System-on-Chip Design (SOC)

John Tramel CSULB CECS460 Lecture 1

July 25, 2016

Introduction

* The demand for more powerful products coupled with the capacity of today's silicon technology has brought System-on-Chip (SOC) design into the mainstream
* Design complexity now demands design reuse in order to provide the necessary functionality in a time efficient manner
* In order to accomplish the task of building portions (blocks) and integrating them with other's products demands a very rigorous methodology be practiced
* The supplemental textbook - **Reuse Methodology Manual for System-on-a-chip Designs, by Michael Keating**
    * Refers to itself as a "manual" in the attempt to outline a set of best practices for creating reusable designs for use in an SOC design methodology
    * Industry refers to this text simply as "**RMM**"
* 460 will attempt to cover as many topics as is possible in one semester

System-on-Chip

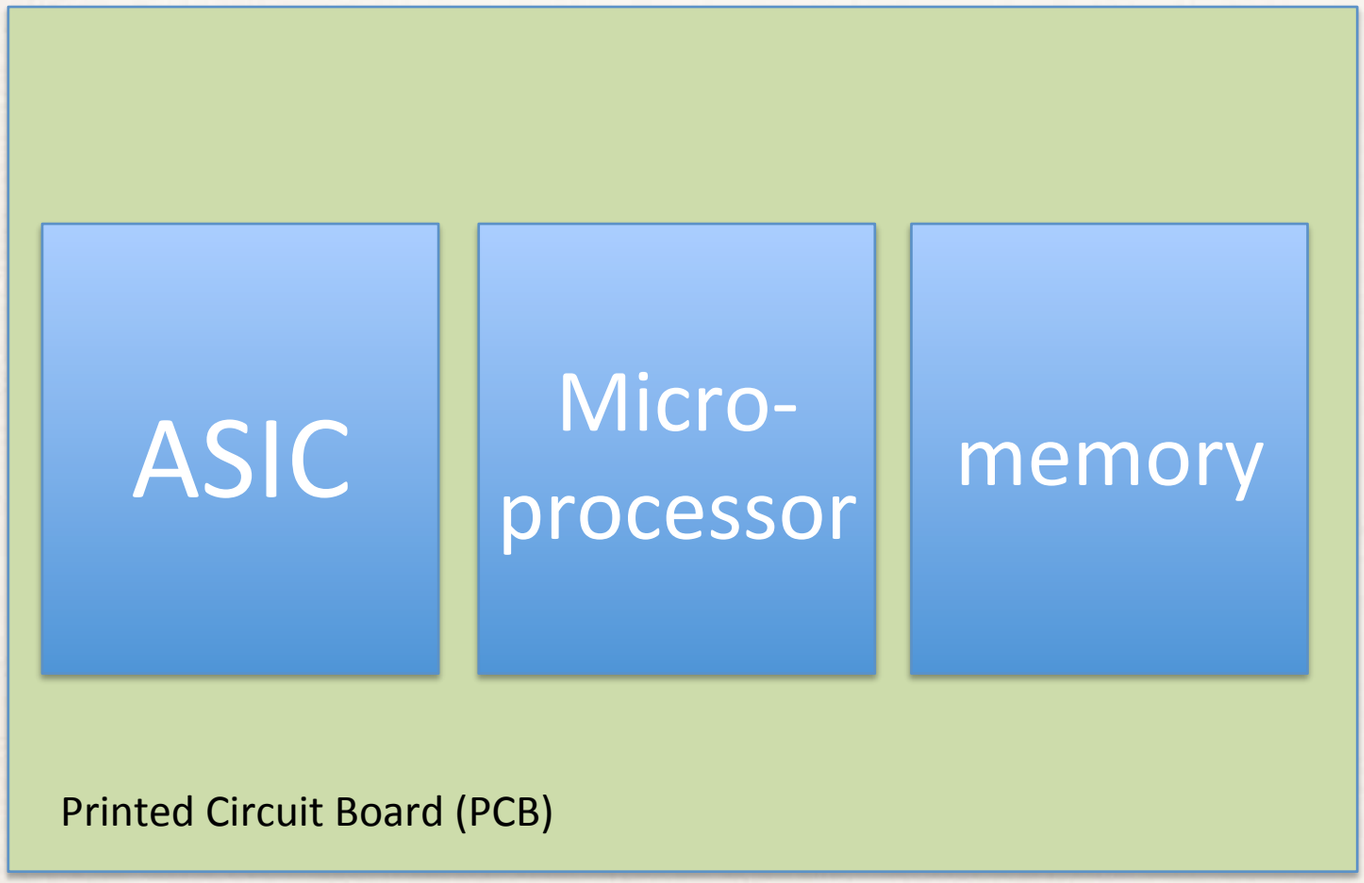
* System-on-Chip (SOC) is an approach where all of the components of a computer or a specific electronic system are integrated into a single integrated circuit (chip)

* SOC takes what previously would have been multiple components on a printed circuit board and combines the “system”into a single chip

* An SOC may contain digital, analog, mixed-signal and occasionally RF functions - all on the same device

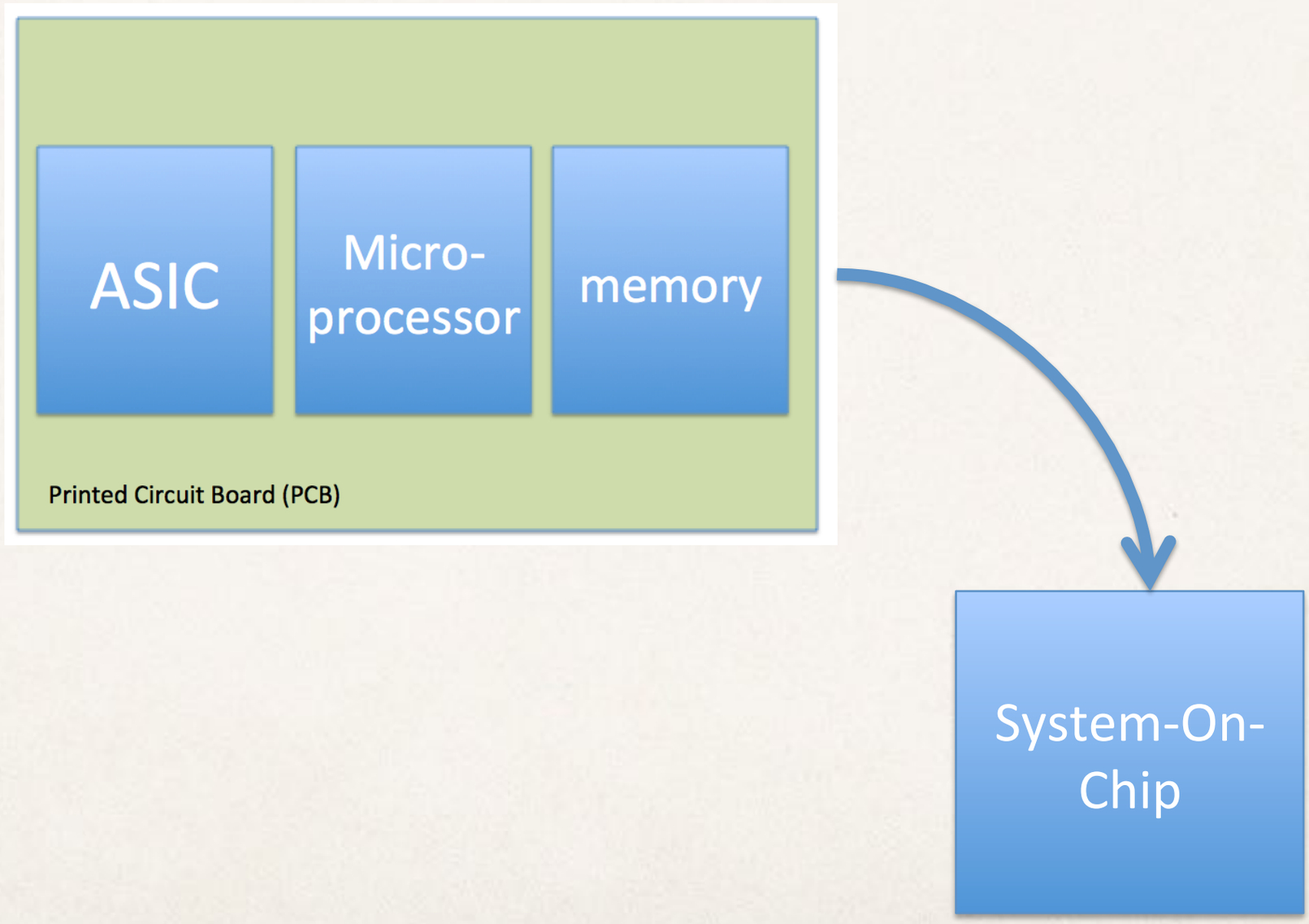
* The common industry term is “Embedded Systems”

SOC Predecessor - ASIC + Processor



* Before processor cores were available as IP designers would design custom ASIC chips and interface them to commercially available processors
* Both would share the same PCB in order to implement the system

System-on-Chip - Visualization



Typical Components of an SOC

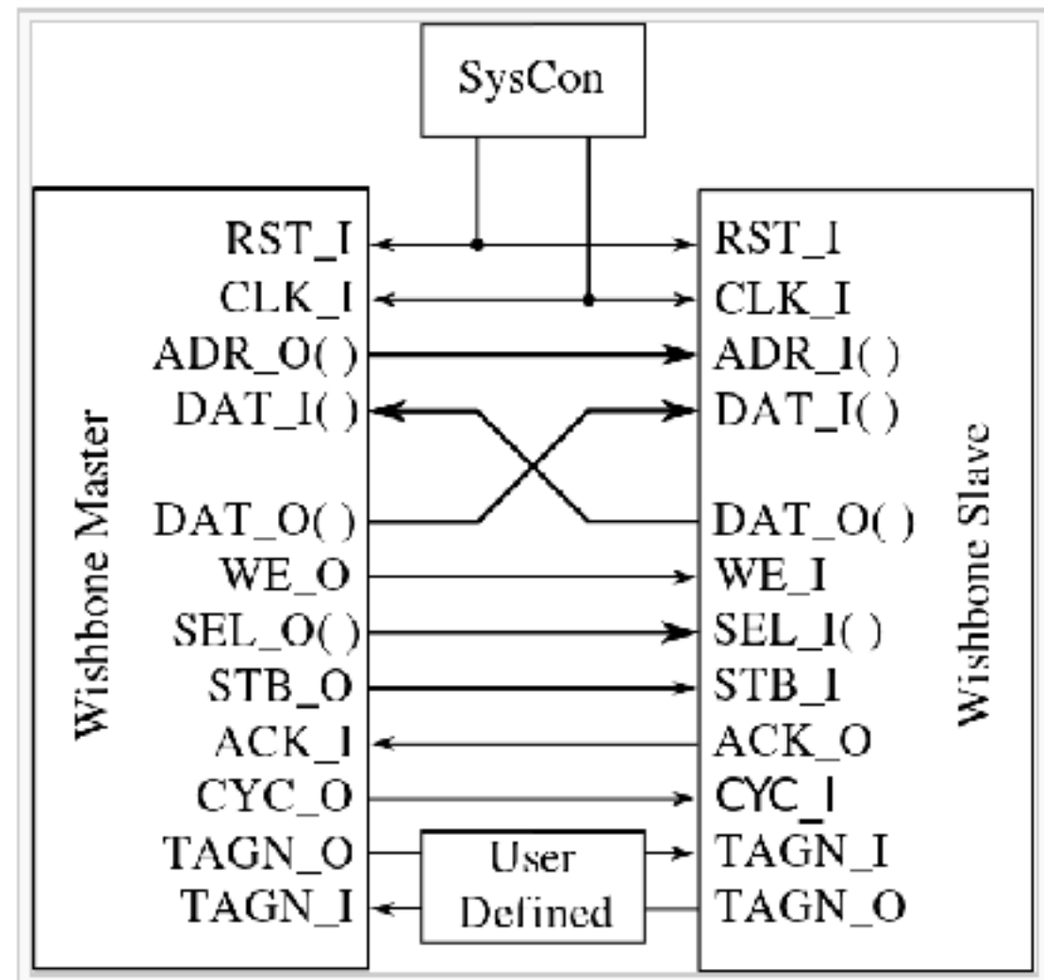
- One or more processor or DSP cores
- Memory blocks including RAM and Flash
- Timing sources including clock generators and phase-locked loops
- Peripheral blocks including counters/timers, power-on reset conditioners,
- External interfaces such as USB, FireWire, I2C, SPI, etc
- Analog interfaces such as ADCs and DACs
- Voltage regulators and power management circuits

SOC Connectivity - AMBA

* SOCs typically have either a proprietary or industry-standard bus to interconnect the processors with the various on-chip components
* Typically ARM-based SOCs utilize the AMBA bus specification
* The Advanced Microcontroller Bus Architecture (AMBA) specification defines and on-chip communications standard for designing high-performance embedded micro controller based systems
* Three distinct buses are defined within the AMBA specification
    * The Advanced High-Performance Bus (AHB)
    * The Advanced System Bus (ASB)
    * The Advanced Peripheral Bus (APB)

SOC Connectivity - Wishbone Bus

* The Wishbone Bus is an open source hardware computer bus intended to let the parts of an integrated circuit communicate with each other
* The aim is to allow the interconnection of differing cores within an SOC design
* The Wishbone Bus is used by many designs in the **OPENCORES** project

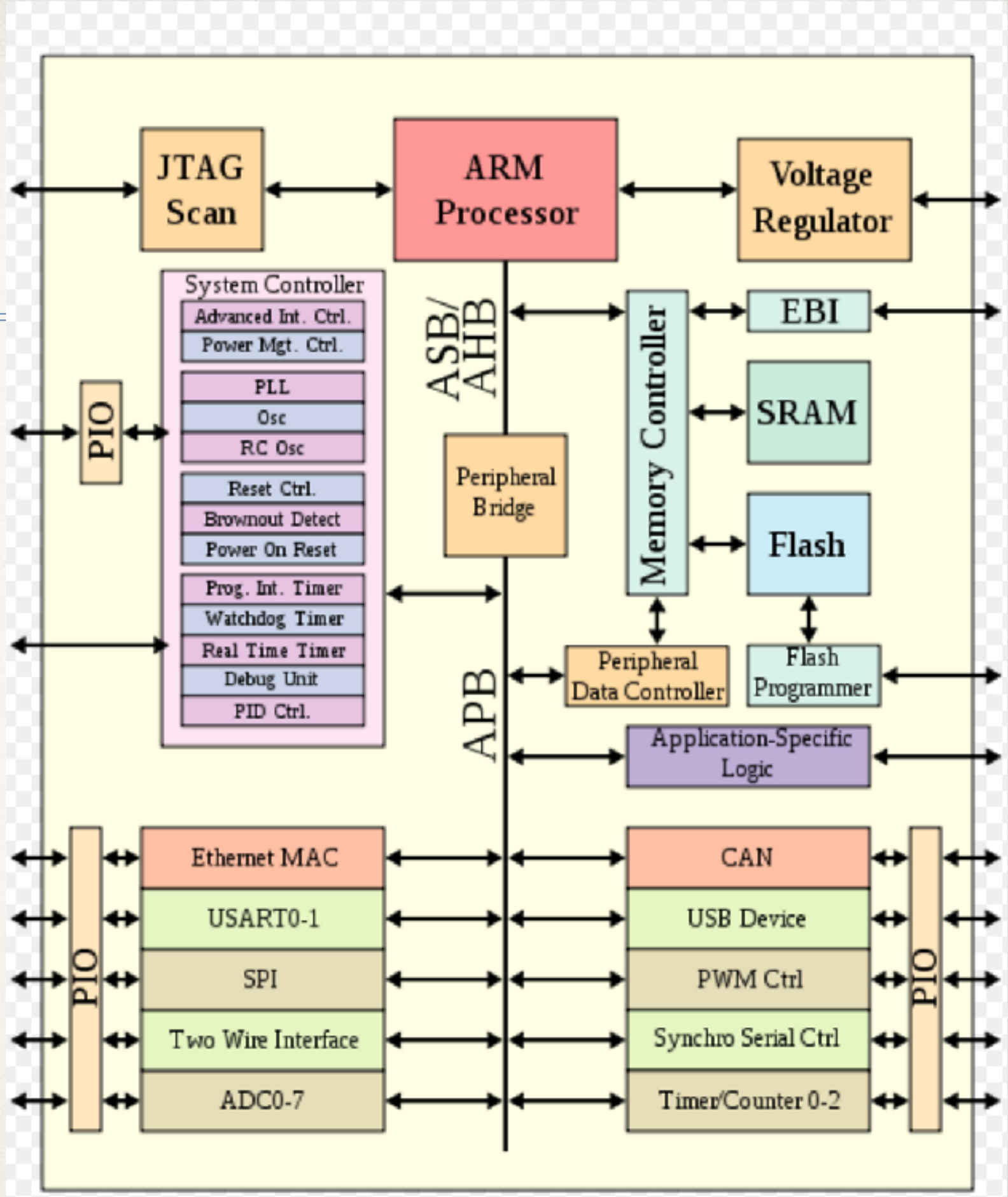


Master and Slave Wishbone's interfaces.

Bus Management

* Order is maintained in the SOC design by designating devices connected to the system bus as either “Masters” or “Slaves”
* Masters may initiate transfers over the bus
* Slaves may only respond to Masters
* Systems that include more than one bus Master require the addition of and additional block, called the “Arbiter”, whose responsibility is to arbitrate who gets the bus when conflicts occur
* DMA is a typical approach implemented in SOCs to free up the Master(s) during data transfers

Typical SOC Block Diagram



Common Set of Problems for SOC Design

- Time-to-market pressures demand rapid development cycles
- Quality of results - performance, area, power [re-spins may take months]
- Increasing chip complexity (large arrays allow increasing large designs)
- Deep sub-micron issues affecting timing closure
- Variety of expertise required to accomplish design
- Design components coming from various sites or vendors with differing methodologies
- SOC processors require software development to proceed hand-in-hand with the chip development

Focus of the Reuse Methodology Manual

Reuse Methodology Manual for System-on-a-Chip Designs by Keating/Bricaud

---

* How reusable macros fit into an SOC development methodology
* How to design reusable soft macros
* How to create reusable hard macros from soft macros
* How to integrate soft and hard macros into an SOC design
* How to verify functionality and timing in large SOC designs

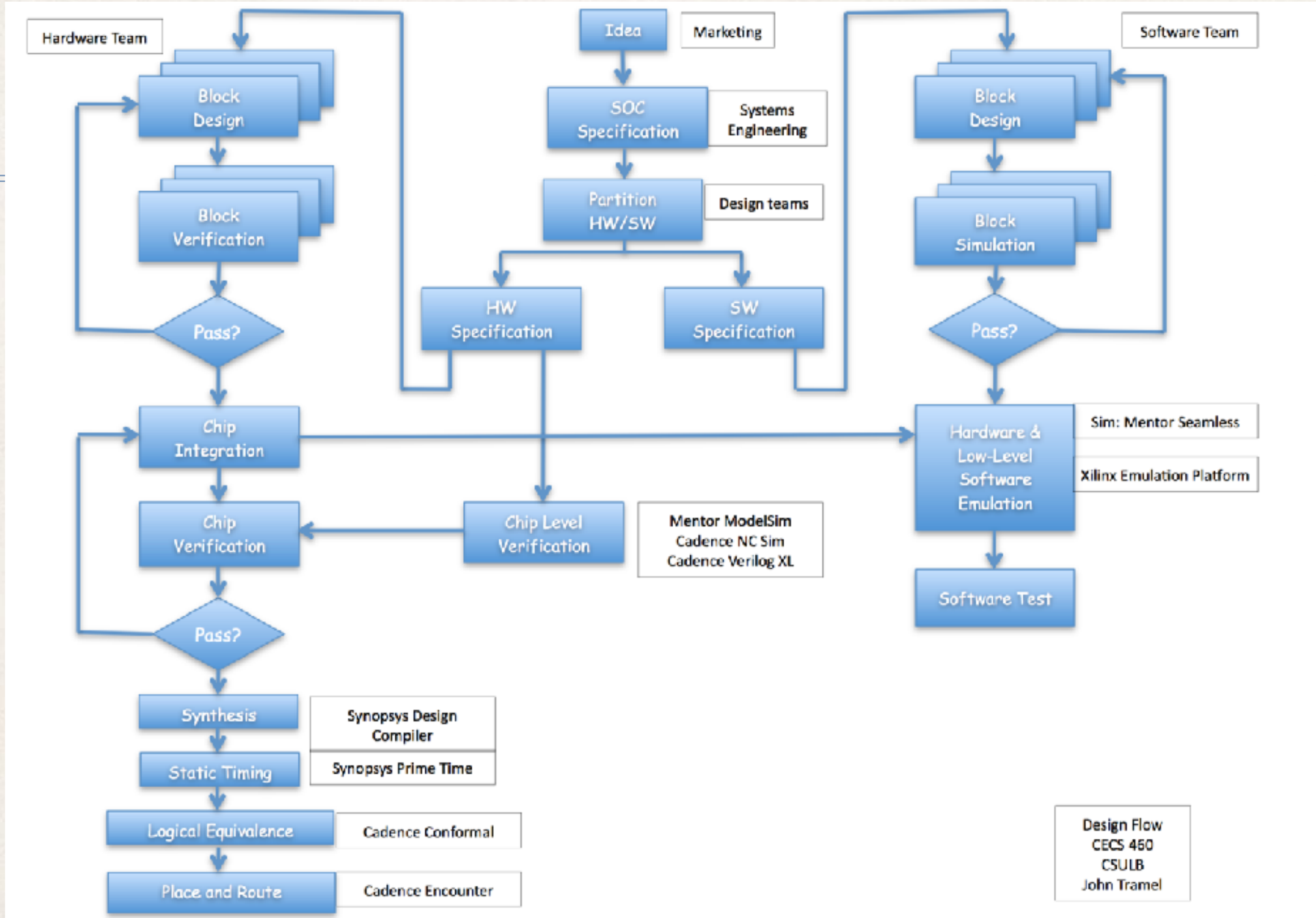
Focus of CECS 460

- Due to obvious limitations we can not accomplish an actual SOC design with hard and soft macros - but we will be able to accomplish an actual SOPC design (more later)
- We will discuss the concepts involved in the development flow along with the industry tools required to accomplish each step
- We will be developing an SOPC throughout the course so the topics discussed in lecture may be practiced in the labs
- We will focus on block development for reuse
- We will focus on design integration to produce an SOPC using the Digilent family of of Xilinx-based development boards
- We will integrate a simple processor into our design and write the necessary control software

Focus of CECS 460 - Continued

- Terminology
  - ASIC - Application Specific Integrated Circuit
  - SOC - System-on-chip created by the inclusion of an embedded processor to an ASIC design
  - SOPC - System-on-chip design targeting FPGAs rather than costly/time consuming ASICs
    - Subblock - A subcomponent of a core, block, or IP, that is too small to be a stand-alone design component
    - Soft macro - Core, block or IP that is delivered to the integrator as synthesizable RTL code
    - Hard macro - Core, block or IP that is delivered to the integrator as a GDSII file: fully designed, placed and routed, and verified in the target technology
- Design for reuse goals
  - Good coding style with adequate comments
  - Good documentation
  - Well-designed verification environments and suites

SOC Flow


Hardware Team
Block Design
Block Verification
Pass?
Chip Integration
Chip Verification
Pass?
Synthesis
Static Timing
Logical Equivalence
Place and Route
Cadence Conformal
Cadence Encounter
Idea
Marketing
SOC Specification
Systems Engineering
Partition HW/SW
Design teams
HW Specification
SW Specification
Chip Level Verification
Mentor ModelSim
Cadence NC Sim
Cadence Verilog XL
Synopsys Design Compiler
Synopsys Prime Time
Software Team
Block Design
Block Simulation
Pass?
Hardware & Low-Level Software Emulation
Sim: Mentor Seamless
Xilinx Emulation Platform
Software Test
Design Flow
CECS 460
CSULB
John Tramel

Comments on the SOC Flow

* This flow is applicable to three similar but distinct design activities: 1) ASIC design, 2) SOC design (ASIC + processor), 3) SOPC design (FPGA + processor). What is similar is that the end result of each activity is a digital design functioning within a single device
* While an SOC/ASIC may be considered “custom circuits” the FPGA will implement a “custom design” in a commercially available programmable device
* Consider the ASIC flow as a subset of the SOC flow in that the SOC flow includes an embedded processor with the required software
* The complexity of today’s FPGAs have mandated that the methodology once used for ASIC design be applied also to SOPC designs
* SOC flow should enable design reuse and allow for rapid adaptation in order to meet customer’s or industry’s quickly changing appetites