EE382N-4
Embedded Systems Architecture Course

Mark McDermott

Fall 2009

Goals of the Course

High-Level Goals

- Understand the scientific principles and concepts behind embedded systems, and
- Obtain hands-on experience in programming embedded systems.

By the end of the course, you will be able to

- Understand the “Big Ideas" in embedded systems
- Obtain direct hands-on experience on both hardware and software elements commonly used in embedded system design.
- Understand basic real-time resource management theory
- Understand the basics of embedded system application concepts such as signal processing and feedback control
- Understand, and be able to discuss and communicate intelligently about
  - embedded processor architecture and programming
  - I/O and device driver interfaces to embedded processors with networks, multimedia cards and disk drives
  - OS primitives for concurrency, timeouts, scheduling, communication and synchronization

What are the "Big Ideas"

- **HW/SW Architecture**
  - Non processor centric view of architecture
- **Bowels of the "operating system"**
  - Specifically, the lower half of the OS
  - Concurrency
- **Real world design**
  - performance vs. cost tradeoffs
- **Analyzability**
  - how do you "know" that your drive-by-wire system will function correctly?
- **Application-level techniques**
  - Power Aware Programming

So, what is an embedded system?

Simple answer -

Anything that uses a "processor" but isn't a general-purpose computer.

The user "sees" a smart (special-purpose) system as opposed to the computer inside the system

"how does it do that?"

"it has a computer inside it!"

"it does not or cannot run Windows or MacOS!"

But it might run Windows CE or Linux…

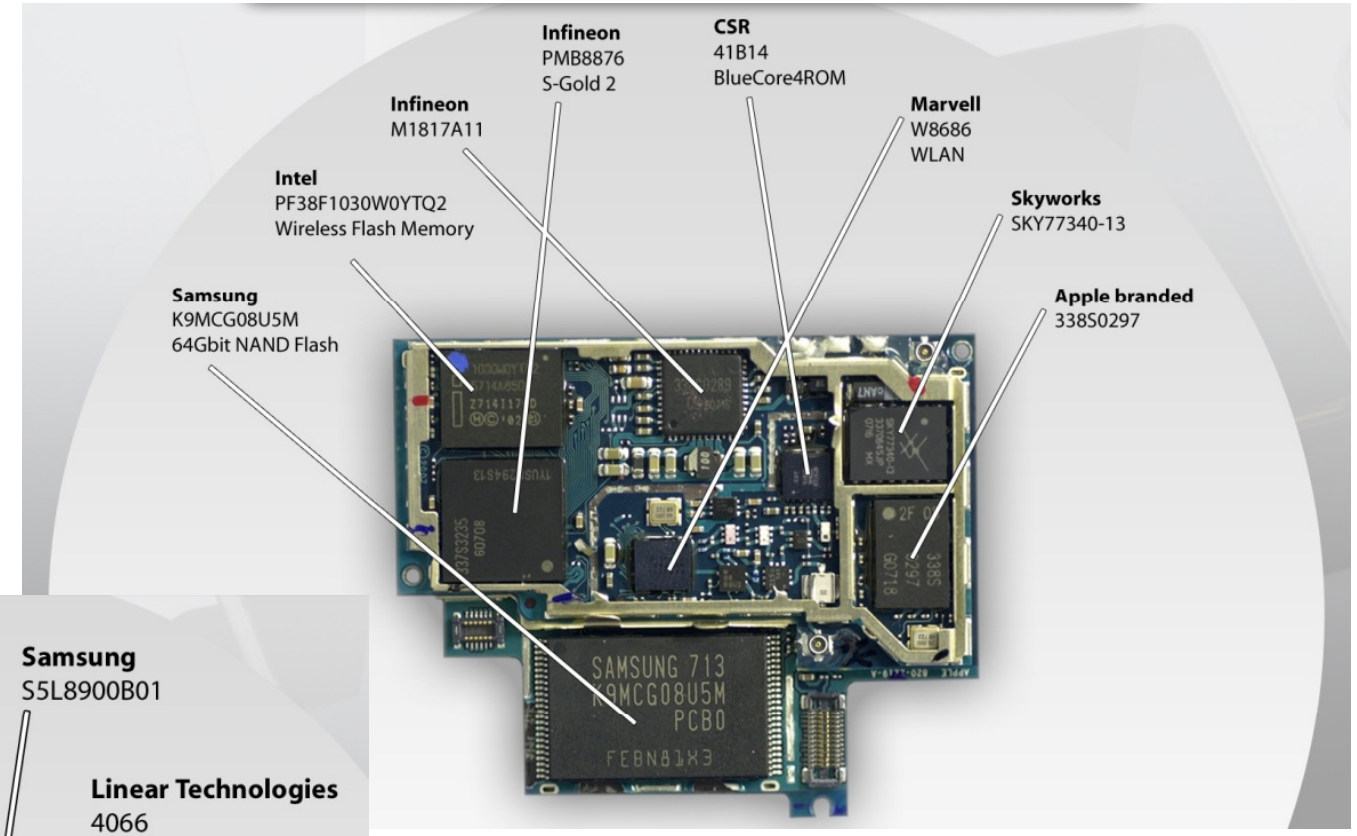
The end-user typically does not or cannot modify or upgrade the internals

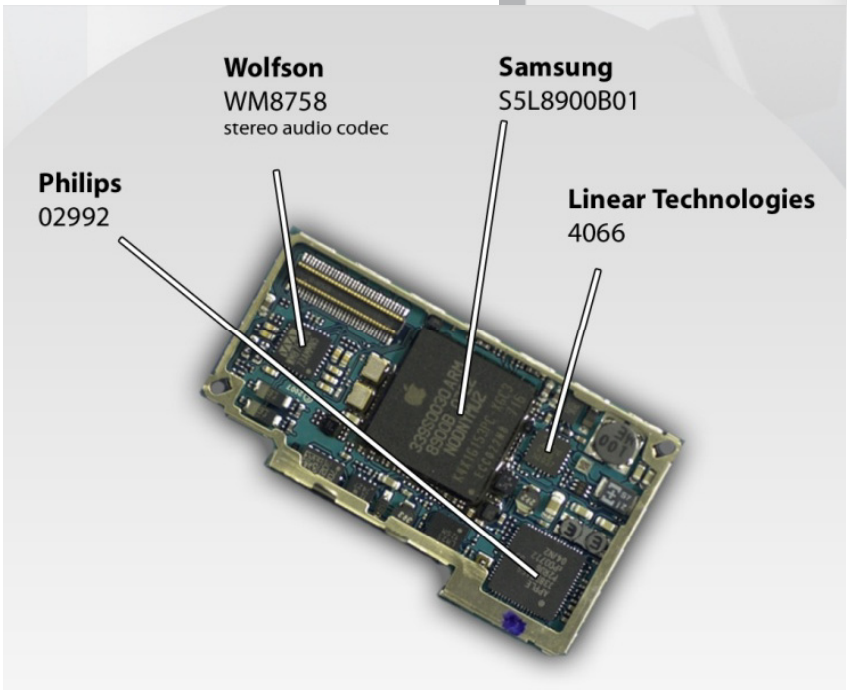
Courtesy Intel

Prototypical example of an embedded system: iPhone

- **Contains:**
  - Applications processor (CPU): Samsung/ARM S5L8900B01 512 Mbit SRAM
  - Flash Memory: Samsung 65-nm 8/16 GB (K9MCG08U5M)
  - I/O controller: Broadcom BCM5973A
  - Touchscreen controller: Philips LPC2221/02992
  - Audio processor: Wolfson WM8758
  - Wireless LAN processor: Marvell 90-nm 88W8686
  - Baseband processor: Infineon PMB8876 S-Gold 2 EDGE
  - RF Transceiver: Infineon M1817A11 GSM
  - Flash Memory: 4 GB (K9HBG08U1M) MLC NAND Flash
  - Amplifier: Skyworks SKY77340-13 Signal Amplifier
  - Flash Memory: Intel PF38F1030W0YTQ2 (32 MB NOR + 16 MB SRAM)
  - Bluetooth: CSR BlueCore 4

Apple iPhone





Four General Categories of Embedded Systems

- **General Computing**
  - Applications similar to desktop computing, but in an embedded package
  - Video games, set top boxes, wearable computers, automatic tellers
- **Control Systems**
  - Closed loop feedback control of real time system
  - Vehicle engines, chemical processes, nuclear power, flight control
- **Signal Processing**
  - Computations involving large data stream
  - Radar, Sonar, video compression
- **Communication & Networking**
  - Switching and information transmission
  - Telephone system, Internet

Courtesy Intel

Types of Embedded System Functions

- **Control Laws**
  - PID control
  - Fuzzy logic, ...
- **Sequencing logic**
  - Finite state machines
  - Switching modes between control laws
- **Signal processing**
  - Multimedia data compression
  - Digital filtering
- **Application specific interfacing**
  - Buttons, bells, lights,...
  - High speed I/O
- **Fault response**
  - Detection & reconfiguration
  - Diagnosis

Courtesy Intel

Distinctive Embedded System Attributes

- **Reactive: computations occur in response to external events**
  - Periodic events (e.g., rotating machinery and control loops)
  - Aperiodic events (e.g., button closures)
- **Real-Time: timing correctness is part of system correctness**
  - Hard real-time
    - Absolute deadline, beyond which answer is useless
    - May include minimum time as well as maximum time
  - Soft real-time
    - Missing a deadline is not catastrophic
    - Utility of answer degrades with time difference from deadline
  - Example:
    - a train is entering an urban area...
    - the railway gate in the city allows automotive traffic to go over the tracks
    - when should the railway gate close?

In general,

Real Time does not equal "Real Fast"

Courtesy Intel

Typical Embedded System Constraints

- **Small Size, Low Weight**
  - Handheld electronics
  - Transportation applications weight costs money
- **Low Power**
  - Battery power for 8+ hours (laptops often last only 2 hours)
  - Limited cooling may limit power even if AC power available
- **Harsh environment**
  - Heat, vibration, shock
  - Power fluctuations, RF interference, lightning
  - Water, corrosion, physical abuse
- **Safety critical operation**
  - Must function correctly
  - Must not function incorrectly
- **Extreme cost sensitivity**
  - $.05 adds up over 1,000,000 units

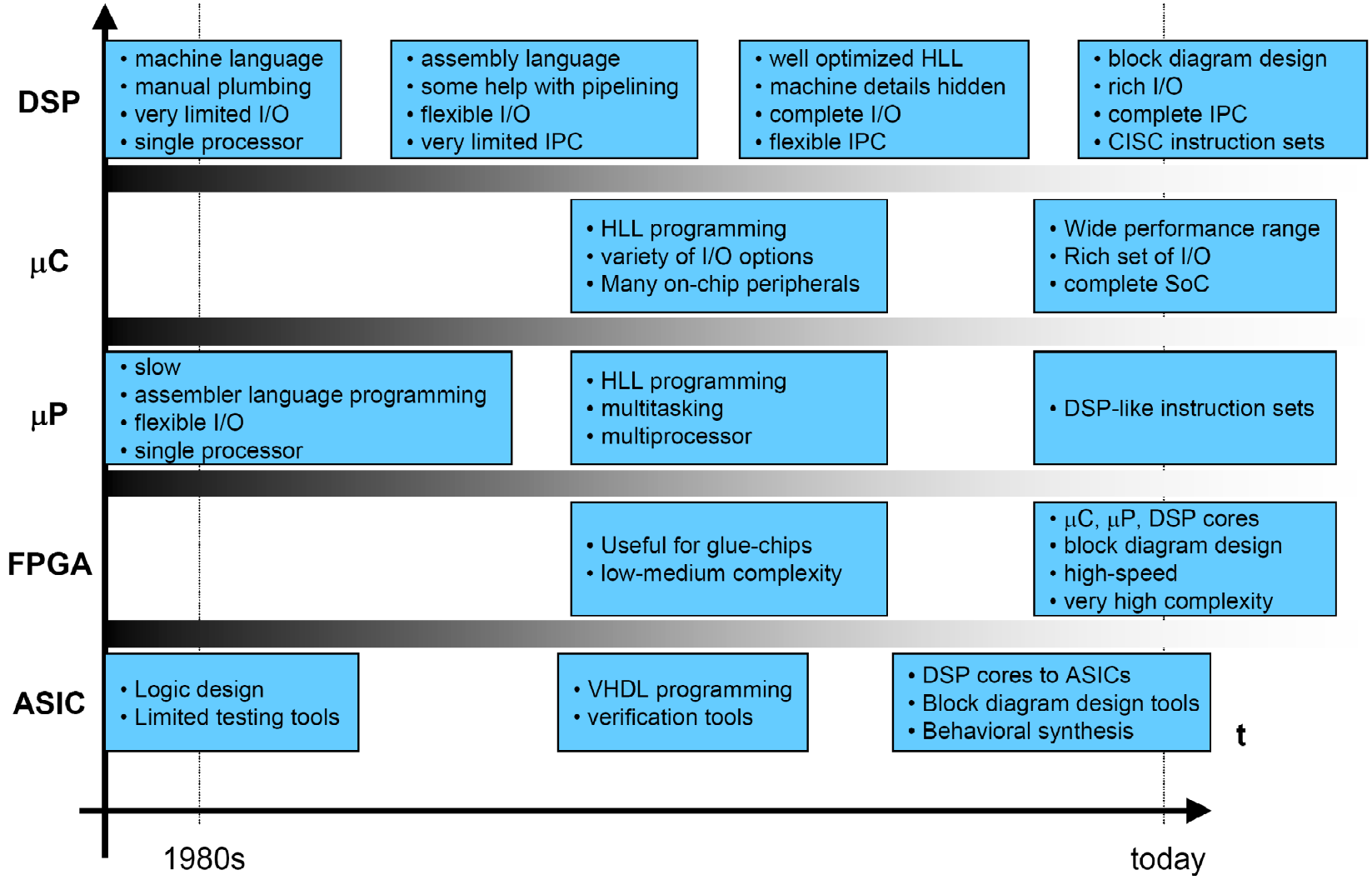
Courtesy Intel

A Hierarchy of Embedded Computer Design

	Level	Name	Modules	Primitives	Descriptive Media
HARDWARE	1	Electronics	Gates, FF's	Transistors, Resistors, etc.	Circuit Diagrams
	2	Logic	Registers, ALU's	Gates, FF's	Logic Diagrams
	3	Organization	Processors, Busses, Memories	Registers, ALU's ...	Register Transfer Notation
FIRMWARE	4	Microprogramming	Assembly Language	Microinstructions	Microprogram
SOFTWARE	5	Assembly Language Programming	OS Routines	Assembly Language Instructions	Assembly Language Programs
	6	Procedural Programming	Applications, Drivers, etc.	OS Routines High Level Languages	HLL Programs
	7	Embedded Applications	Embedded Systems	Procedural Constructs	Problem Oriented Programs

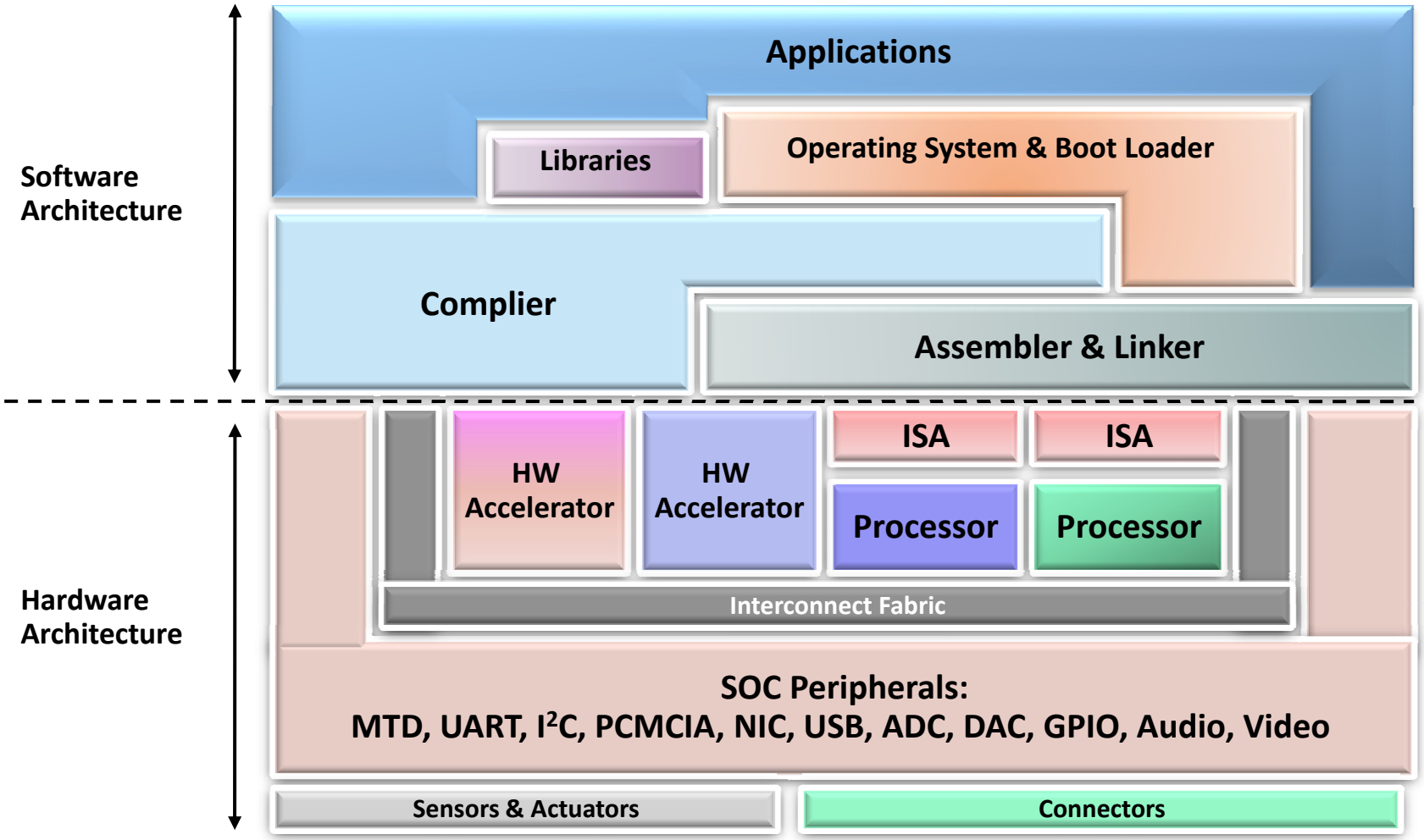
Focus of this class

Embedded Computer Design Evolution

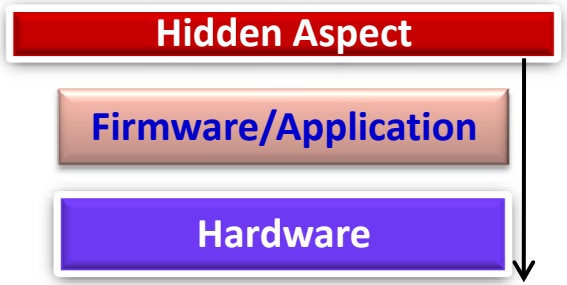


bmcnair@stevens.edu

A Typical Embedded System

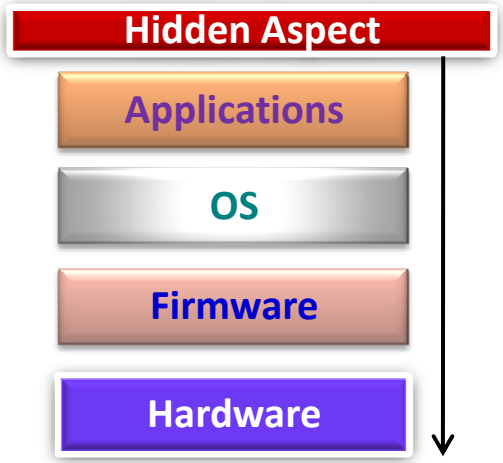


Types of System Abstraction



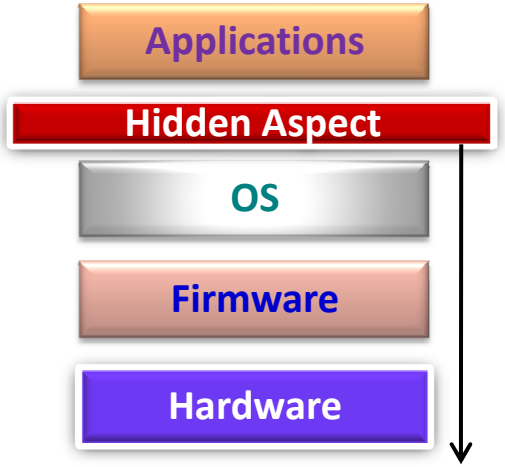
Basic Embedded System

Example:
PID controller



Complex Embedded System

Example:
Set top Box

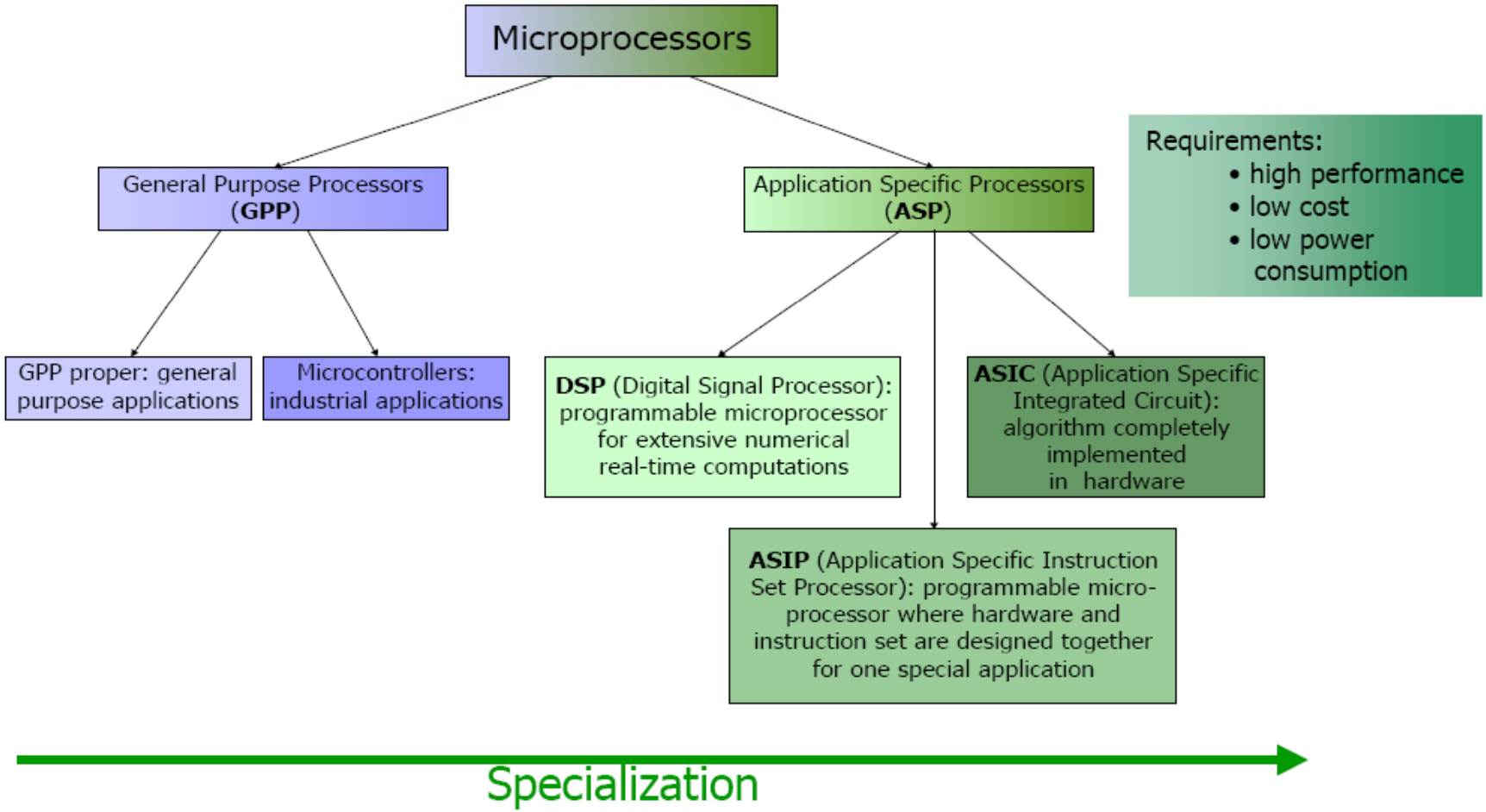


Standalone System

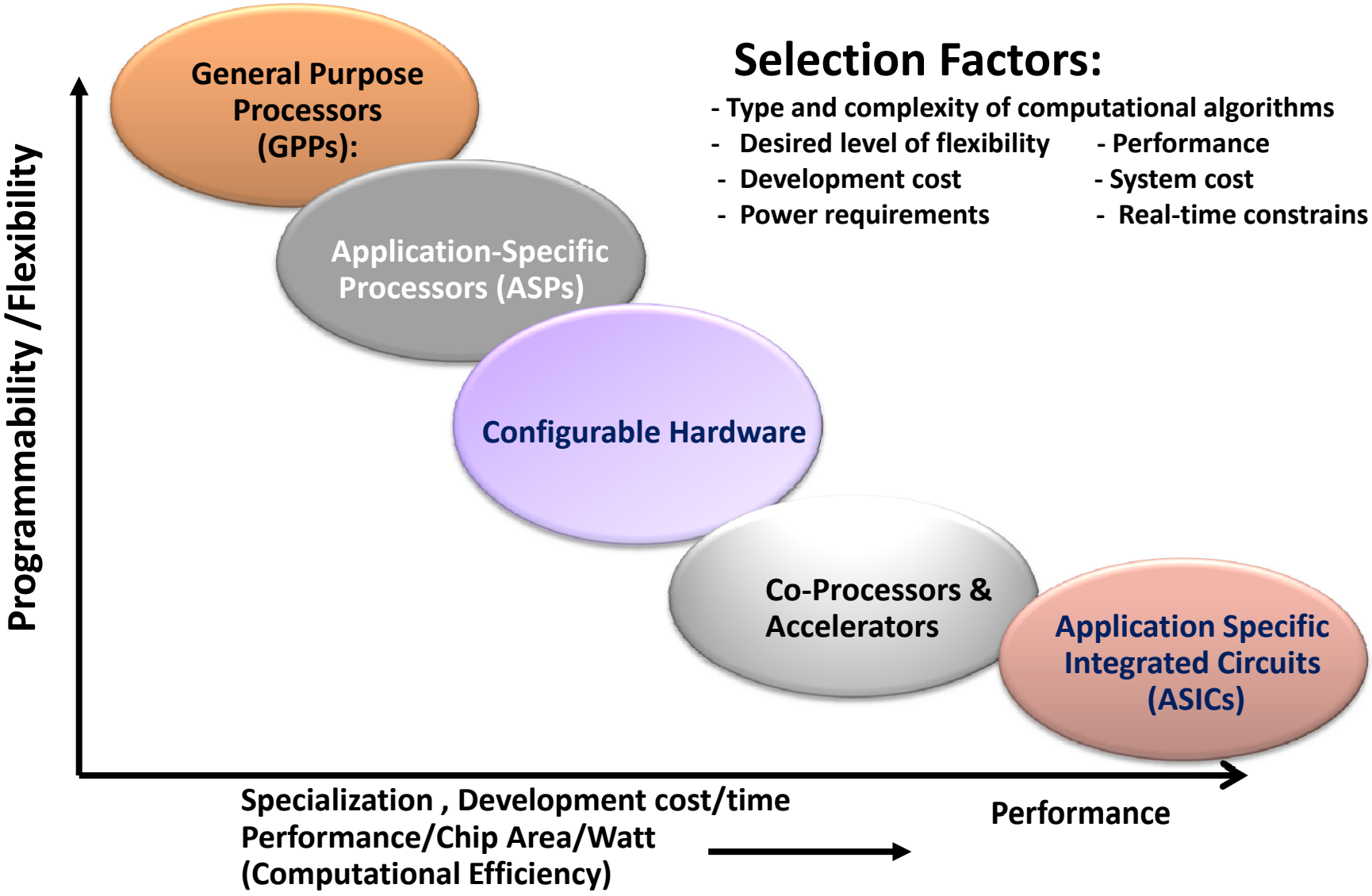
Example:
iPhone

bmcnair@stevens.edu

Processor Classification



Computing Element Choices

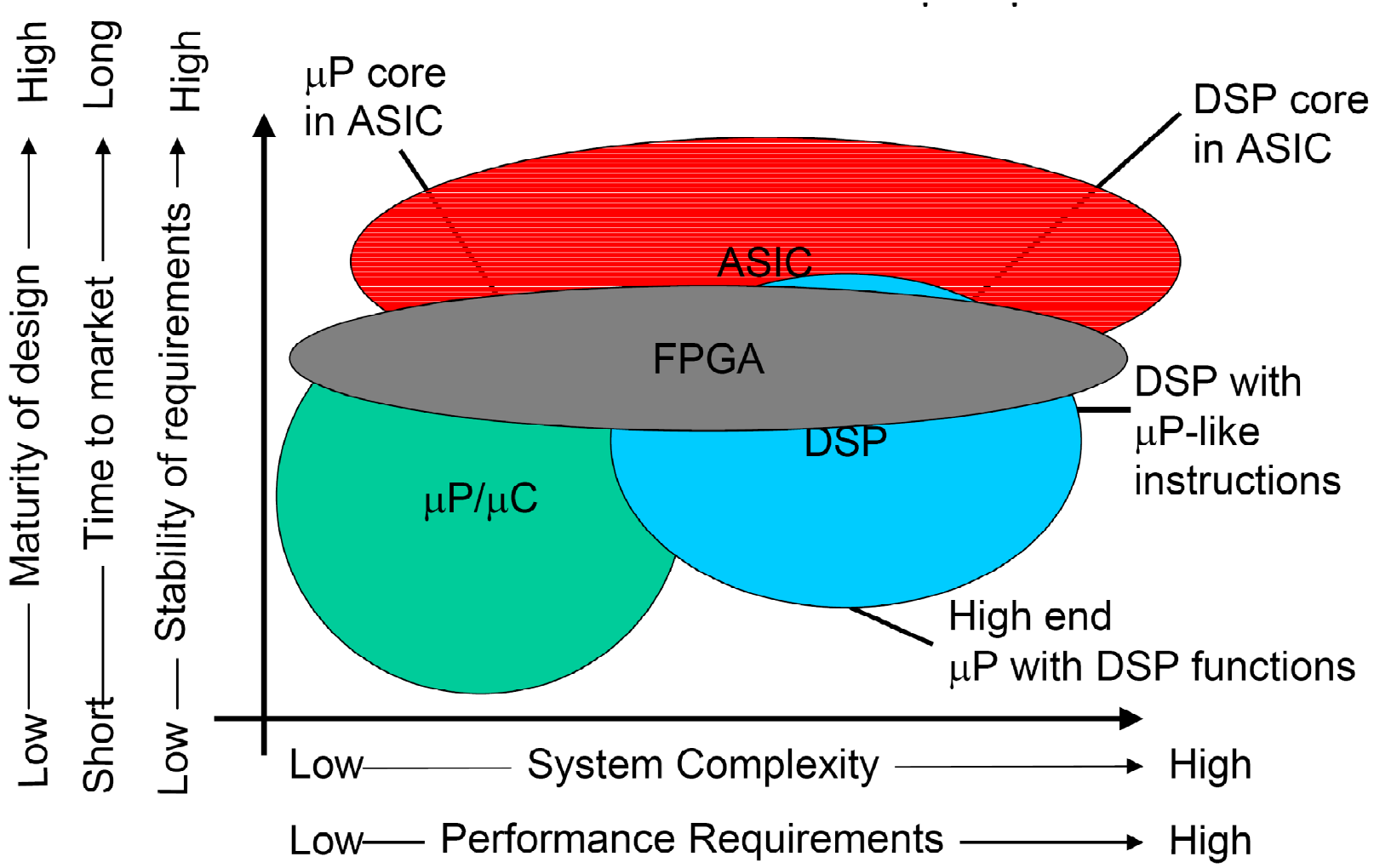


Selection Factors:

- Type and complexity of computational algorithms
- Desired level of flexibility
- Performance
- Development cost
- System cost
- Power requirements
- Real-time constrains

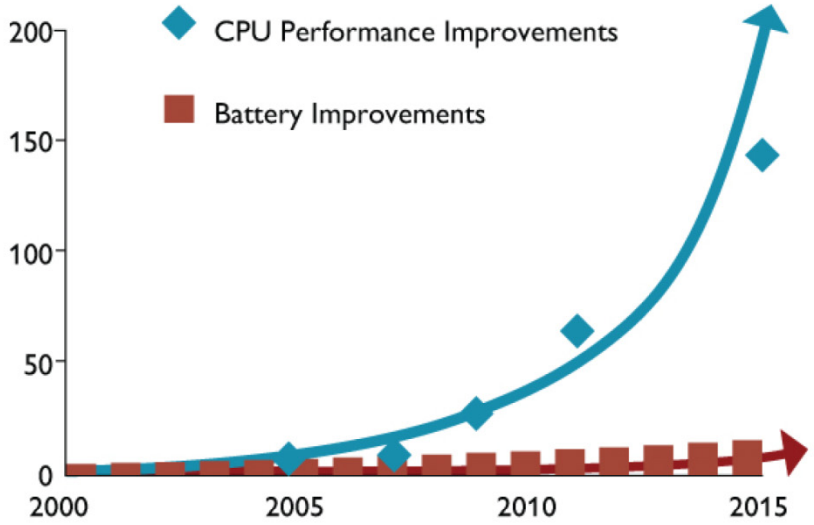
Courtesy Intel

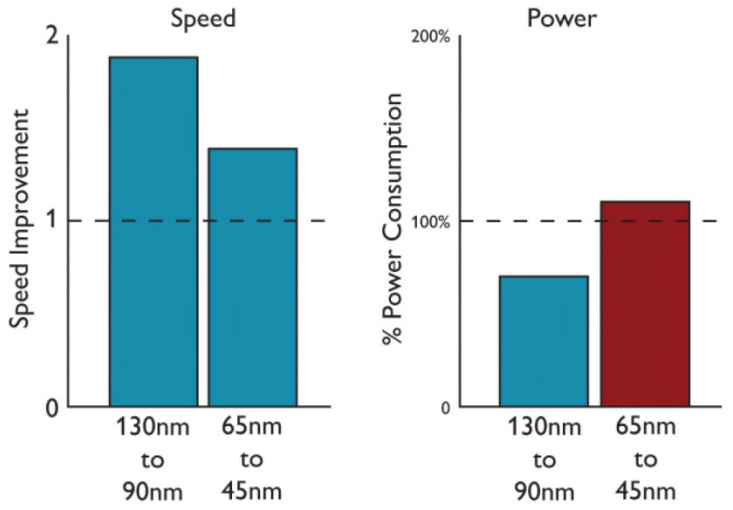
DSP vs. ASIC vs. FPGA vs. µP/ µC



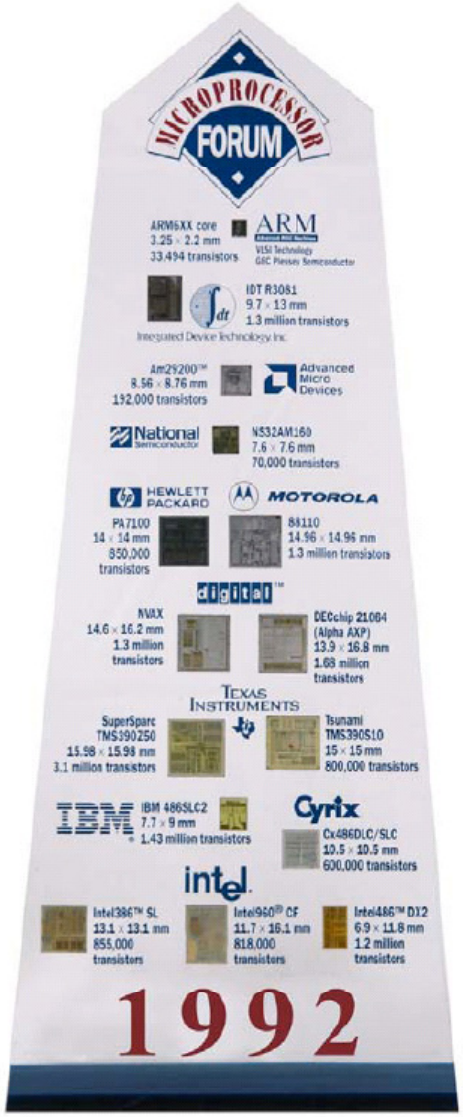
bmcnair@stevens.edu

The Real Problem facing Embedded Systems: POWER



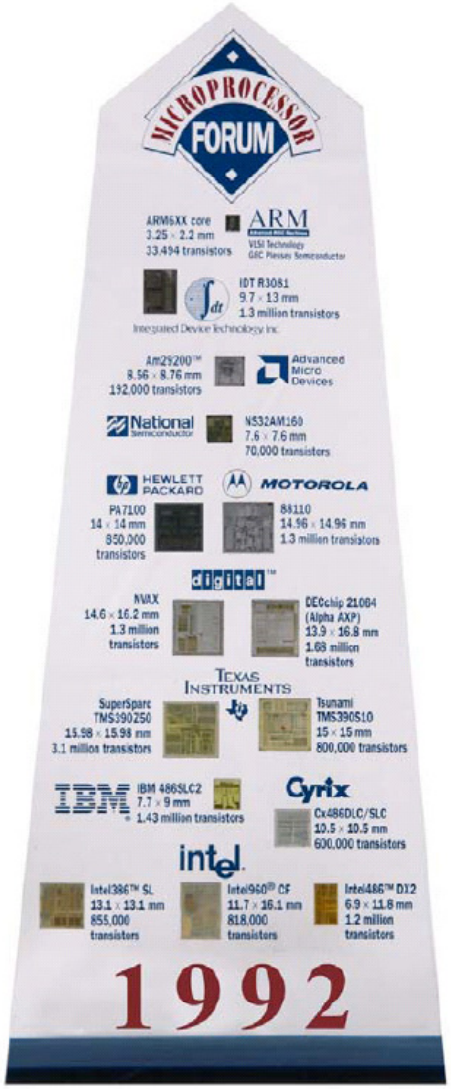


Why ARM?


MICROPROCESSOR FORUM
ARM6XX core
3.25 × 2.2 mm
33,494 transistors
ARM
IDT R3081
9.7 × 13 mm
1.3 million transistors
Integrated Device Technology Inc.
Am29200™
8.56 × 8.76 mm
132,000 transistors
Advanced Micro Devices
National
N532AM160
7.6 × 7.6 mm
70,000 transistors
HEWLETT PACKARD
MOTOROLA
PA7100
14 × 14 mm
850,000 transistors
88110
14.96 × 14.96 mm
1.3 million transistors
digital™
NVAX
14.6 × 16.2 mm
1.3 million transistors
DECchip 21064 (Alpha AXP)
13.9 × 16.8 mm
1.68 million transistors
TEXAS INSTRUMENTS
SuperSparc TMS390Z50
15.98 × 15.98 mm
3.1 million transistors
Tsunami TMS390S10
15 × 15 mm
800,000 transistors
IBM
IBM 486SLC2
7.7 × 9 mm
1.43 million transistors
Cyrix
Cx486DLC/SLC
10.5 × 10.5 mm
600,000 transistors
intel
Intel386™ SL
13.1 × 13.1 mm
855,000 transistors
Intel960® CF
11.7 × 16.1 mm
818,000 transistors
Intel486™ DX2
6.9 × 11.8 mm
1.2 million transistors
1992


MICROPROCESSOR FORUM
ARM
MIPS
29K
N32x16
PA
88xxx
NVAX
Alpha
SPARC
x86
x86
x86
i960
x86
1992

Because it is the "other" architecture



Course Syllabus

Hardware	Software
ISAs for Embedded Systems	Monitors & Bootloaders
GPP Microarchitectures	Embedded Linux
ARM ISA	Device Driver Development
FPGA Architectures	Interrupt Handlers
DSP Microarchitectures	Debugging Embedded Systems
Blackfin ISA	Real Time Operating Systems
Reconfigurable Logic	OS Services and Middleware
I/O Subsystems	SW Library Development
Networks-on-Chips	File System Development
Intelligent Sensors	SW Optimization
SOC Architecture	Testing Embedded Systems

Grading

- Homework: 10%
- Labs: 30%
- Exam 1: 15%
- Exam 2: 15%
- Project: 30%

Penalty for late submission of homework and class project: 25% per working day. (Maximum: 100%).

Overview of the Lab Assignments

- **Lab #1**
  - Write an ARM assembly language program (ALP) implementation of a memory test. Use the ARM ADS system to verify.
- **Lab #2**
  - Develop an Interrupt handler to measure the latency of Linux running on the TLL2020 platform using the I/O ports on the FPGA.
- **Lab #3**
  - Write an ALP and/or C-routine which slows down and speeds up the processor clock on the ARM processor without disrupting the I/O capability. The processor will be performing a memory tests on the DRAM that is controlled by the FPGA.

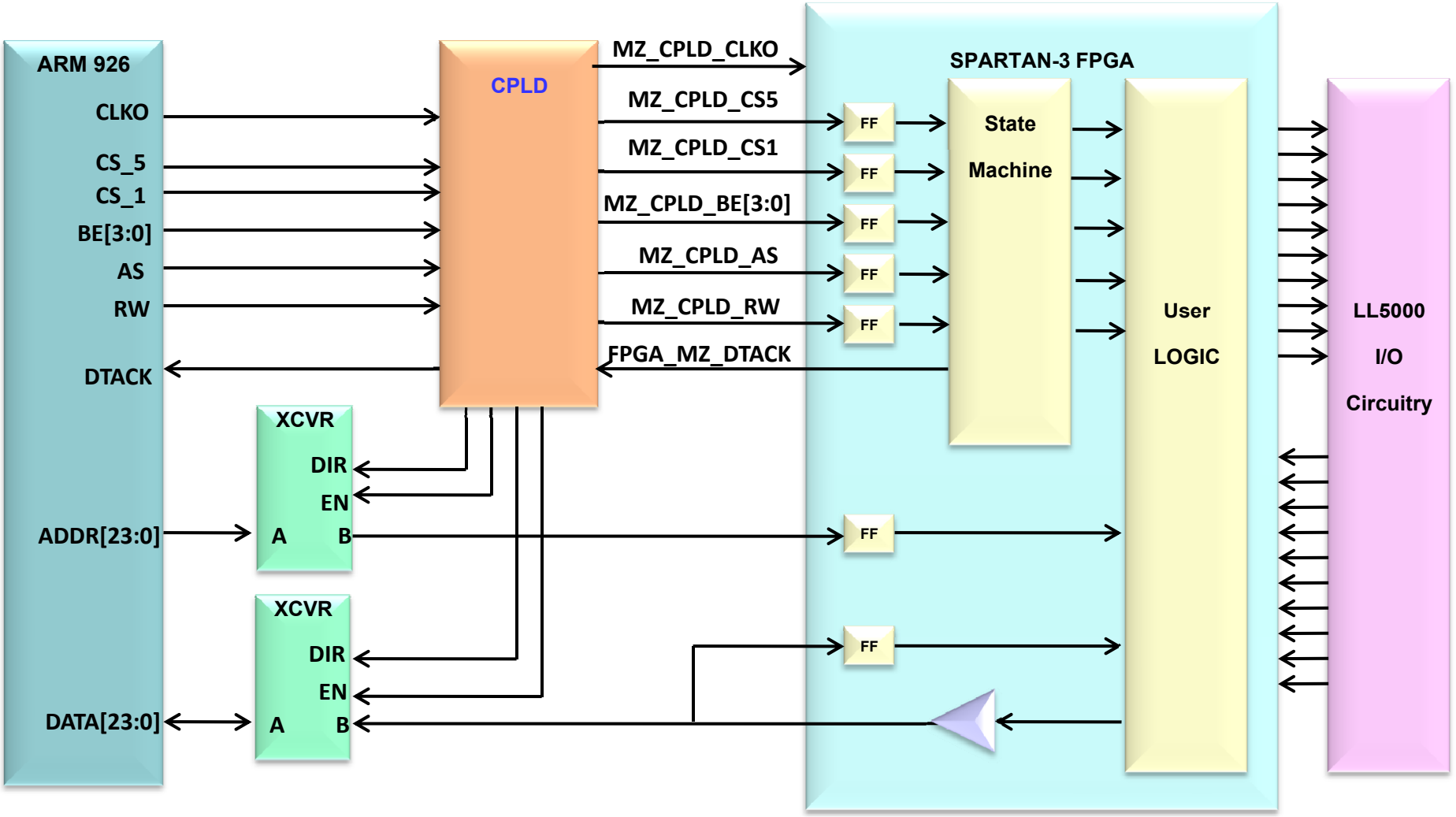
Project Details

- **Projects must be based on the course syllabus.**
  - **Proposal due Sep 24th**
- **Team size is 3-5 (no single person teams)**

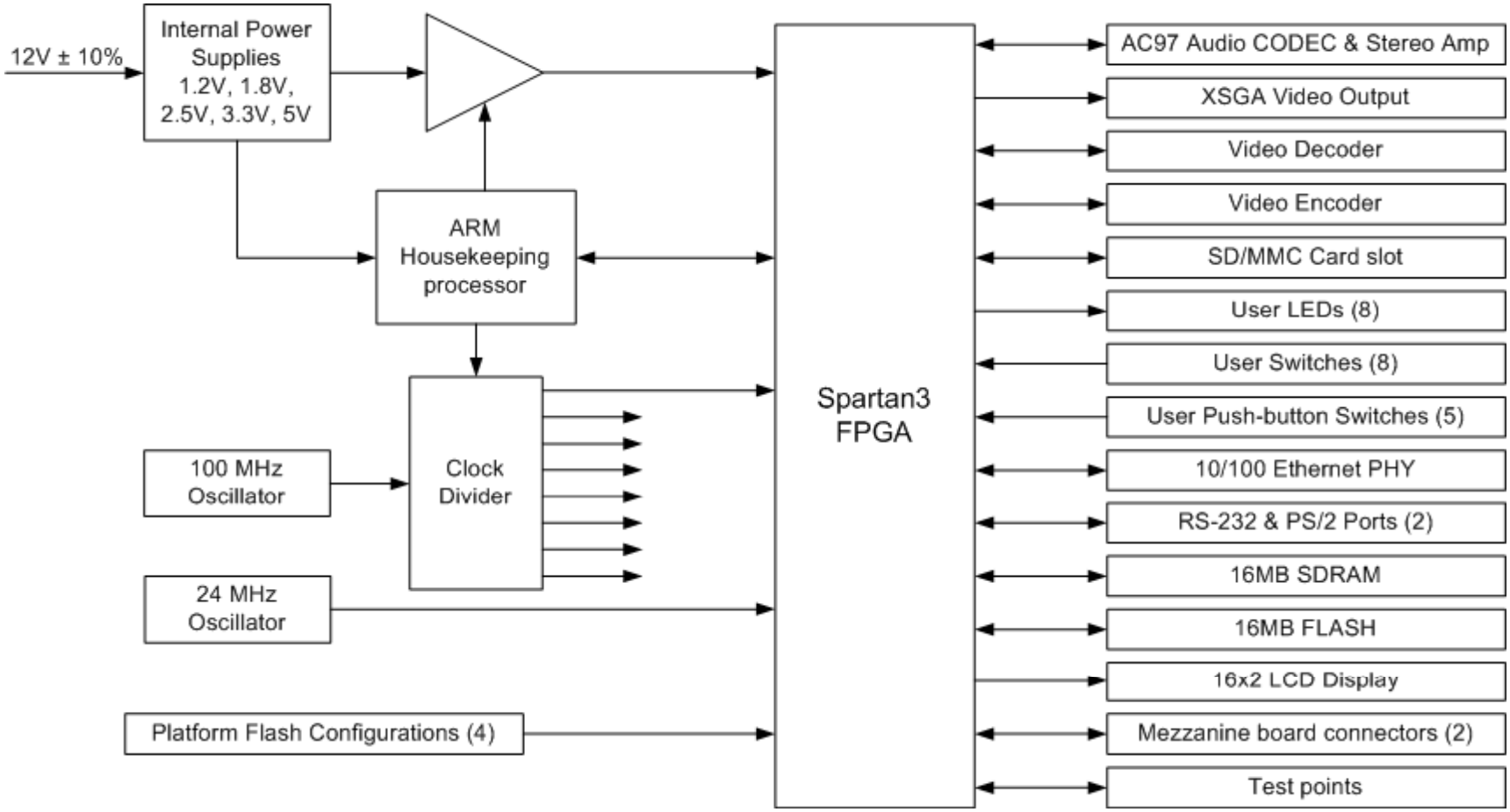
Project ideas:

- Arithmetic coprocessor implementing complex numerical algorithms
- MPEG compression / decompression accelerator
- JPEG encoder / decoder accelerator
- Encryption for the Advanced Encryption Standard (Rijndael algorithm)
- Public key encryption using Montgomery or Galois Field multiplier
- IP Packet Forwarding Engine (with possibly with Encryption / Decryption)
- USB 2.0 (Universal Serial Bus) Function Controller
- Audio MAC controller
- Ethernet controller for packet transmission, reception and encapsulation & decapsulation
- Develop a minimum footprint SW debugger in the Linux kernel
- http based GUI running on ARM Linux kernel
- eCOS port to TLL-6219 ARM board.
- gdb interface to TLL-6219 board
- Game or graphics coprocessor/accelerator for rendering applications
- Audio front & back end for DRM system
- Channel Estimation for DRM in FPGA
- Develop hardware debugger in FPGA and viewer running on Linux.

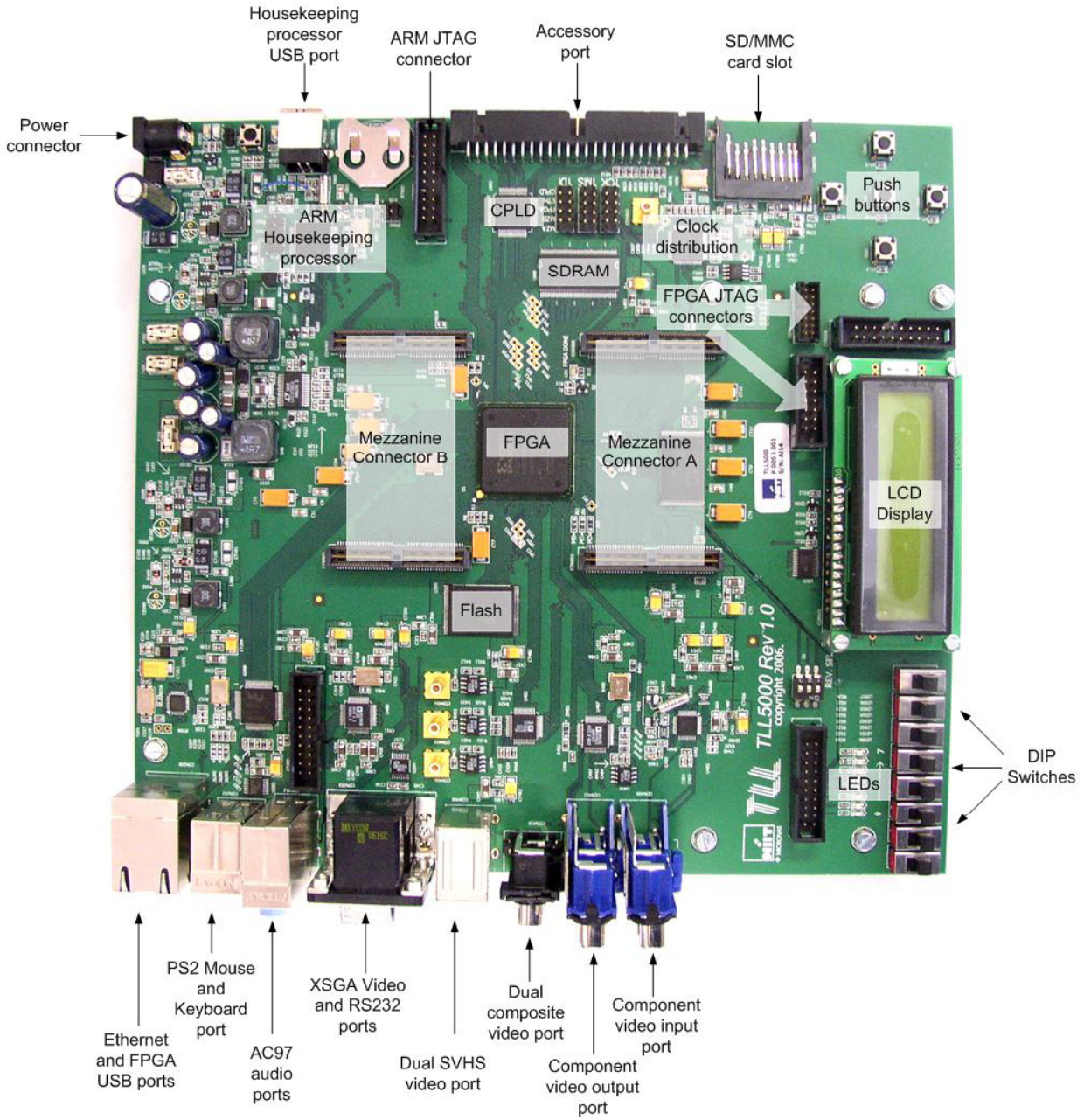
TLL5000/TLL6219 Prototype System Block Diagram



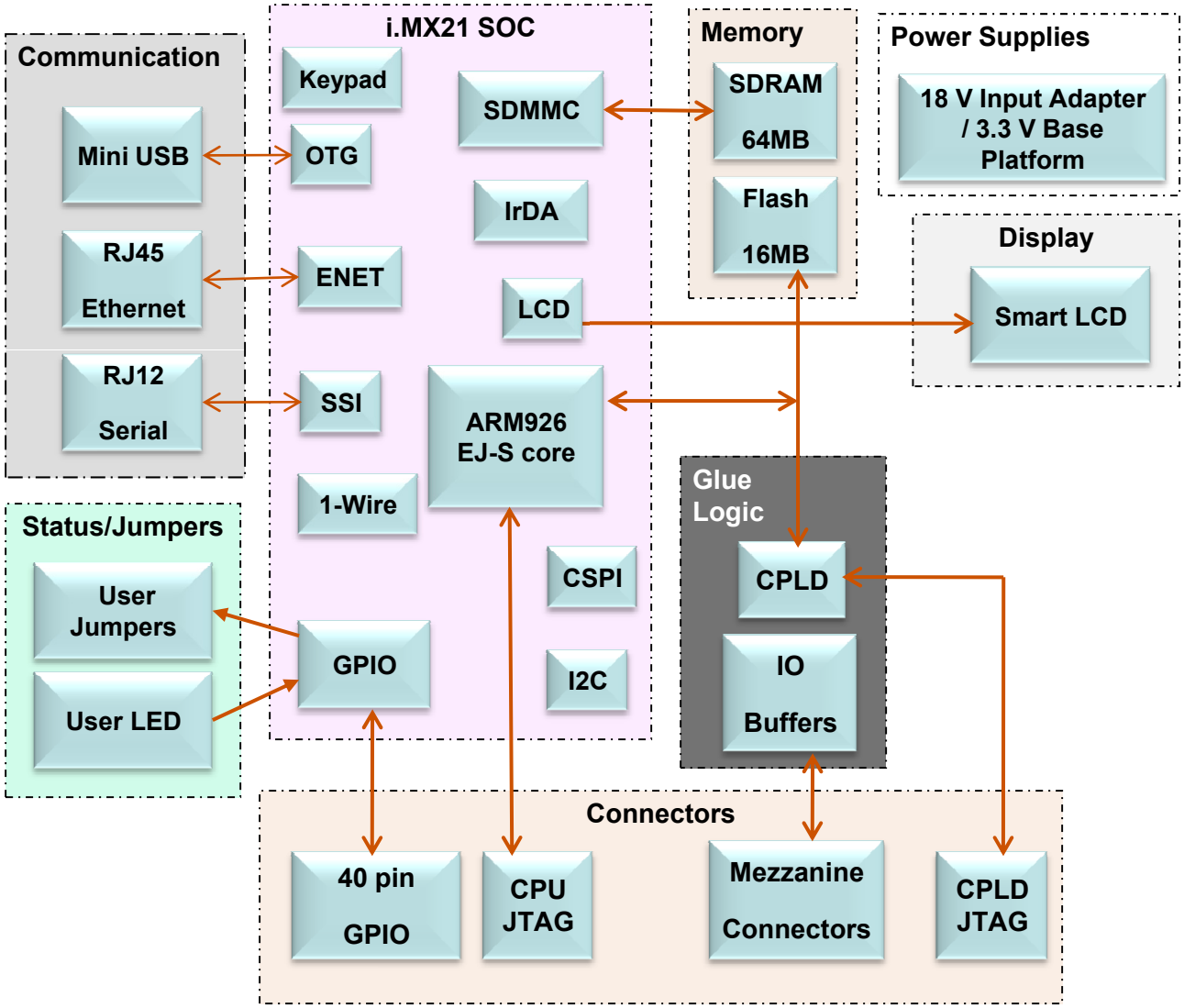
TLL-5000 Block Diagram



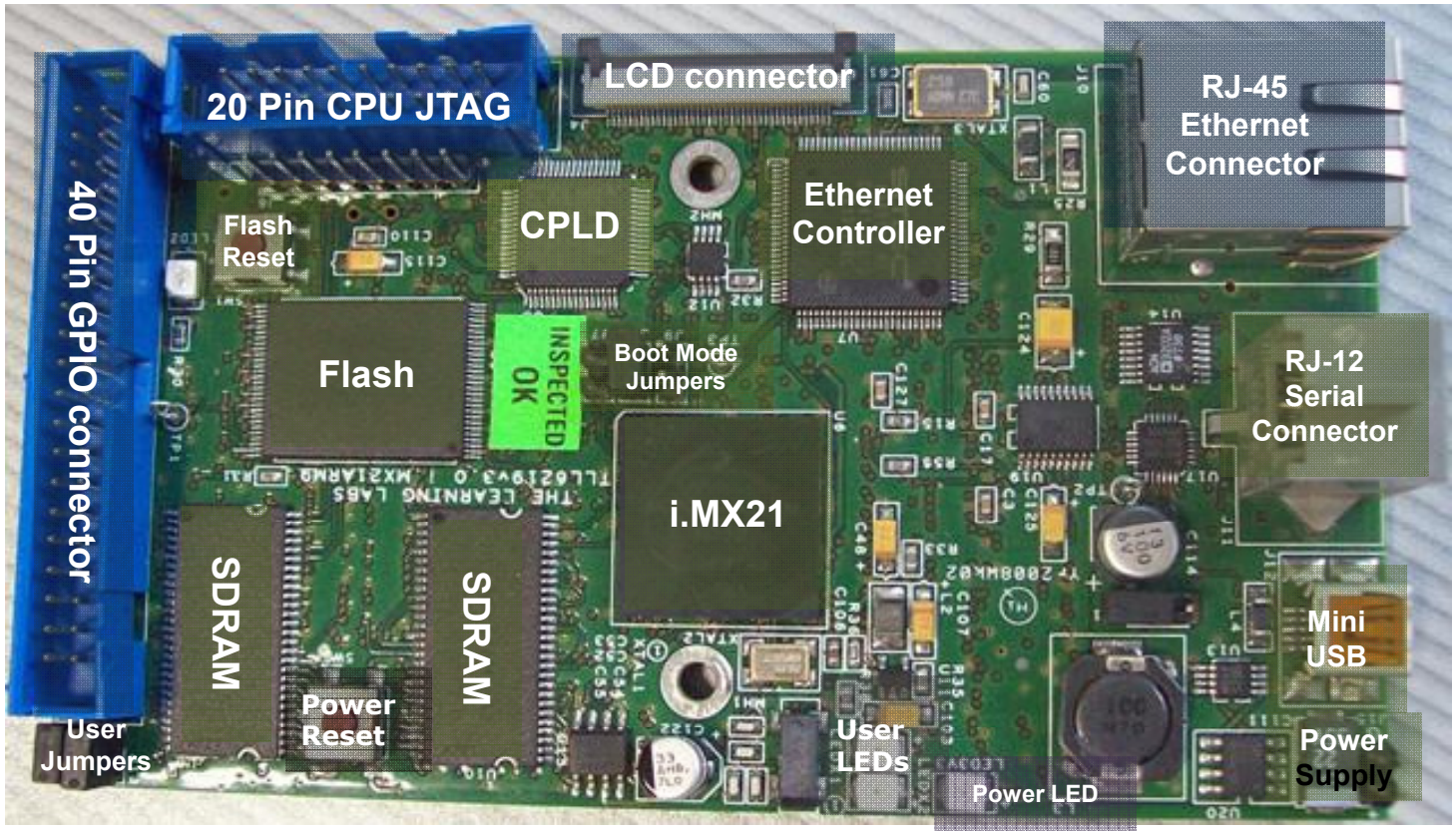
TLL-5000 Baseboard



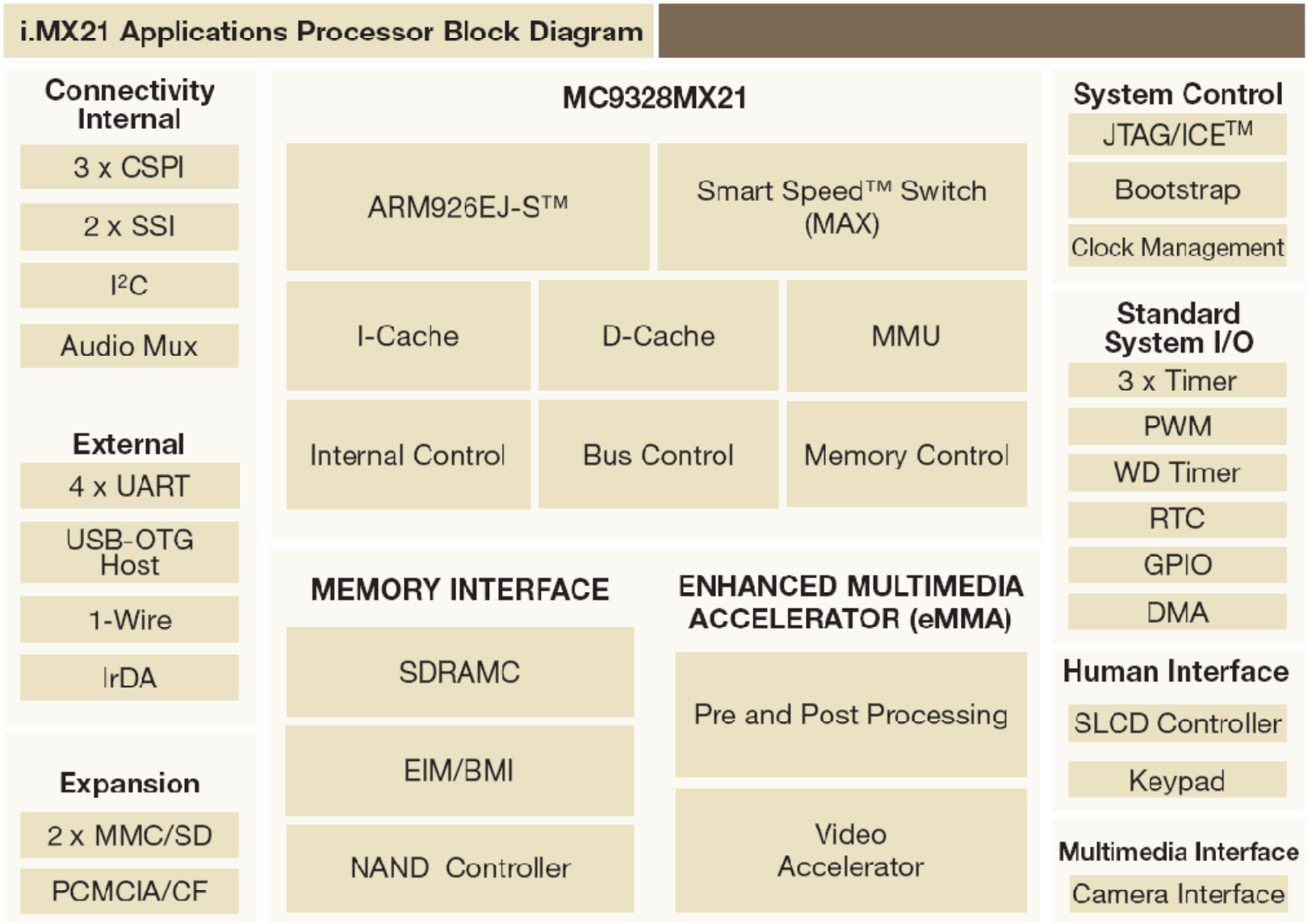
TLL 6219 Block Diagram



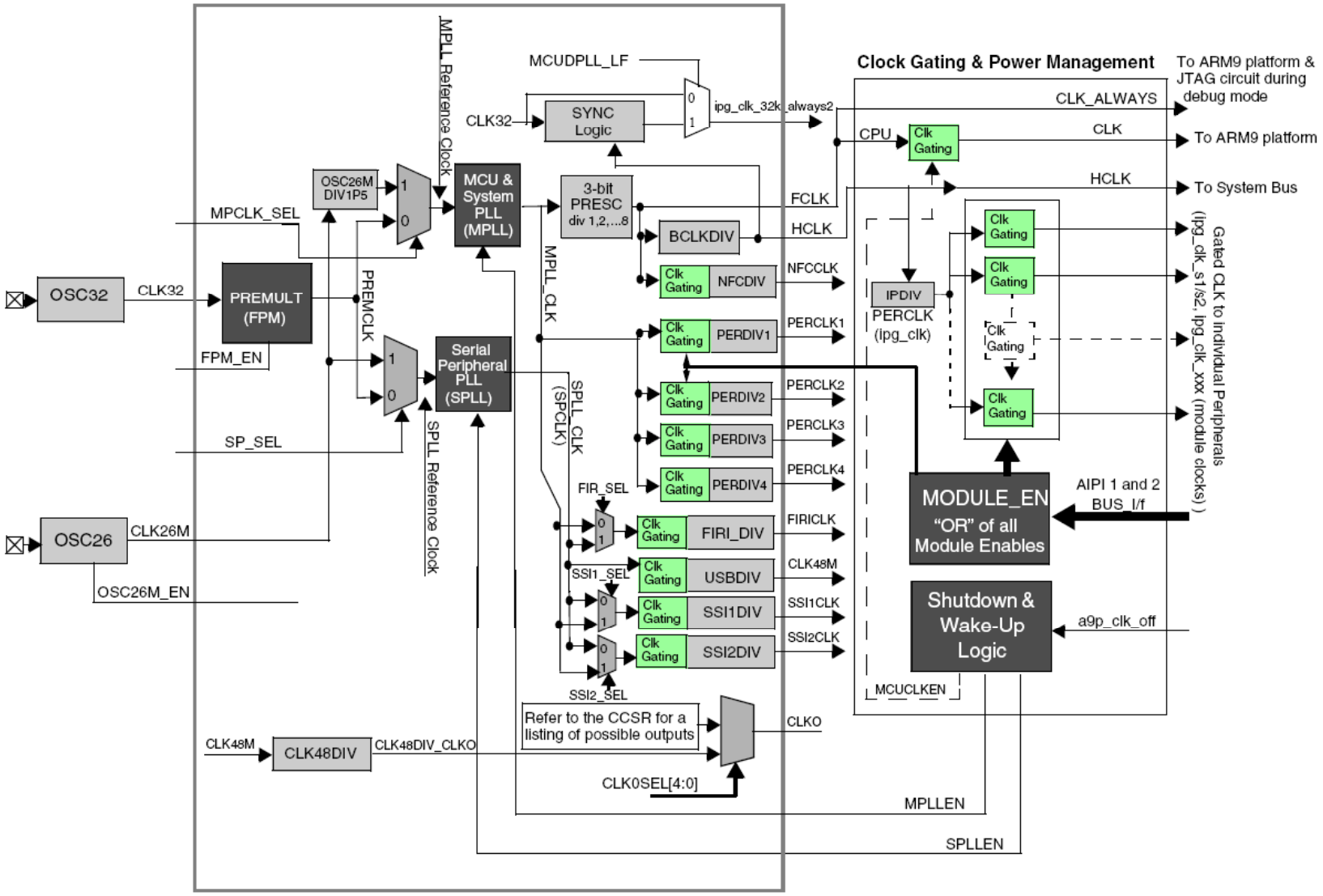
TLL6219 ARM 926-EJS Board



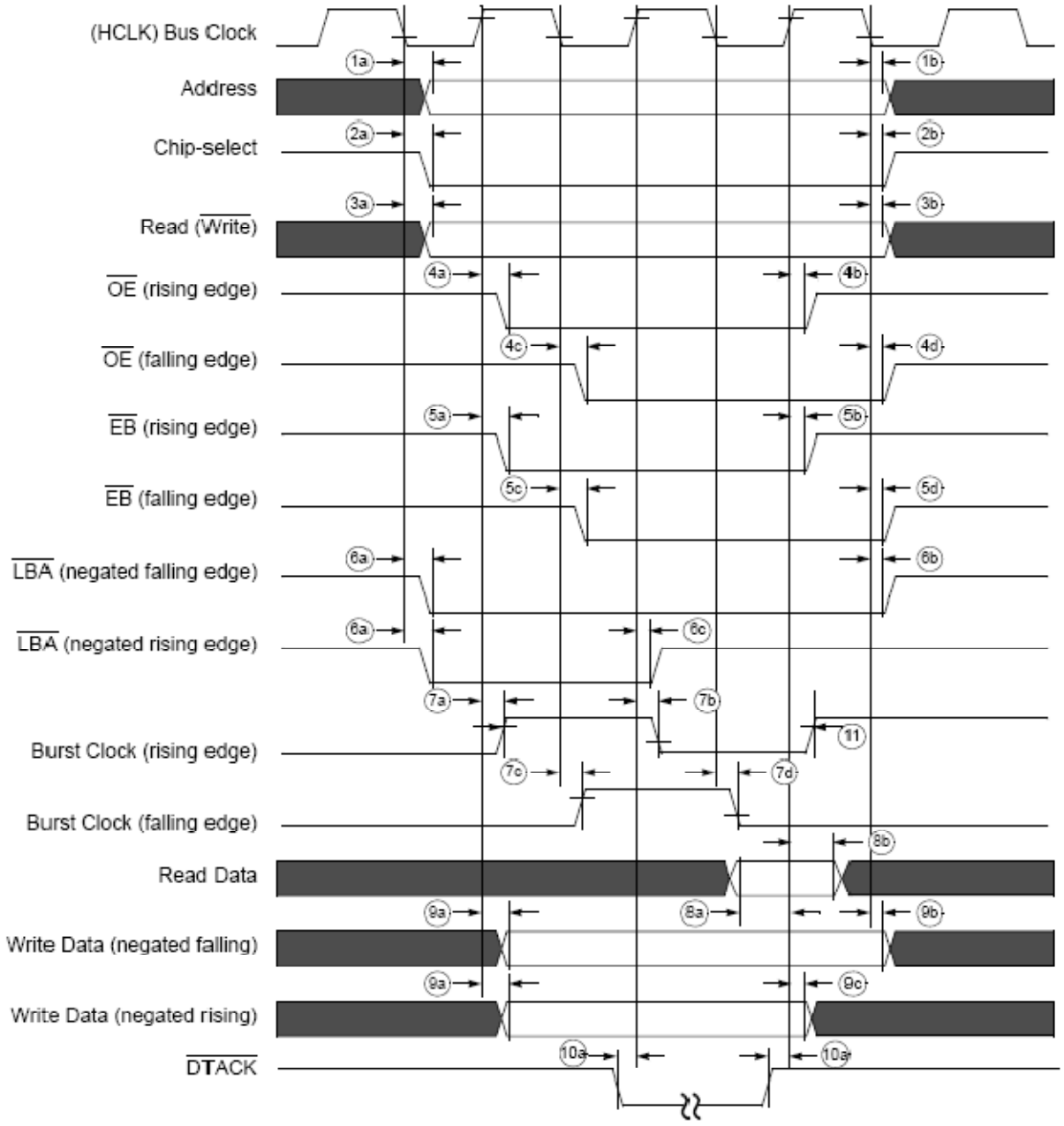
i.MX21 Features



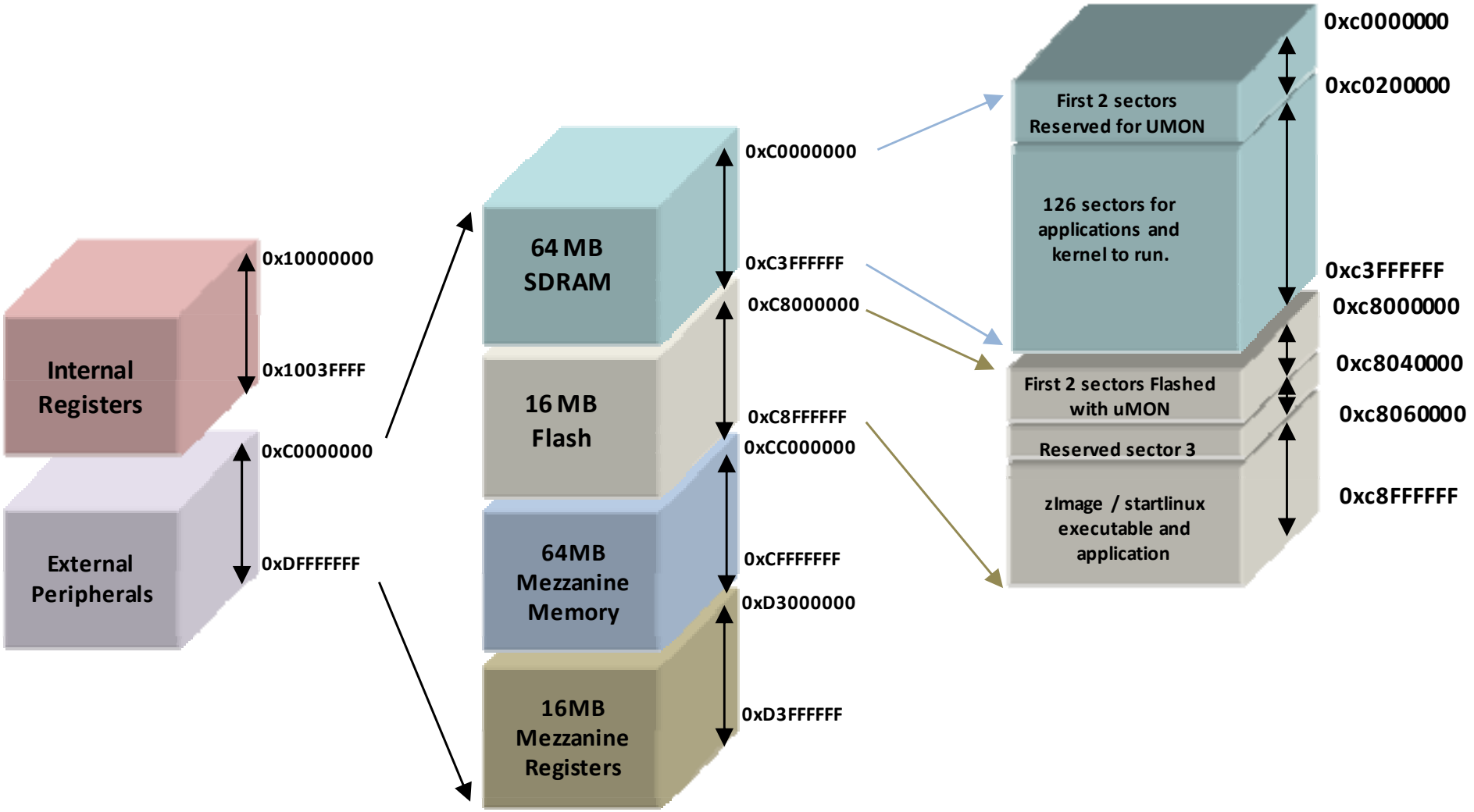
ARM926 Clock Generation



ARM926 Bus Cycle Waveforms and Timing



TLL6219 Memory Map



Web based resources

- Linux Devices: http://www.linuxdevices.com
- Embedded Linux Journal: http://embedded.linuxjournal.com
- Embedded.com: http://www.embedded.com/
  - Embedded Systems Programming magazine
- Circuit Cellar: http://www.circuitcellar.com/
- Electronic Design Magazine: http://electronicdesign.com/
- Berkeley Design technology, Inc.: http://www.bdti.com
- DSP Guru: http://www.dspguru.com/
- EEMBC: http://www.eembc.org/home.php
- EE Times Magazine: http://www.eet.com/
- Sensors Magazine: http://www.sensorsmag.com
- Embedded Systems Tutorial: http://www.learn-c.com/
- TechOnLine: http://www.techonline.com/
- Collections of embedded systems resources
  - http://www.ece.utexas.edu/~bevans/courses/ee382c/resources/
  - http://www.ece.utexas.edu/~bevans/courses/realtime/resources.html