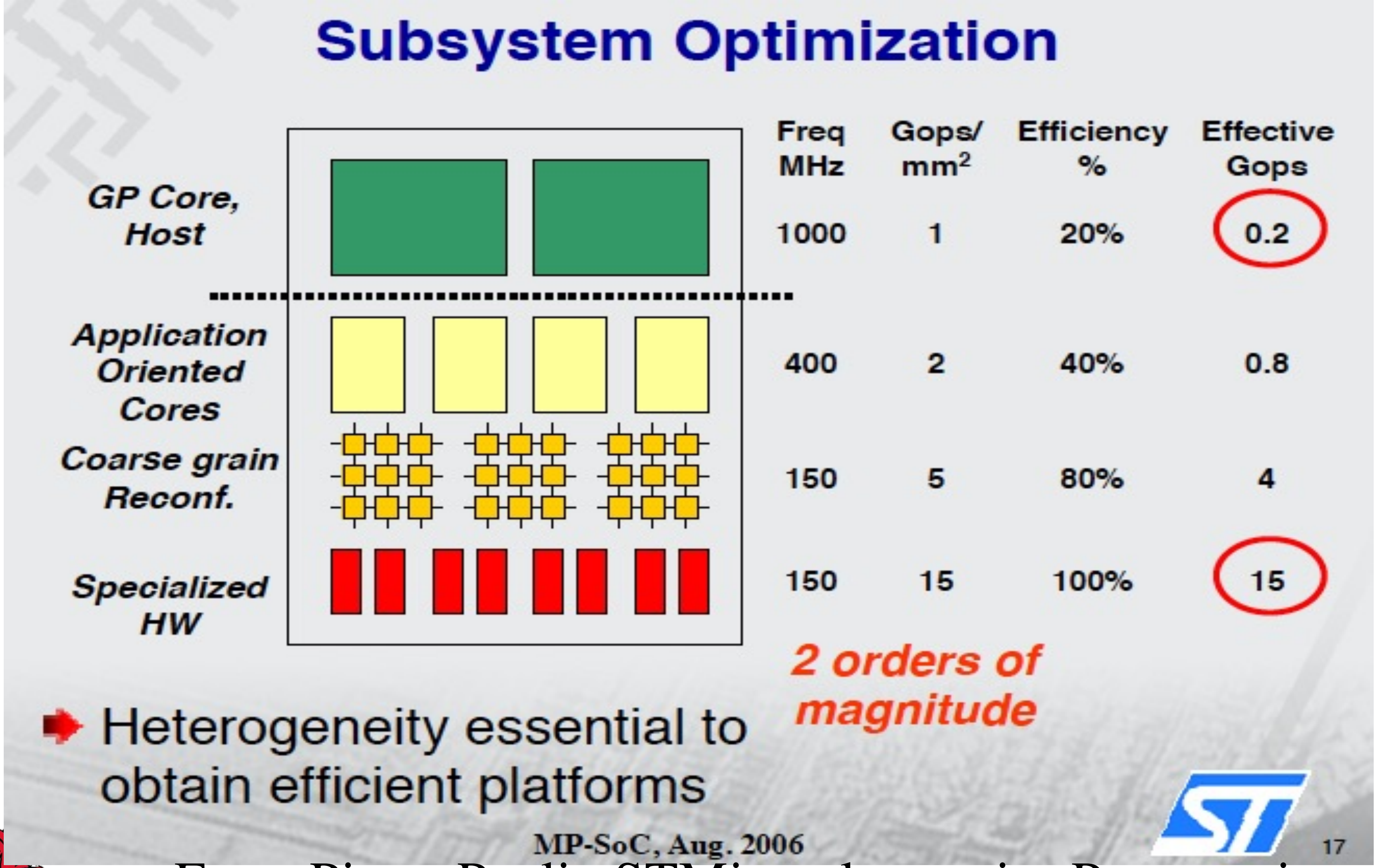
➢ Audio:	
⇨High-end set top box	800 MIPS
➢ Graphics (HD 720p, 30fps):	
⇨OpenGL 1.1 -> 240 Ops/Pixels	7 GOPS
⇨OpenGL 2.0 -> 400 Ops/Pixels	11 GOPS
➢ H.264 encode (HD 720p, 30fps)	
⇨Video pipeline coder :	8 GOPS
⇨Bit stream processor:	8 GOPS
⇨Deblocking filter:	8 GOPS
⇨Hierarchical motion estimation:	25~160 GOPS
➢ Digital TV	
⇨2004: 9000 Ops/Pixel	450 GOPS
⇨2008: 18000 Ops/Pixels	900 GOPS

MP-SoC, Aug. 2006

16

From Pierre Paulin STMicroelectronics Presentation

Component Performance



Subsystem Optimization

	Freq MHz	Gops/ mm^2	Efficiency %	Effective Gops
GP Core, Host	1000	1	20%	0.2
Application Oriented Cores	400	2	40%	0.8
Coarse grain Reconf.	150	5	80%	4
Specialized HW	150	15	100%	15

2 orders of magnitude

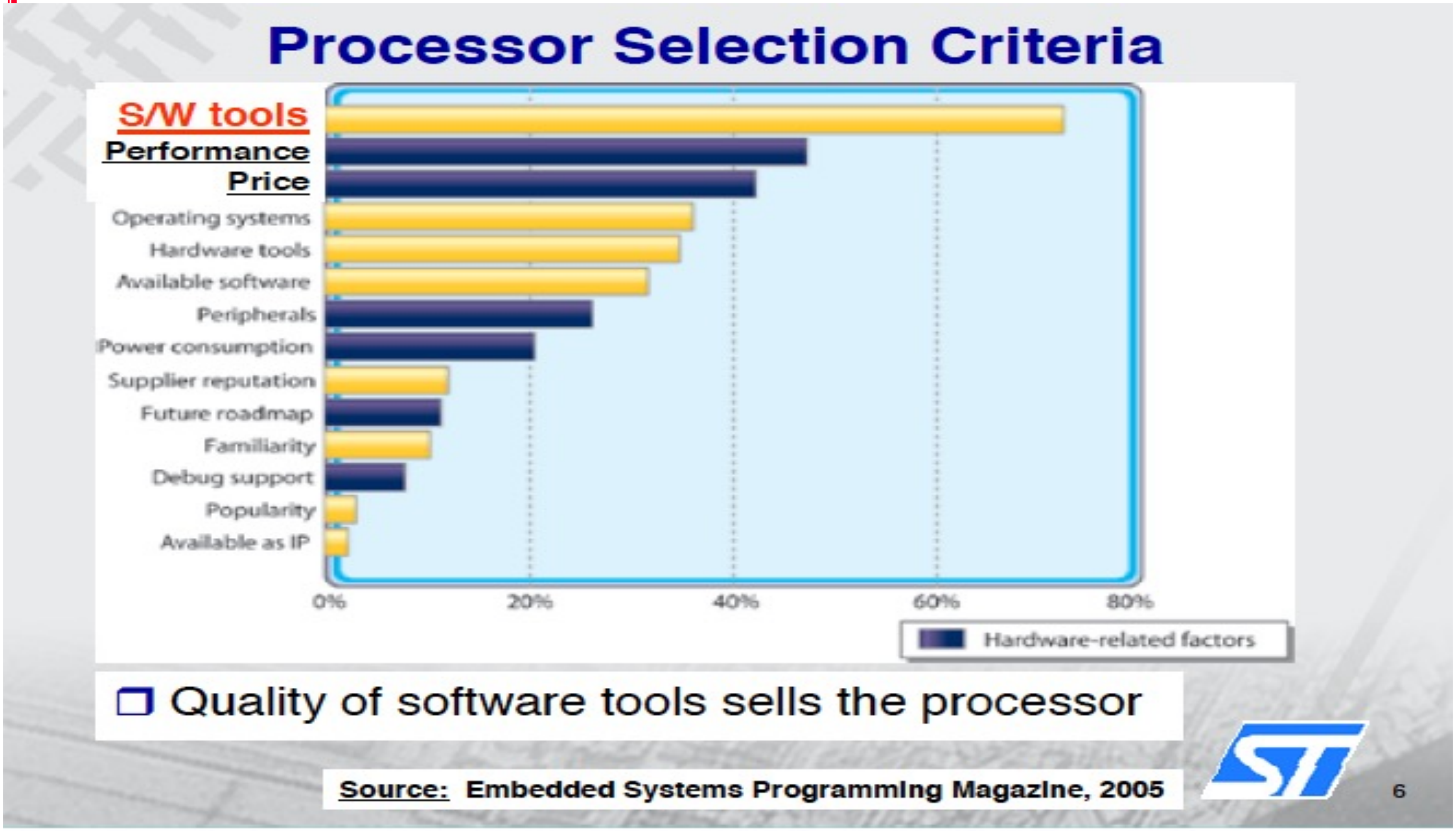
- Heterogeneity essential to obtain efficient platforms

MP-SoC, Aug. 2006

17

From Pierre Paulin STMicroelectronics Presentation

Software Considerations Important !



Processor Selection Criteria

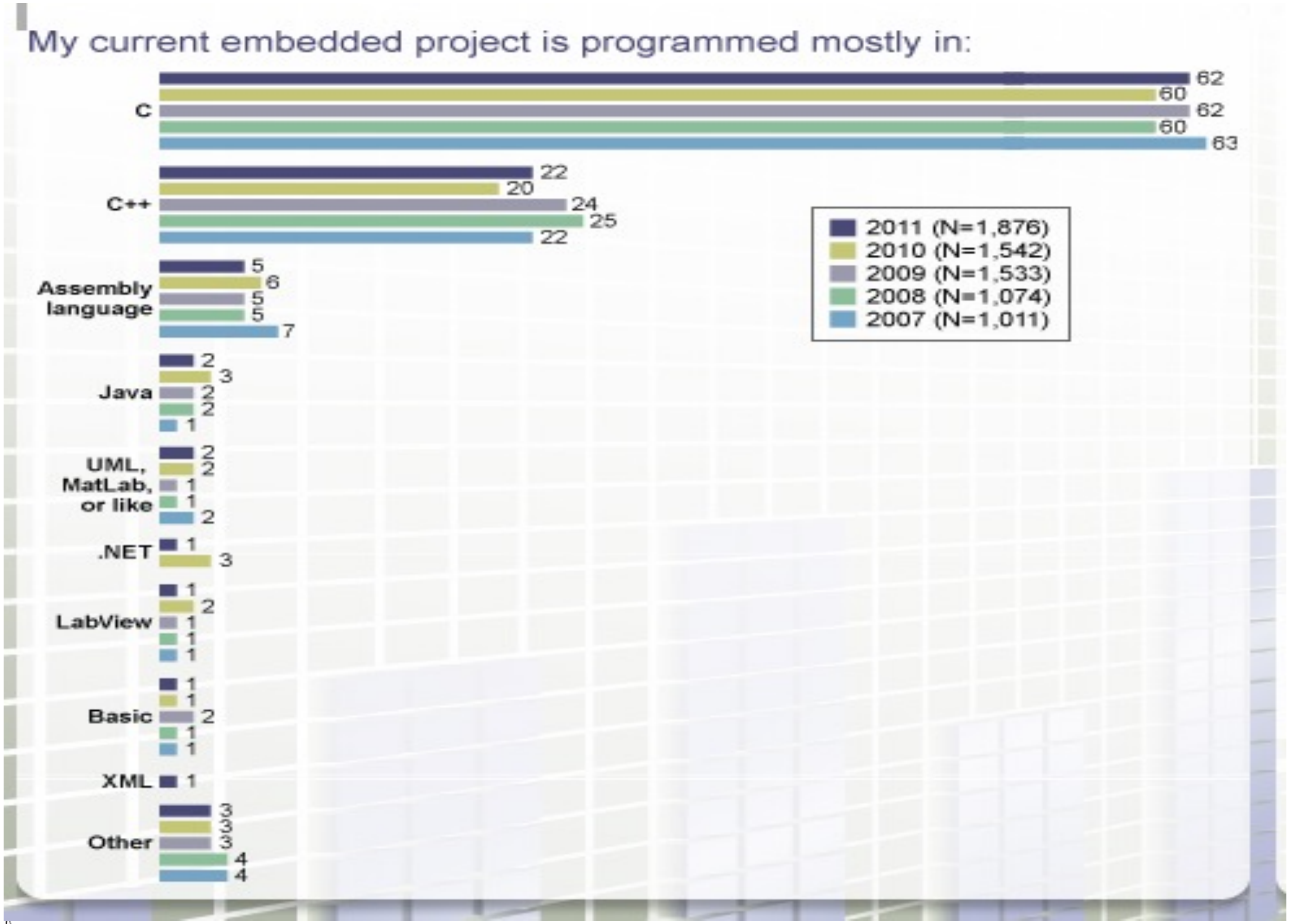
❐ Quality of software tools sells the processor

Source: Embedded Systems Programming Magazine, 2005

6

From Pierre Paulin STMicroelectronics Presentation

Languages Used



2011 EE Times Surveys

- 25% of the respondents are considering using embedded Linux in their next design.
- The software engineering staff has almost as much say as hardware staff choosing a microprocessor
- The use of multicore, while still low, continues to rise.
- Outsourcing continues to rise, with India growing its lead as the top outsourcing destination.

Future Trends

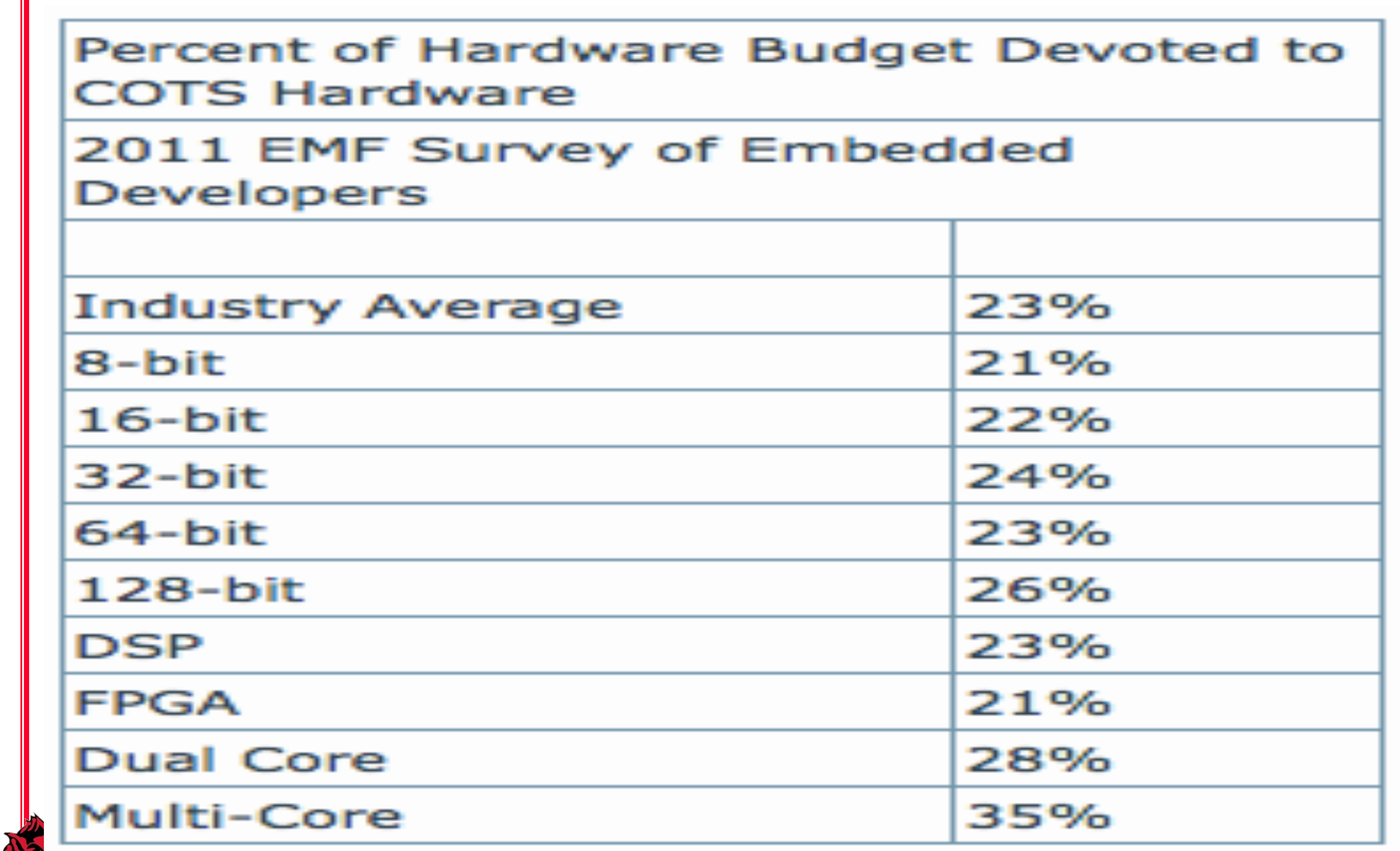
- > 20,000 downloads of FreeRTOS from embedded.com
- Processor upgrades growing
  - New Features
  - Previous Processor to Slow
  - New Processor has better growth Plan
  - #1 Processor ?

Todays Hardware Components

- Microcontroller-based systems
- DSP processor-based systems
- ASIC technology
- FPGA technology

- What is in Your Cell Phone ?

EE Times 2011 Survey-Hardware

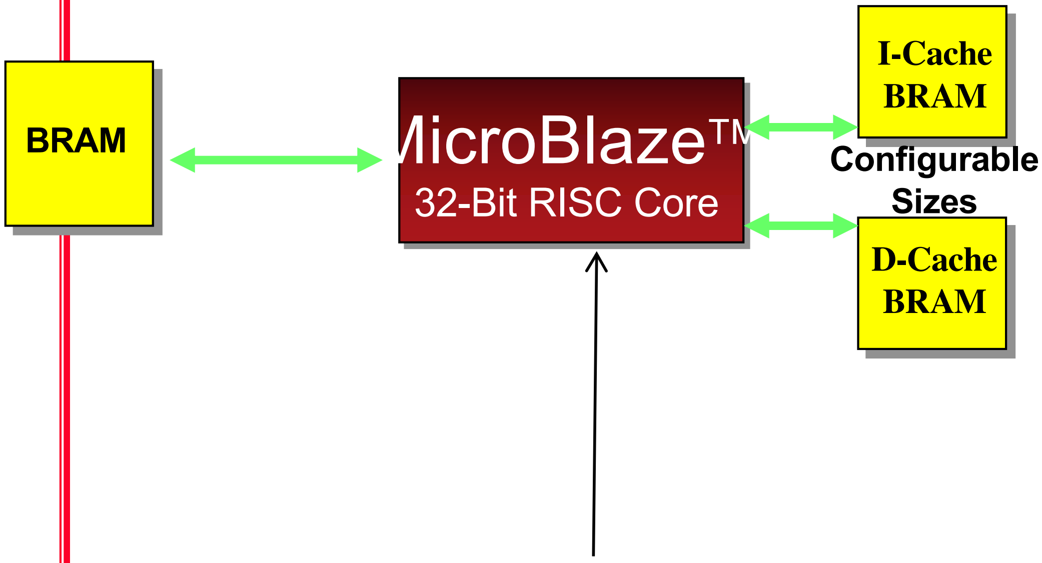


Percent of Hardware Budget Devoted to COTS Hardware	
2011 EMF Survey of Embedded Developers	
Industry Average	23%
8-bit	21%
16-bit	22%
32-bit	24%
64-bit	23%
128-bit	26%
DSP	23%
FPGA	21%
Dual Core	28%
Multi-Core	35%

Our Class Components

- We will Create System on Chip (SoC) Within an FPGA (Poor Mans Sandbox)
- Hardware
- 1 CPU = Microblaze soft IP
  - + System Components
    - Bus, I/O Devices, Interrupts
  - + Accelerator (Heterogeneous)
- Software = C, Assembler

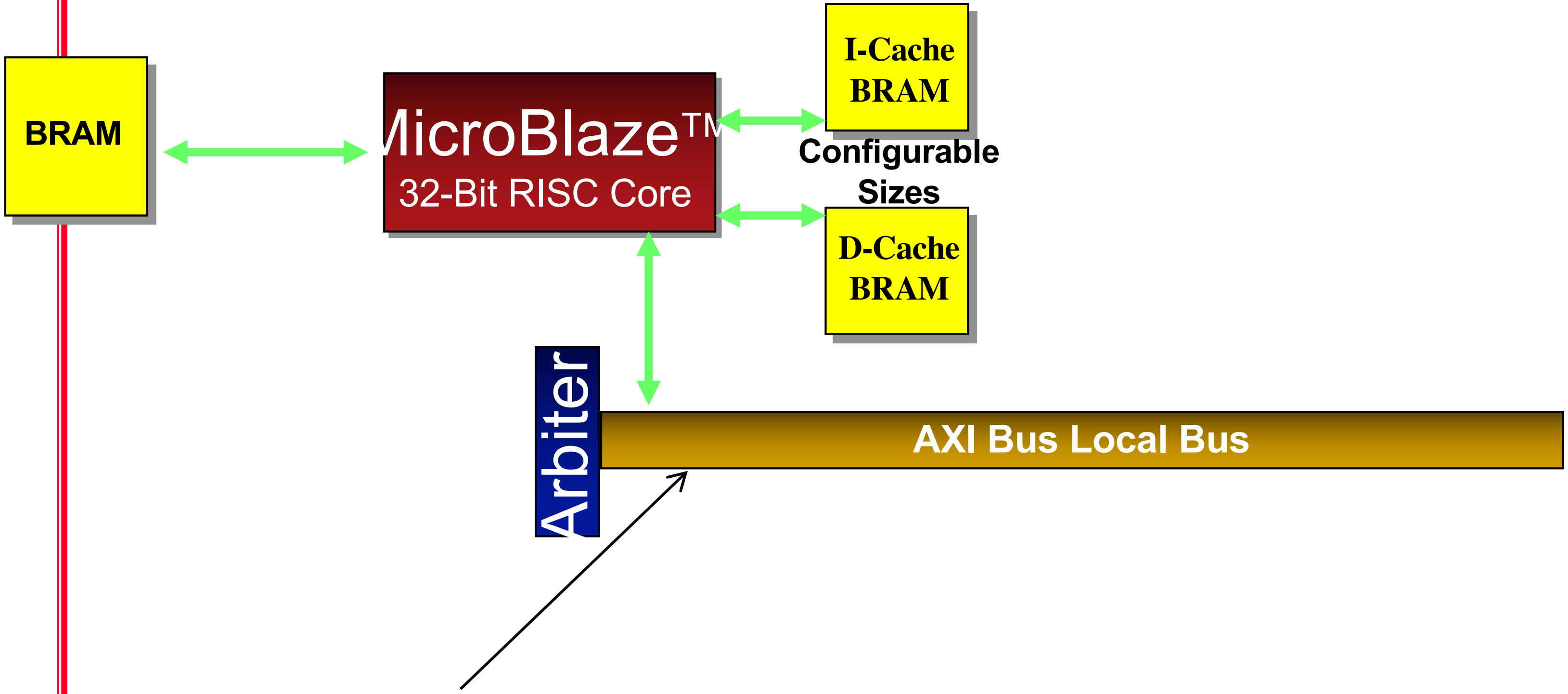
MicroBlaze Processor-Based Embedded Design



We will study architecture+ISA

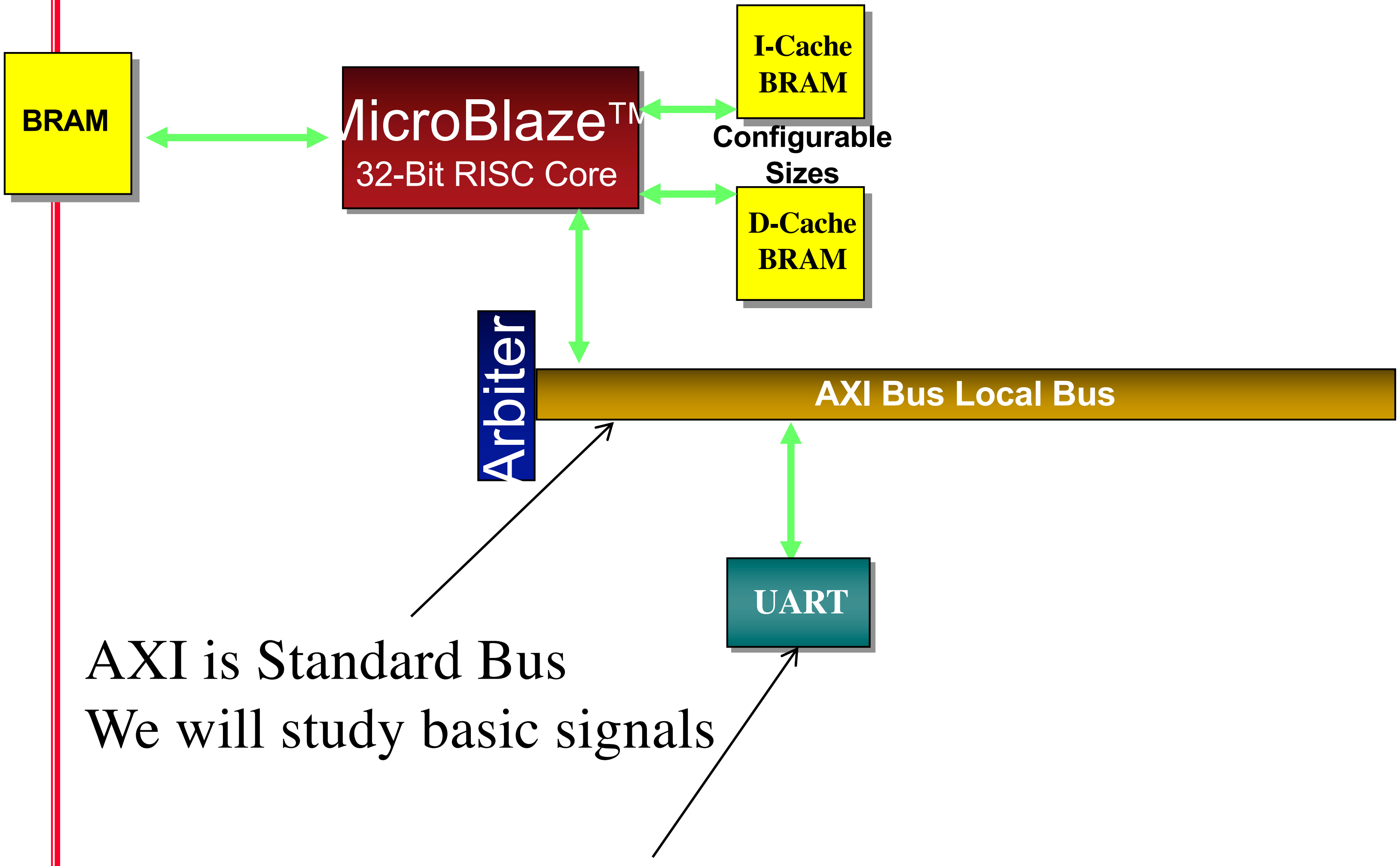
We will work in both Assembler and C

MicroBlaze Processor-Based Embedded Design



AXI is Standard Bus
We will study basic signals

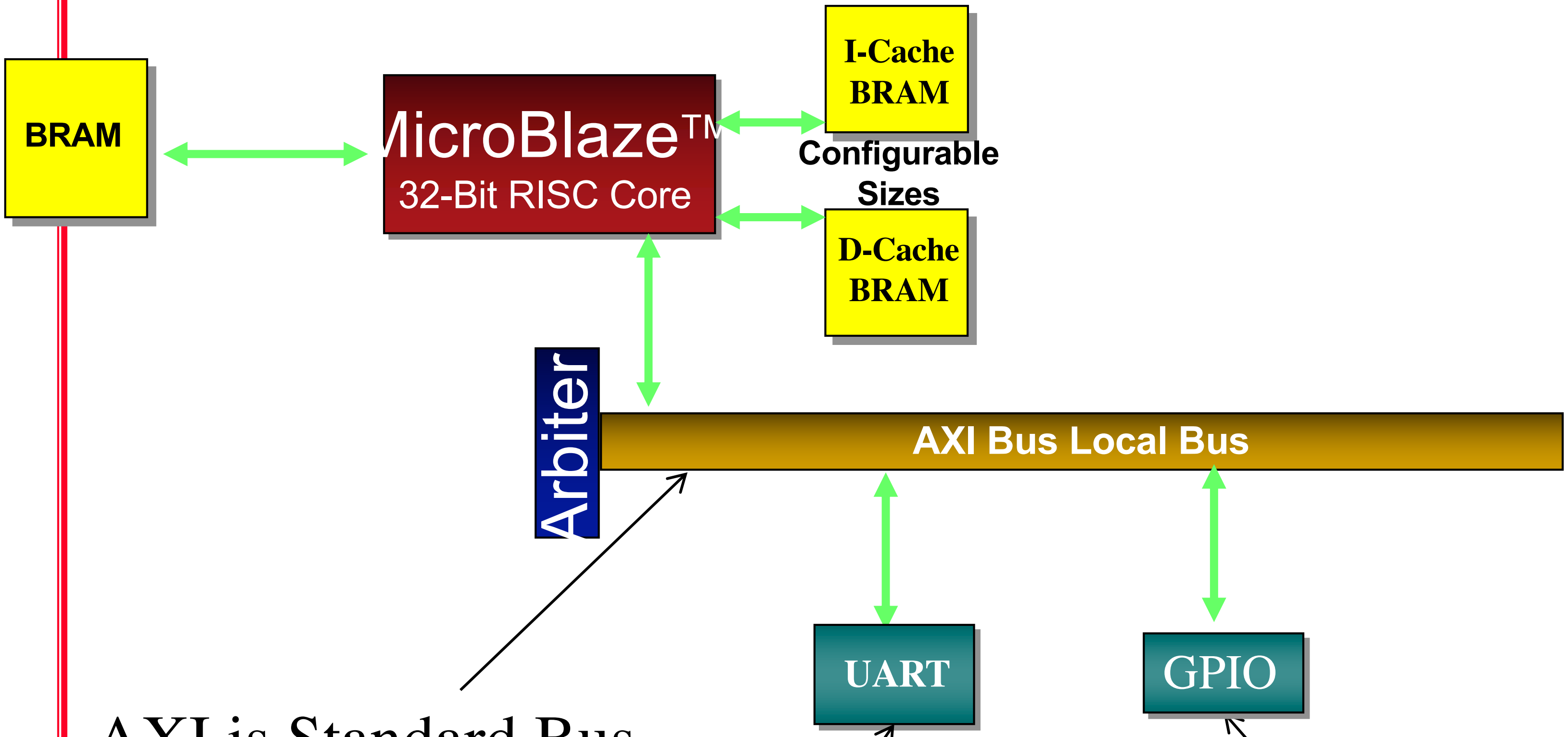
MicroBlaze Processor-Based Embedded Design



AXI is Standard Bus
We will study basic signals

Serial Port to PC

MicroBlaze Processor-Based Embedded Design

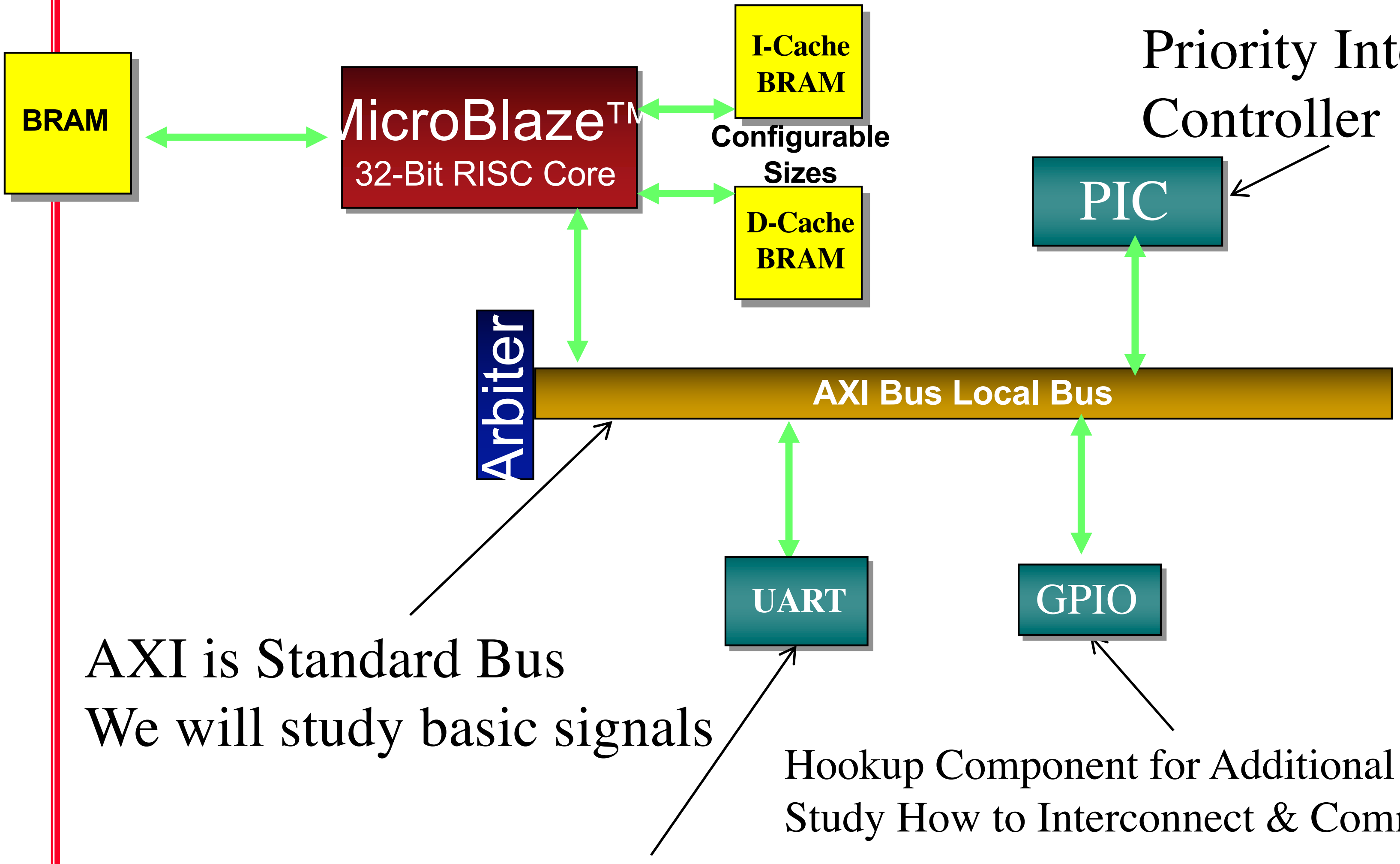


AXI is Standard Bus
We will study basic signals

Hookup Component for Additional Sensors
Study How to Interconnect & Communicate

Serial Port to PC

MicroBlaze Processor-Based Embedded Design



Priority Interrupt Controller

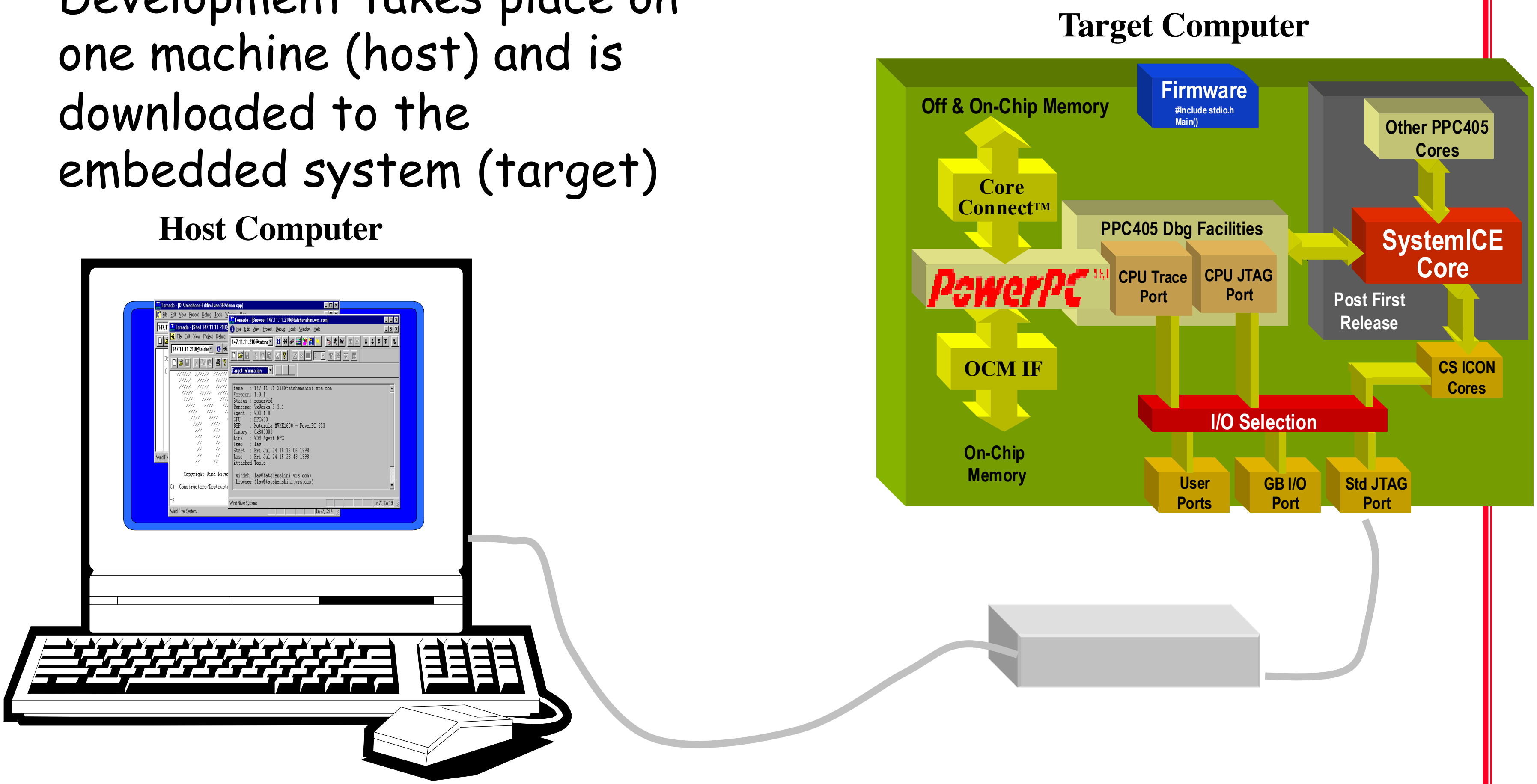
AXI is Standard Bus
We will study basic signals

Hookup Component for Additional Sensors
Study How to Interconnect & Communicate

Serial Port to PC

Embedded versus Desktop Development Environments

- Development takes place on one machine (host) and is downloaded to the embedded system (target)



A cross-compiler is run on the host

From Xilinx EDK Presentation

Our Lab Environment

- Embedded Development Kit (EDK)?
  - The Embedded Development Kit is the Xilinx software suite for designing complete embedded programmable systems
  - The kit includes all the tools, documentation, and IP that you require for designing systems with Xilinx MicroBlaze™ soft processor cores
  - It enables the integration of both hardware and software components of an embedded system

From Xilinx EDK Presentation

Next Week

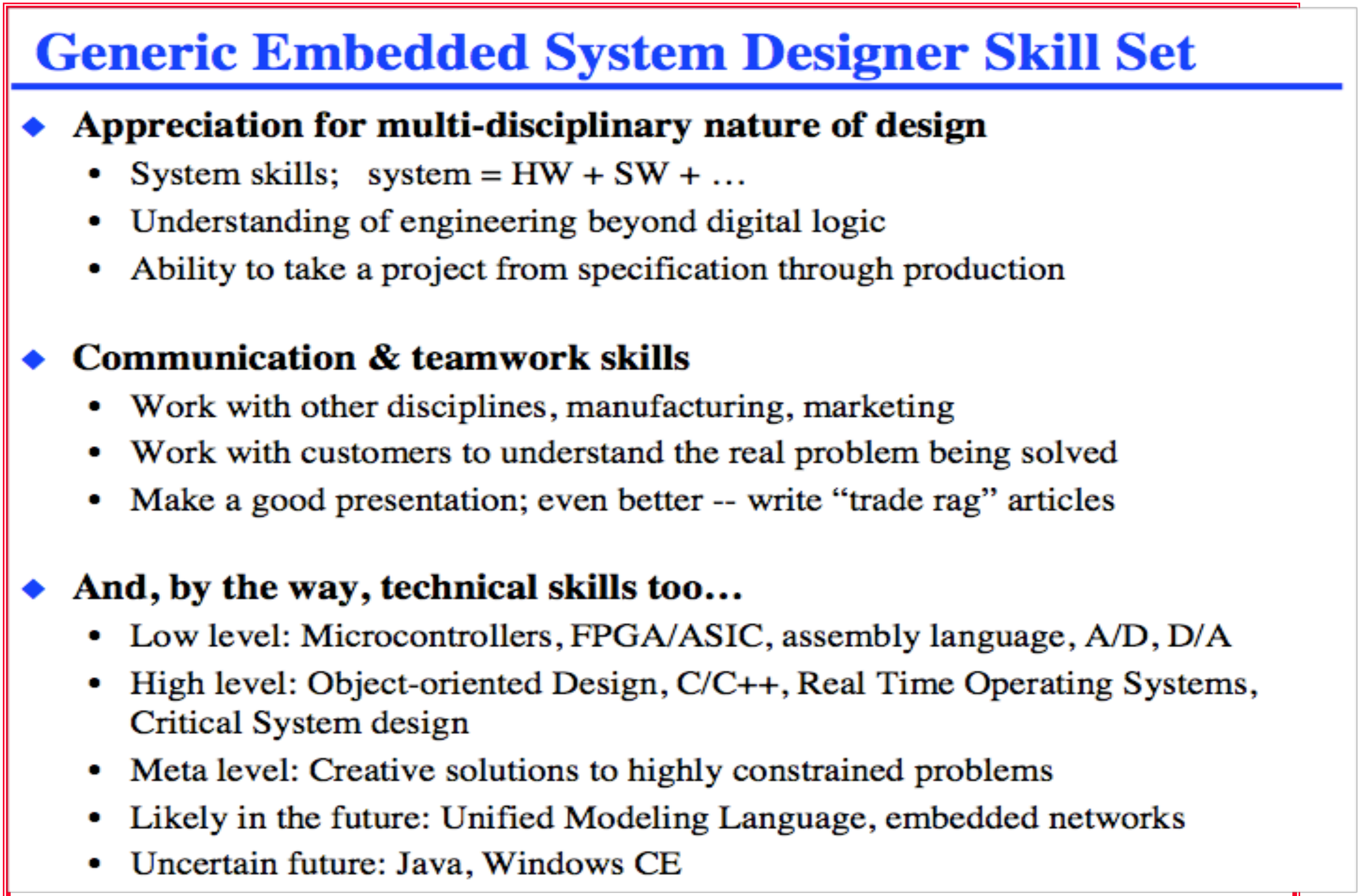
- Lab: Prelab
  - Build your SoC
- Next Lectures
  - Pointer Review and bit twiddling in C

Embedded Systems

- What you will study:
  - Embedded Systems interfacing and design
  - Where Hardware and Software co-exist
    - Hardware Organization:
      - CPU: Basic components (how to build in CSCE 2214)
      - Bus Interfacing: Signals and protocols for communication between CPU & all other components
      - Memory: Decoding and hooking up to Bus
      - Peripherals
        - I/O getting data in and out
        - Priority Interrupt Controller: How things get the CPU's attention
        - Custom Components: Accelerators and additions

Embedded Systems

- What you will study:
  - Embedded Systems interfacing and design
  - Where Hardware and Software co-exist
    - Software Organization:
      - Internal CPU Arithmetic and Boolean Instructions
      - Data Movement into and out of CPU: How to communicate with other system components
      - Protocol Stacks (How C/Java Functions & Subroutines actually get implemented)
    - Interrupt Routines:
      - Special Instructions that allows external devices to request service



Generic Embedded System Designer Skill Set

- **Appreciation for multi-disciplinary nature of design**
  - System skills; system = HW + SW + …
  - Understanding of engineering beyond digital logic
  - Ability to take a project from specification through production
- **Communication & teamwork skills**
  - Work with other disciplines, manufacturing, marketing
  - Work with customers to understand the real problem being solved
  - Make a good presentation; even better -- write “trade rag” articles
- **And, by the way, technical skills too…**
  - Low level: Microcontrollers, FPGA/ASIC, assembly language, A/D, D/A
  - High level: Object-oriented Design, C/C++, Real Time Operating Systems, Critical System design
  - Meta level: Creative solutions to highly constrained problems
  - Likely in the future: Unified Modeling Language, embedded networks
  - Uncertain future: Java, Windows CE

From Koopman http://www.ece.cmu.edu/~ece649/lectures/01_intro.pdf

Summary

- This course will give you appreciation for the fun and difficulty of designing and building and embedded system
  - If you are a “Software Person”: you will learn how your software is being implemented. You will learn how to write embedded software
  - If you are a “Hardware Person”: you will learn how your hardware is being used and controlled. You will learn how to create hardware that is usable by software.