CSCE 4114
Embedded Systems
Class website:
http://hthreads.github.io/classes/embedded-systems

David Andrews

Rm 527 JBHT

dandrews@uark.edu

Welcome !

- Today we will just cover programmatics and introduction to Embedded Systems

Office Hours :

Andrews: MWF 3:00 – 4:00

Office 527 JBHT

TA: Tendayi Kamucheka:

Hours 11:00 -12:00 M/W

9:00 – 10:00 Fri

Office TA Bullpen 4th floor

tfkamuch@uark.edu

Programmatics

Grade Breakdown

1. 2 exams (Mid/Final) := 30% each.
   2nd Exam builds on earlier material but is NOT comprehensive.
2. Homework := 10%
3. Laboratories := 30%

Midterm Tentative Oct 11. Will confirm2 weeks before.

Class = Lecture + Lab

1. Lectures: Concepts and theory
   3:1 rule applies (Three hours study for every hour lecture)
2. Labs: Application of the Theory
   Frustrating but satisfying and fun !
   You get to build autonomous car !

Programmatics

- Homework: Assignments given in class
  Usual due date is 1 week later.
  - Gives you a chance to read, try and then ask questions in next class.
  - Due at Beginning of class thru Blackboard.
    - Each late day costs 25%
    - Will waive for valid reason presented ***before*** due date

Programmatics

- Labs
  - ~6 Labs during semester.
  - We will use Xilinx Arty Boards as System On Chip (SoC)
  - You will assemble an Automonous Car
  - Week #1: Prelab (downloading software) at home
  - Lab writeups + Grading explained in more detail in lab (Week 2)

Programmatics

- Textbook:

  "Programming Embedded Systems"

  Vahid, Givargis, Miller

1. Sign in or create an account at:

   learn.zybooks.com

2. Enter zyBook code:

   UARKCSCE4114AndrewsFall2024

3. Subscribe

   $64 Open August 7st -> Dec 23th

Programmatics

- Lecture Materials:
  1. From Textbook:
  2. Additional materials including vendor data sheets posted on webpage
  3. Reading assignments will be posted the weekend before prior to coverage. You should read and familiarize yourself with materials before lecture.

"Most" Materials posted as .pdf
However some will be real time on the board
(hint If you miss class you may miss important info)*

Programmatics

- **Lecture attendance:** Expected but attendance won't be taken. Lectures will not be recorded or streamed. You can record the lectures for your personal us but you may not post or share.

- **Lab attendance:** Expected in your designated lab section but attendance won't be taken. There will be pre-lab questions that you will need to show the TA before you can enter the lab. This is to force you to read and familiarize yourself with what you will be doing in lab to make the most of your time as well as the TAs. We share the lab with Digital Design but you may come into the lab anytime during normal business hours that our other labs or Digital Design labs are not meeting.

- **Collaboration:** Each student is responsible for their own coding, lab reports, homework etc. You cannot copy or modify code or solutions from anyone including classmates, Chegg, ChatGPT, etc. Why ? To quote your author Frank Vahid " This of your code/hardware as an essay – you can't copy somebody else's essay, change some words, understand it, and call it yours. Having said that I strongly encourage you to form study groups. What ????

Programmatics

- **Getting Help:** If you have questions we are here to help! Outside of class and the lab you have options:

  1. Use the discussion board on blackboard. Your question might already be asked and answered. If not you can be the first and help classmates who may ask the same question later.

  2. Come to either of our office hours for face to face discussions. If you need to have a face to face and cannot make office hours then contact me and I will try and accommodate a meeting for a different time.

- **Helpful Hint:** You will have one week for each homework. As soon as the homework is posted read through it to make sure you understand what is being asked. This gives you time to drop by for office hours or post on the discussion board. Lack of planning on your part does not constitute a crisis on mine ☺ Do not wait until the night it is due and expect 24 on call service. It won't happen.

Programmatics

- Academic Honesty: Very important. You are required to do your own work.
  - Labs can be done by partners. It's ok to work together, get together after to discuss. However, you need to write up your own report in your own words.
  - TA will give you formats and expectations for reports.
  - Dishonesty will be dealt with swifty !
  - See the Universities Procedure at:

    http://provost.uark.edu/245.php

Assignment

- Lecture
  - Reading:
  - Ch 1.1 - What is an embedded system
  - Ch 2.1, 2.2 C and pointers
- Lab
  - Download and install the Xilinx Vivado webpack 2019.1 onto your laptop/home computer at….
  - https://www.xilinx.com/support/download.html and click on "Vivado Archive" which will take you to….

https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html

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.

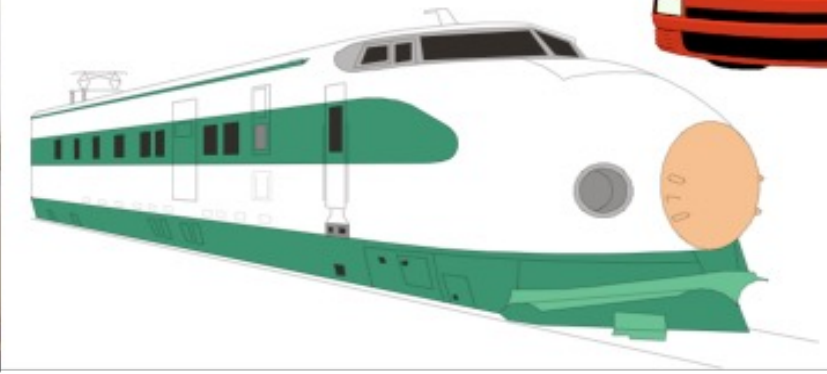
- By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs.

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

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

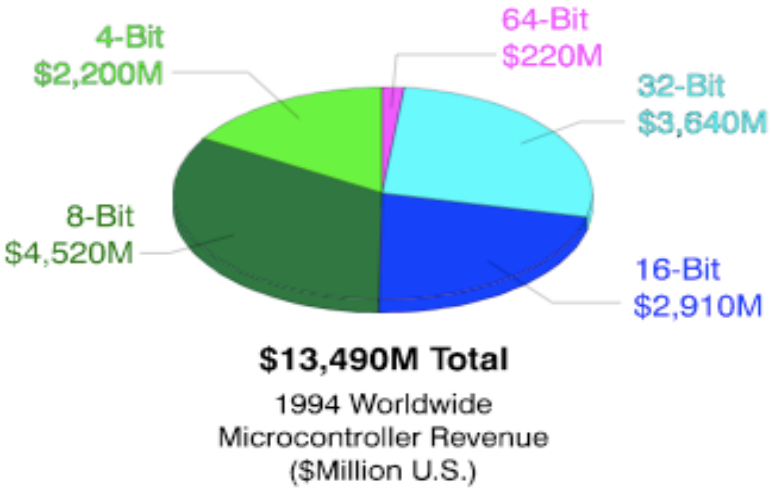
Embedded Versus Desktop

- 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 !

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

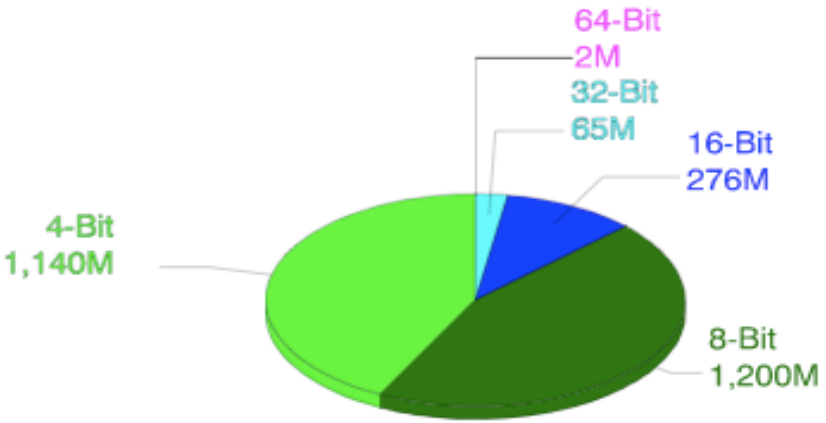
Small Computers Rule The Marketplace

- **~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 Worldworld
Microcontroller Revenue
($Million U.S.)



2,683M Total

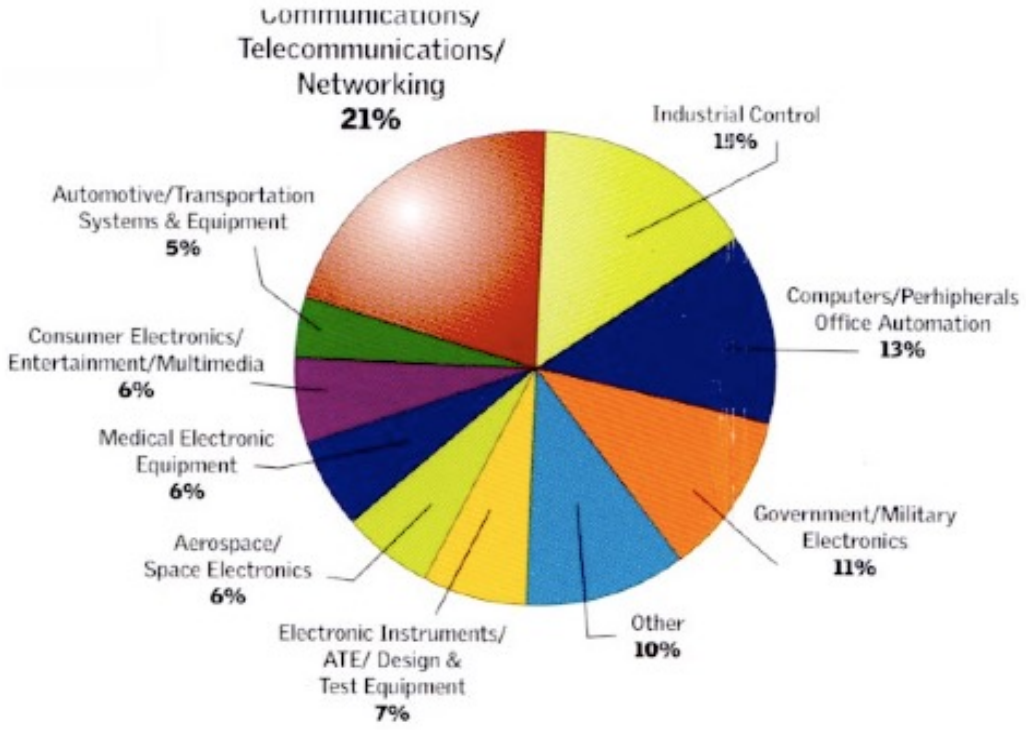
1994 Worldwide
Microcontroller Units
(Million Devices)

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

There Are Many Application Areas

Primary End Product of Embedded Subscribers

Source: *ESP* Dec. 1998 BPA Audit



Will You Be Working in One of These Areas ?

What is Most Popular Language Used In Embedded Systems ?

What is Most Popular Language Used In Embedded Systems ?

Language	% of embedded programmers with current project mostly in this language (2013)
C	60%
C++	21%
Assembly	5%
Java	3%
C#	2%
MATLAB/LabView	4%
Python	1%
.NET	1%
Other	4%

www.embedded.com, 2013 Embedded Market Study

CSCE 4114 Embedded Systems

- What will you 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

CSCE 4114 Embedded Systems

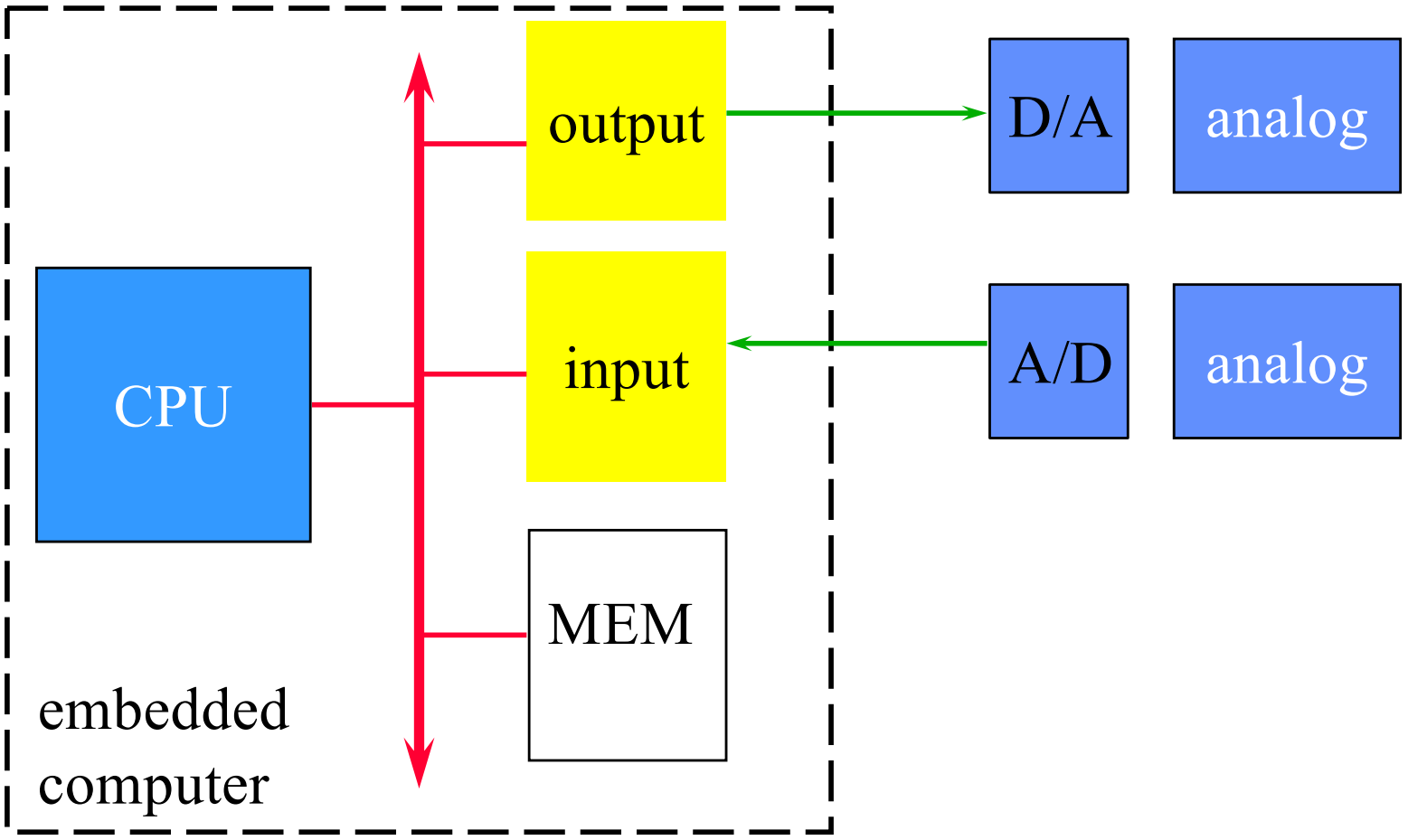
- What will you study ?
  - Embedded Systems interfacing and design
  - Where Hardware and Software co-exist
    - Software Organization:
      - Assembler := CPU's language.
        - Internal CPU Arithmetic and Boolean Instructions
        - Data Movement into and out of CPU: How to communicate with 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

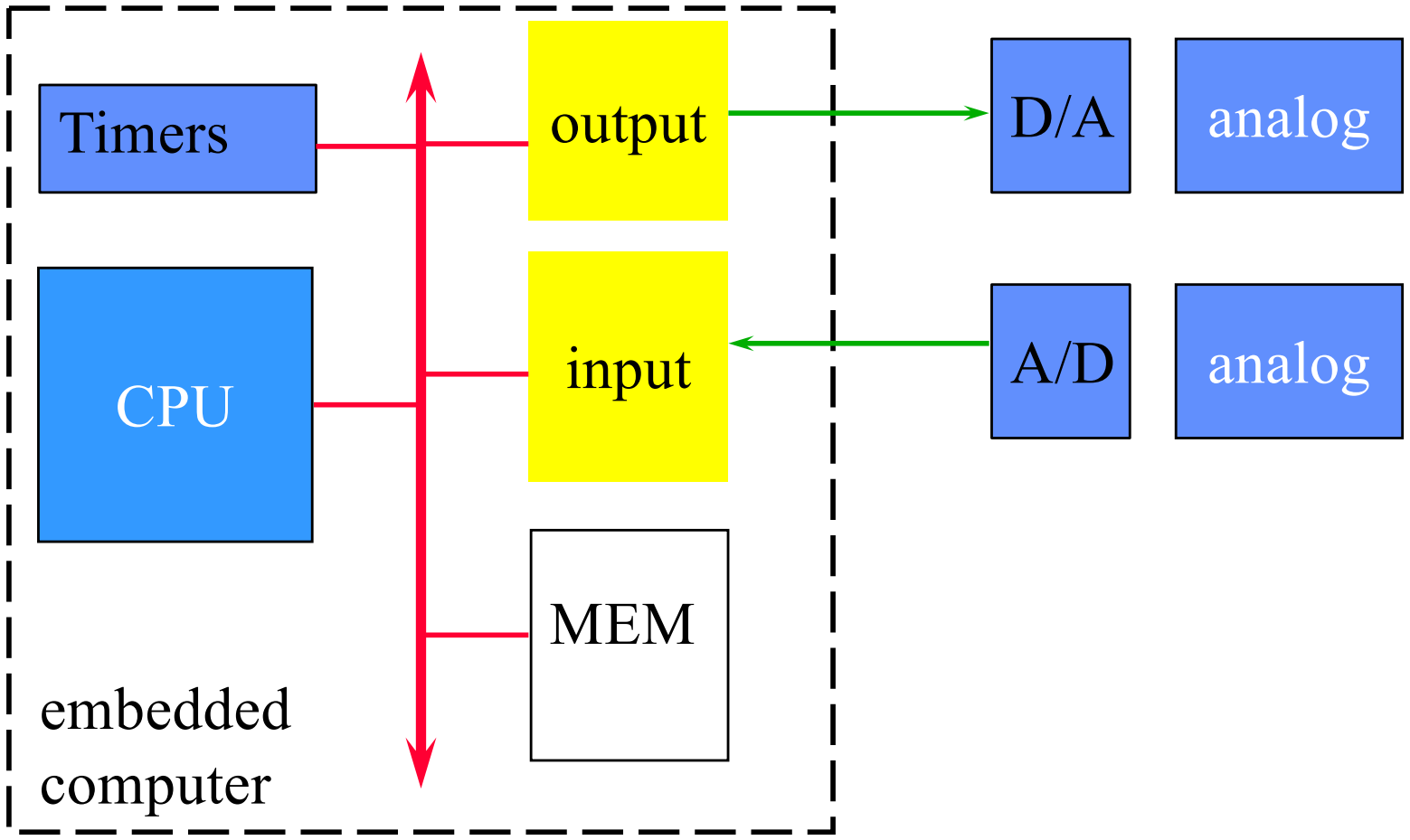
- **Appreciation for multi-disciplinary nature of design**
  - System skills; system = HW + SW + …
  - Understanding of engineering beyond digital logic
  - Ability to take a project from specification through production
- **Communication & teamwork skills**
  - Work with other disciplines, manufacturing, 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

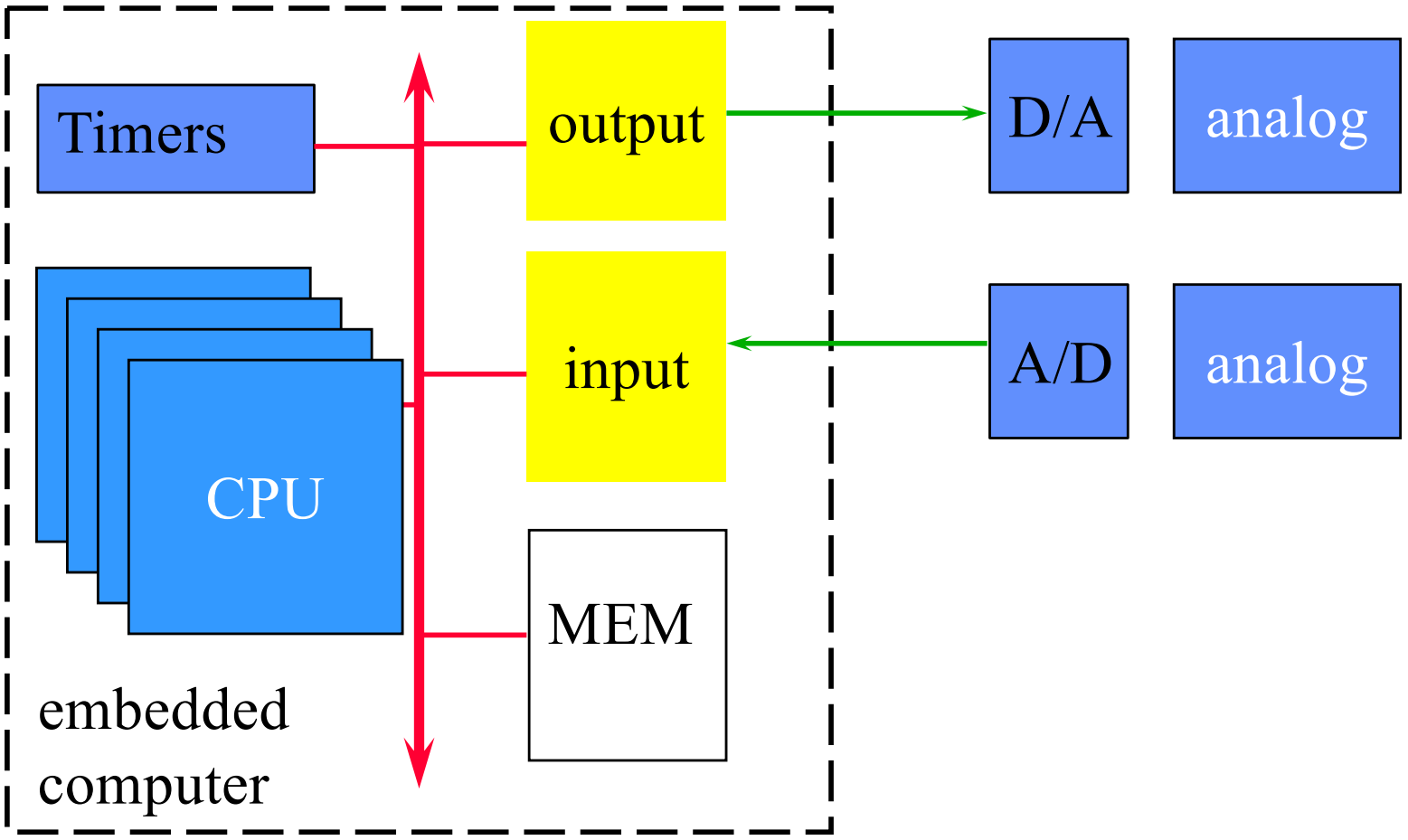
"Generic" Embedded Computer



"Generic" Embedded Computer



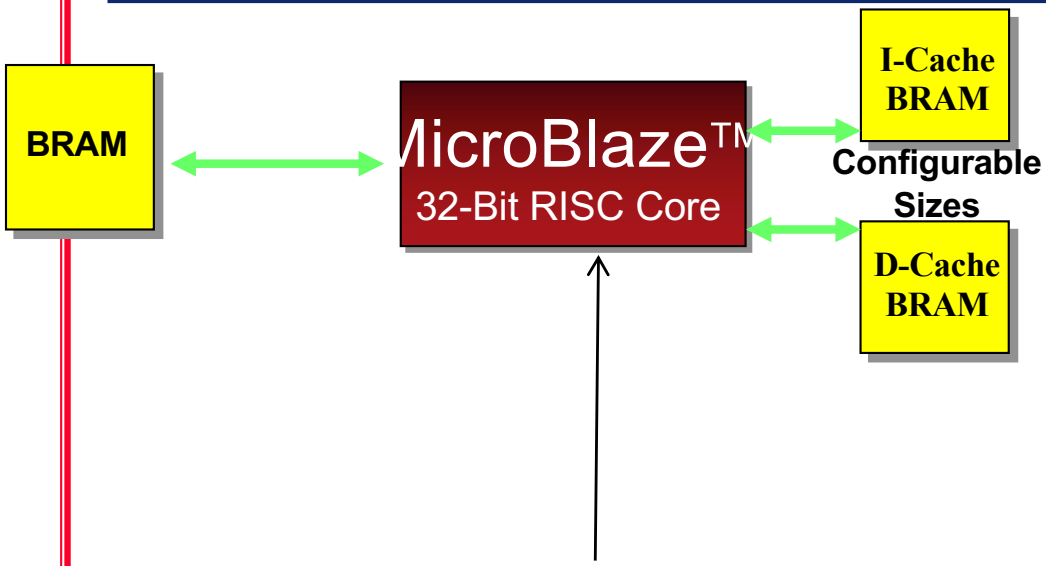
"Generic" Embedded Computer



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

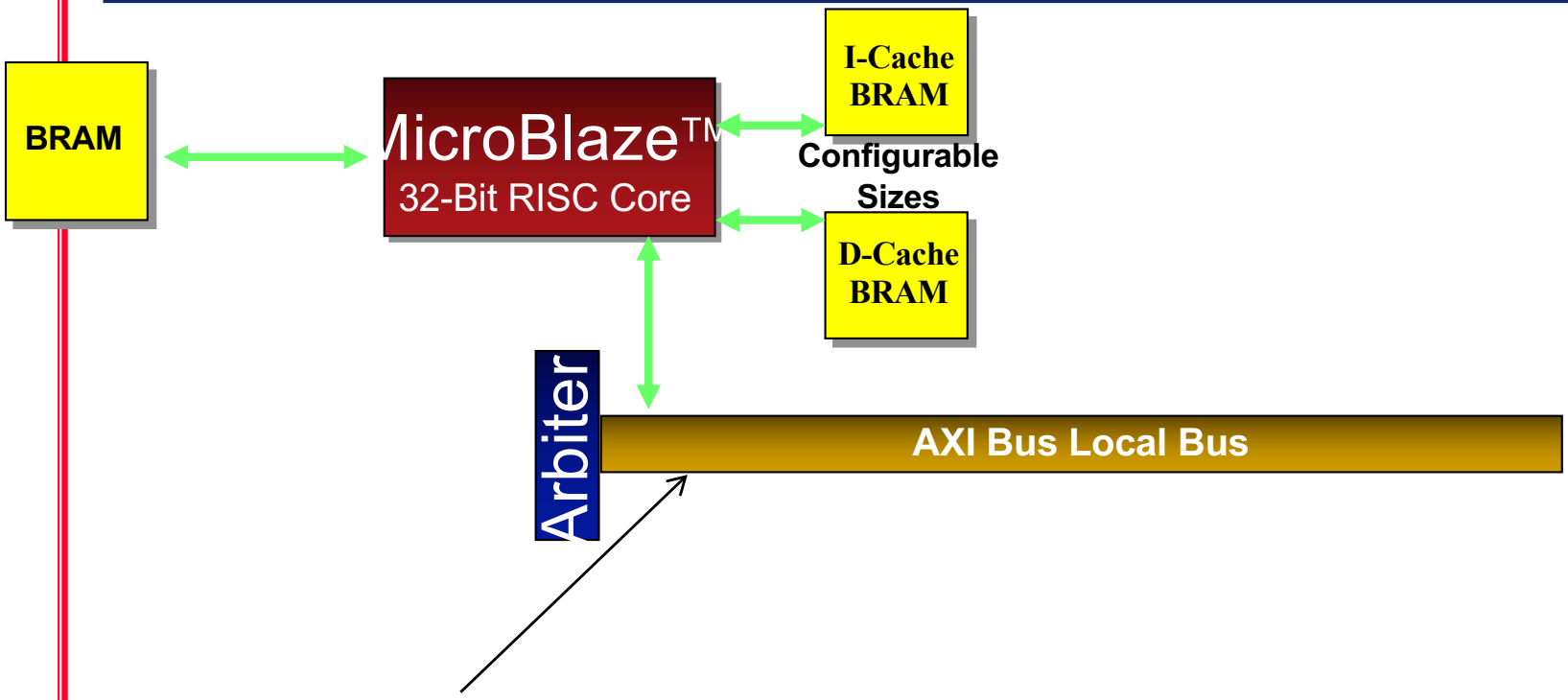
MicroBlaze Processor-Based Embedded Design



We will study architecture+ISA

We will work in both Assembler and C

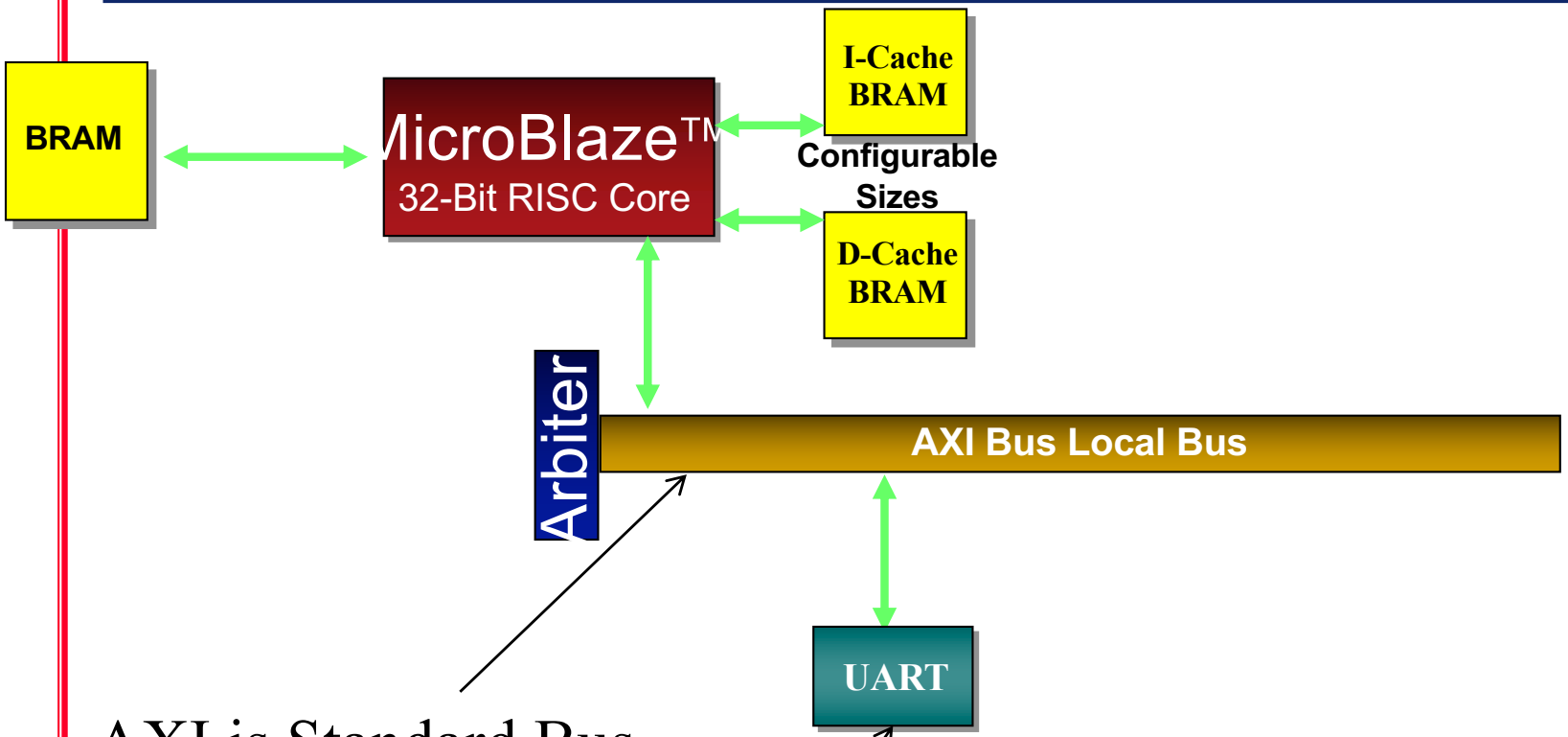
MicroBlaze Processor-Based Embedded Design



AXI is Standard Bus

We will study basic signals

MicroBlaze Processor-Based Embedded Design

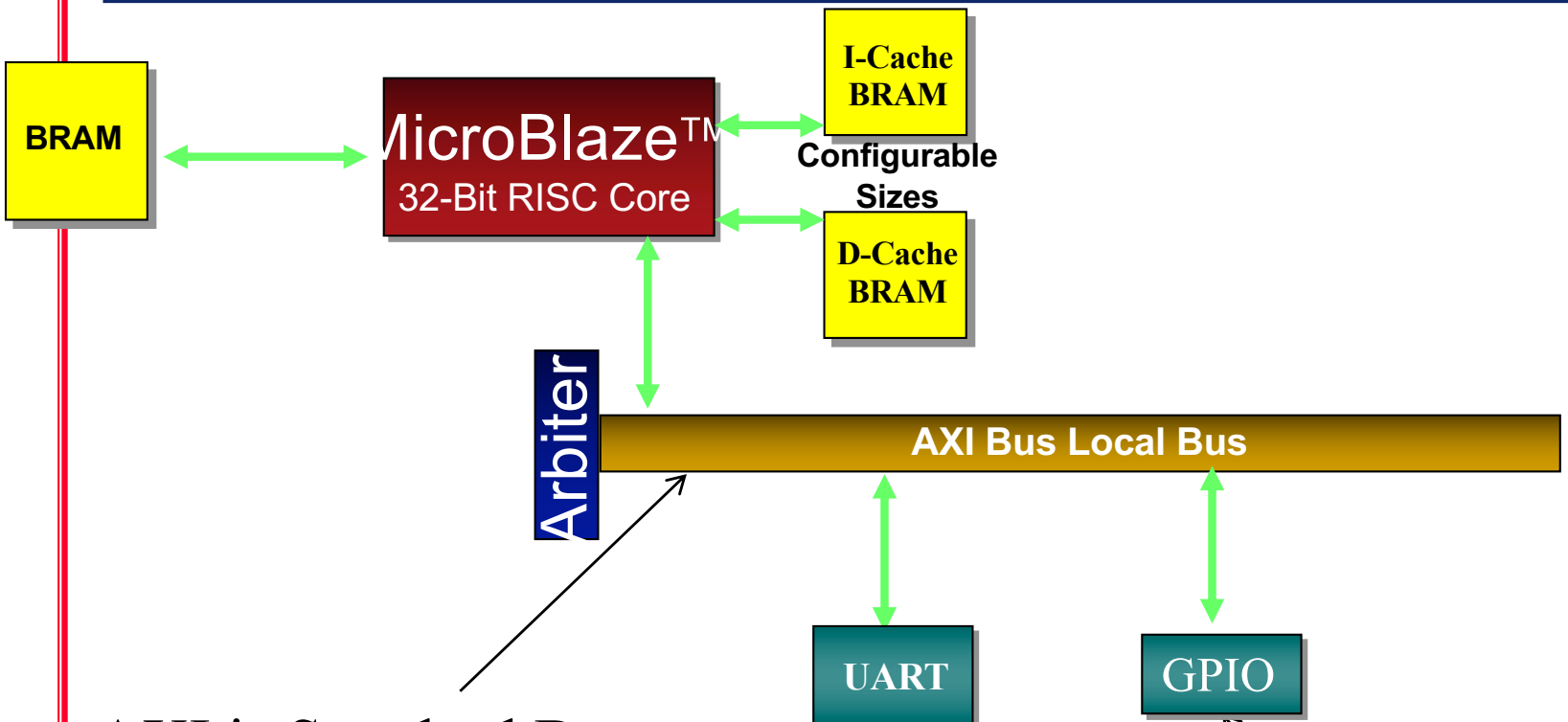


AXI is Standard Bus

We will study basic signals

Serial Port to PC

MicroBlaze Processor-Based Embedded Design

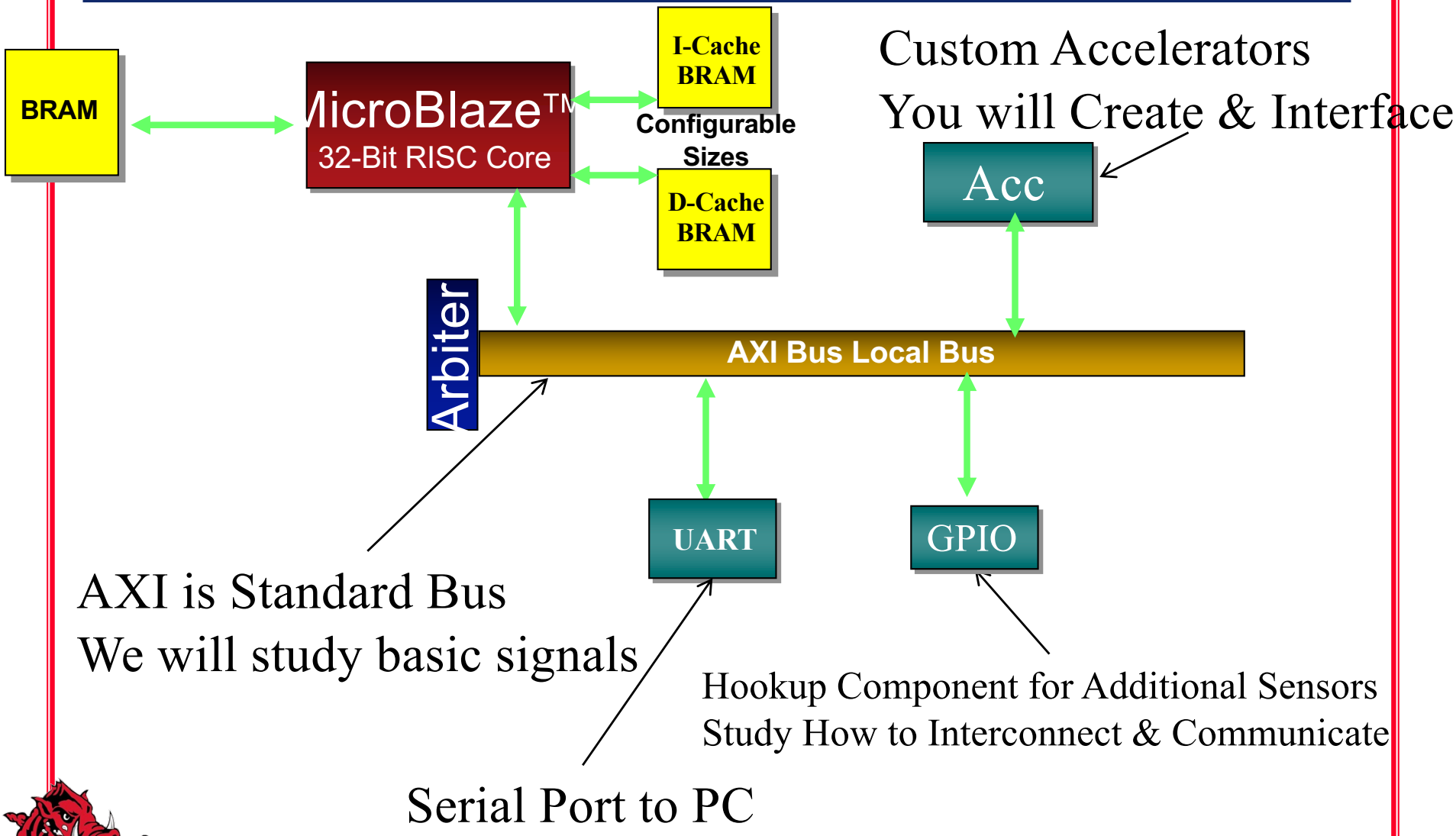


AXI is Standard Bus
We will study basic signals

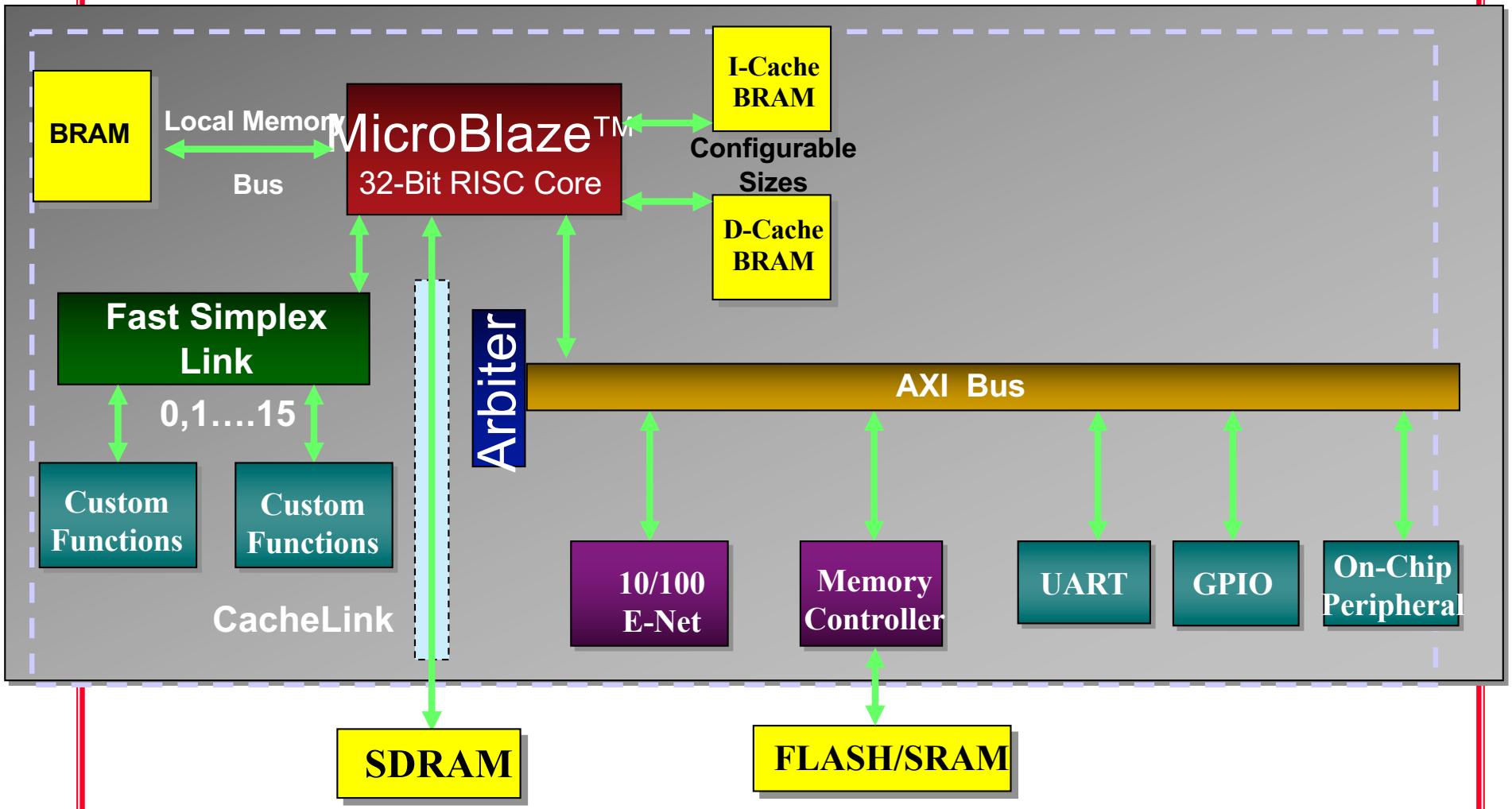
Hookup Component for Additional Sensors
Study How to Interconnect & Communicate

Serial Port to PC

MicroBlaze Processor-Based Embedded Design



System on FPGA Chip (SoC)



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

Summary

- This course will give you appreciation for the fun and difficulty of designing and building an 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.