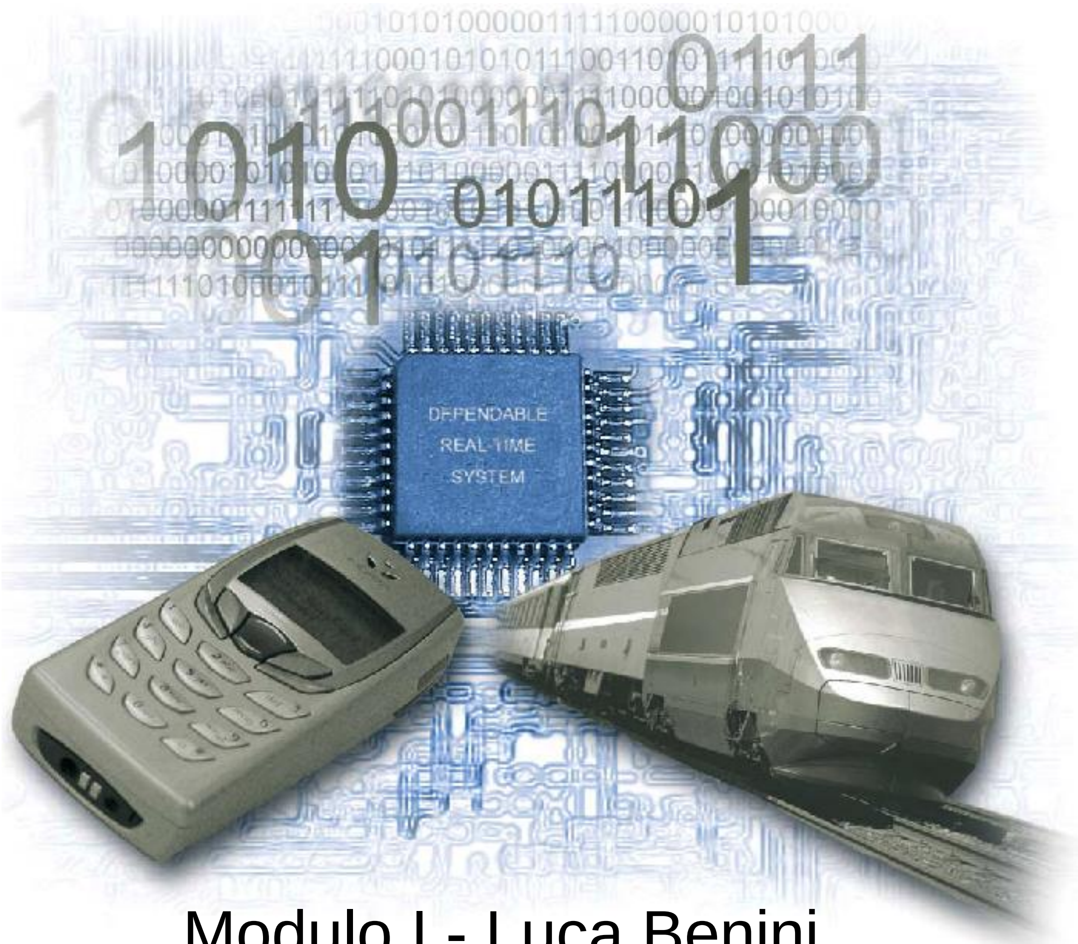
29035 – Laboratorio di ARCHITETTURE E PROGRAMMAZIONE DEI SISTEMI ELETTRONICI INDUSTRIALI T-A



Modulo I - Luca Benini
DEI Università di Bologna

AA 2014-2015

Class Organization

- Web: http://www-micrel.deis.unibo.it/LABARCH/
- Teacher: Prof. Luca Benini luca.benini@unibo.it (Prof. Elisabetta Farella mod. 2)
- Teaching assistants: - Dr. Filippo Casamassima filippo.casamassima@unibo.it - Ing. Domenico Balsamo domenico.balsamo2@unibo.it. Ing. Marco Tomasini marco.tomasini.88@gmail.com
- Organization:
  - Friday morning (9-11) Lectures (Benini) – Feb, Mar, Apr 10 last lecture (Mod 1)
  - Thursday afternoon (15-19): LAB1 – Starting 5/3
  - Friday morning (11-13) LAB1 – Extra time if needed
- Examination:
  - Lab Reports + Discussion (Oral)
- Pre-requisites
  - Basic digital electronics (i.e. CMOS gates, SRAM, DRAM)
  - Basic Computer Architecture
  - C programming (i.e. pointers, compiler, debugger)

Motivation for the Course

- Electronics everywhere
  - Disappearing computer.
  - Ubiquitous computing.
  - Pervasive computing.
  - Ambient intelligence.
  - Post-PC era.
- Basic technologies:
  - ***Embedded Systems.***
  - Communication technologies.








Definition

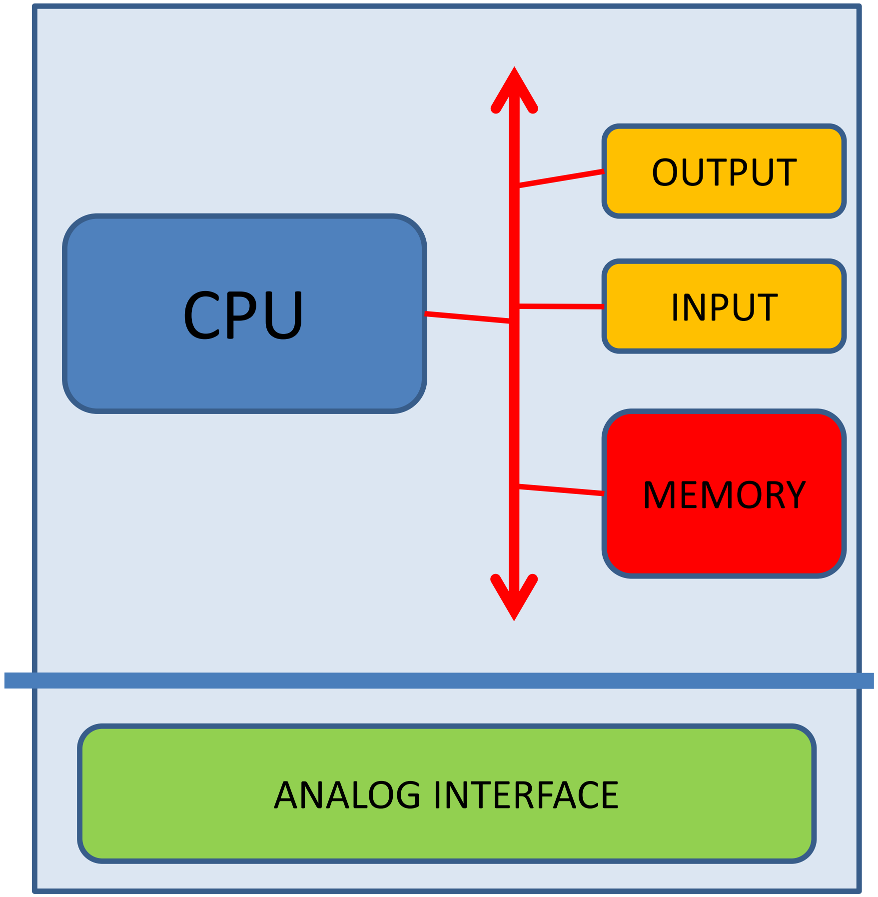
- Embedded system: Any device that includes a programmable computer, but is not itself a general-purpose computer.
- An embedded system has hardware and software parts.
- Take advantage of application characteristics to optimize the design.
- Respond, monitor and control the external environment using sensors and actuators.

Definition (2)

- From Wikipedia:

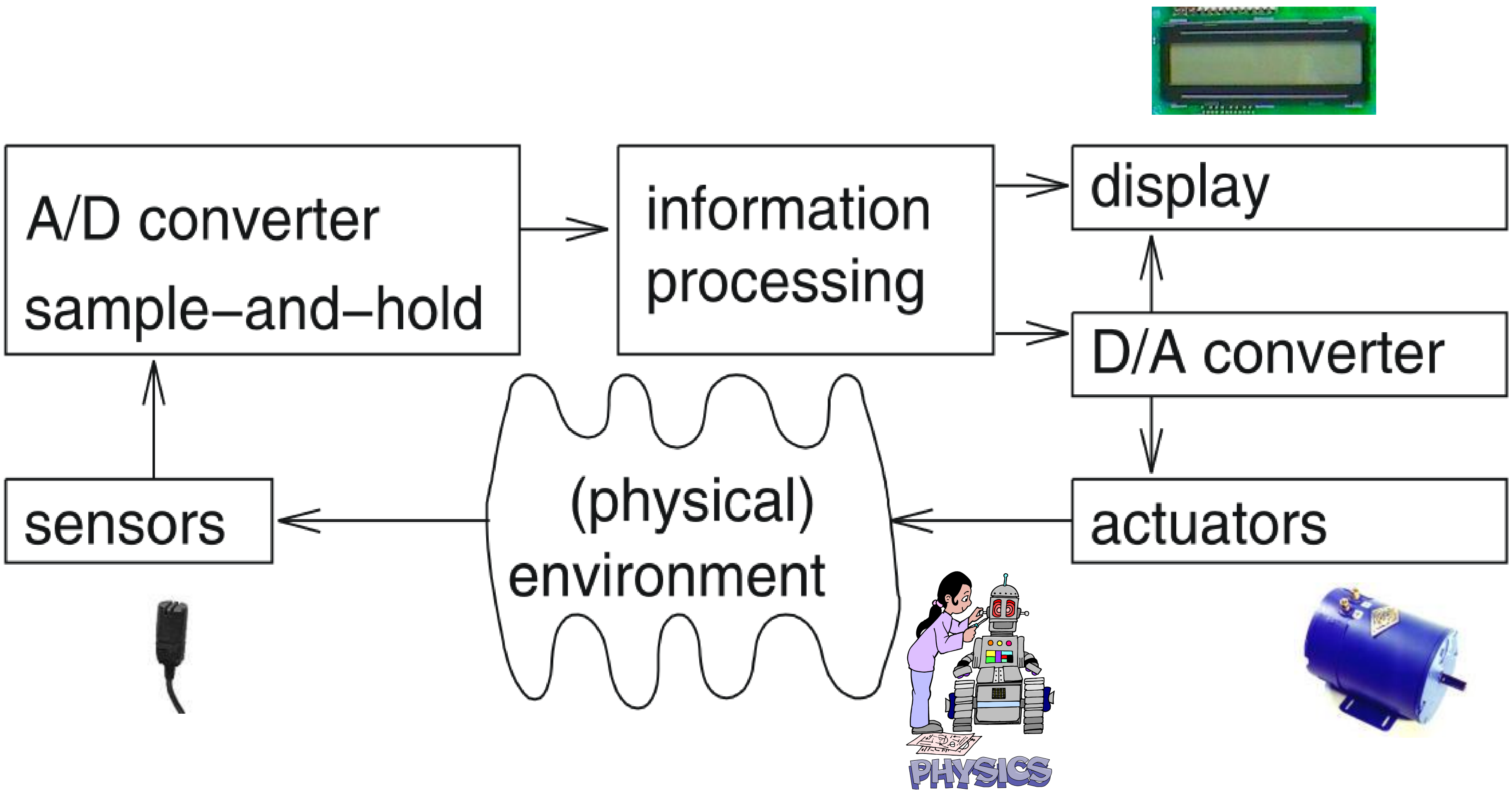
  *An **embedded system** is a special-purpose computer system designed to perform one or a few dedicated functions, often with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems control many of the common devices in use today.*

Embedding a Computer



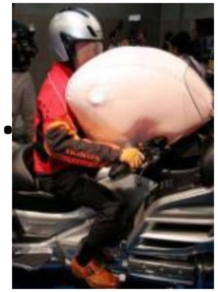
- The computer is embedded into an appliance.
- The embedded computer is not used for general purpose computing.
- The embedded computer interacts with external world: Analog interface is needed.

The Cyber-Physical Loop



Embedded Systems – Where?

- Transportation:
  - Automotive electronics.



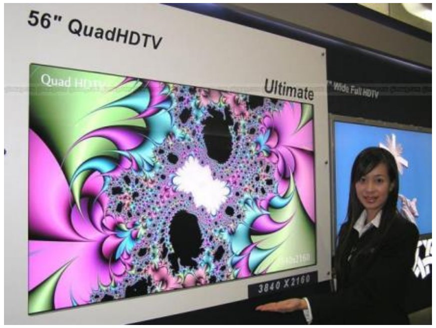
  - Avionics.

  - Trains.

Embedded Systems – Where? (2)

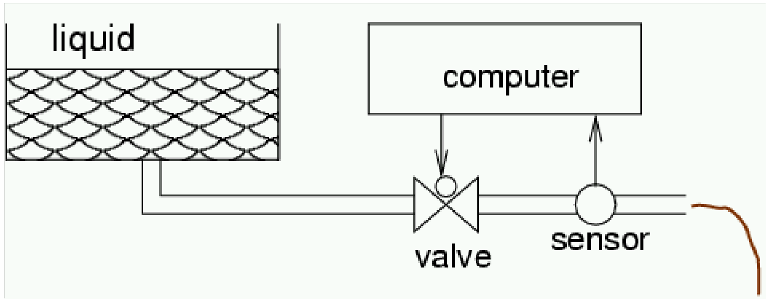
- Consumer:
  - Mobile.

  - Home.

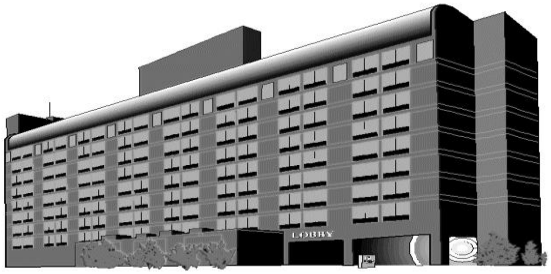


Embedded Systems – Where? (3)

- Smart Spaces:
  – Industrial automation.



  – Smart buildings.



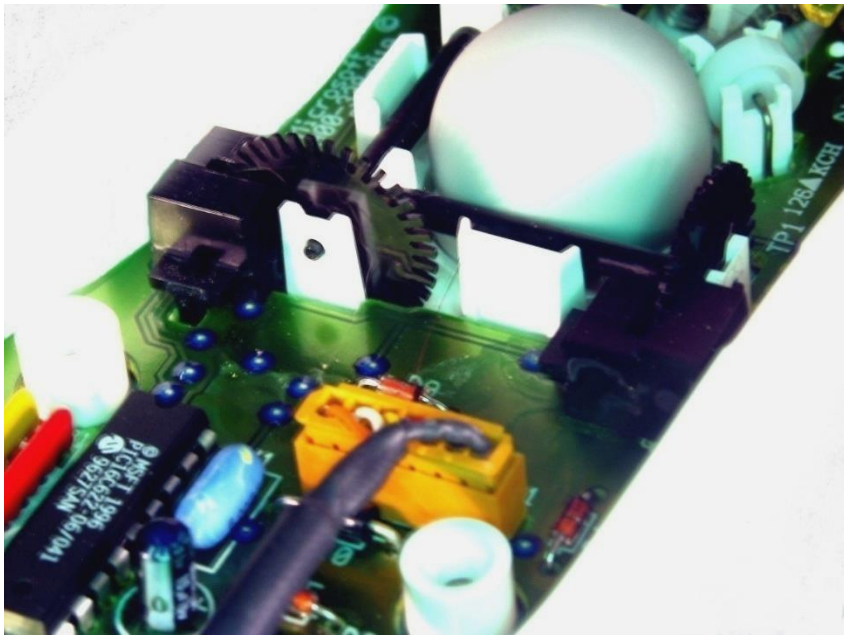
Embedded Systems - Examples

- Product: Sonicare Elite toothbrush.
- Microprocessor: 8-bit .
- Has a programmable speed control, timer, and charge gauge.

Embedded Systems - Examples (2)

- Product: Vendo Vue 40 vending machine.
- Microprocessor:
  Two 16-bit Hitachi H8/300H Processors.
- A robot hand dispenses items.

Embedded Systems - Examples (3)



- Any PC Mouse,
  Keyboard,
  or USB Device.
- Microprocessor:
  8-bit.

Embedded Systems - Examples (4)

- Any Disk Drive Microprocessor:
- Dual 32-bit Marvel.
- ARM SOC & mixed signal DSP.

Embedded Systems - Examples (5)

- Any Printer Microprocessor: Intel, Motorola, or ARM 32-bit RISC.

Embedded Systems - Examples (6)

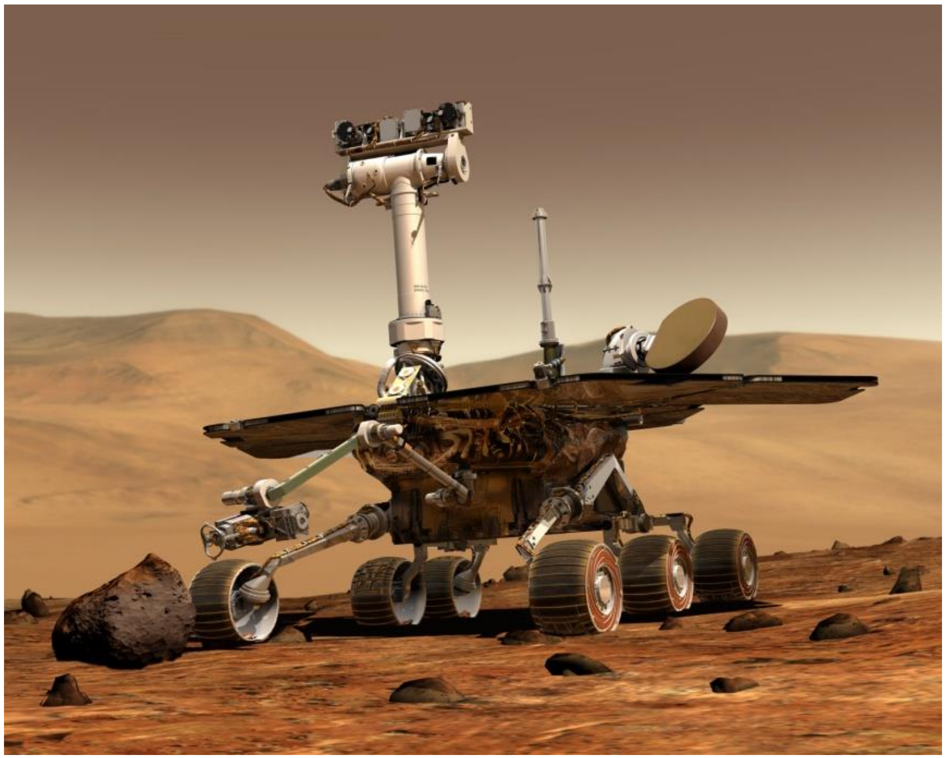
Product: Creative Labs Zen Vision:M Video & MP3 Player.

Microprocessor:
TI TMS320 DSP.

Embedded Systems - Examples (7)

- Canon EOS 30D Digital Camera.
- DIGIC II Image Processor.

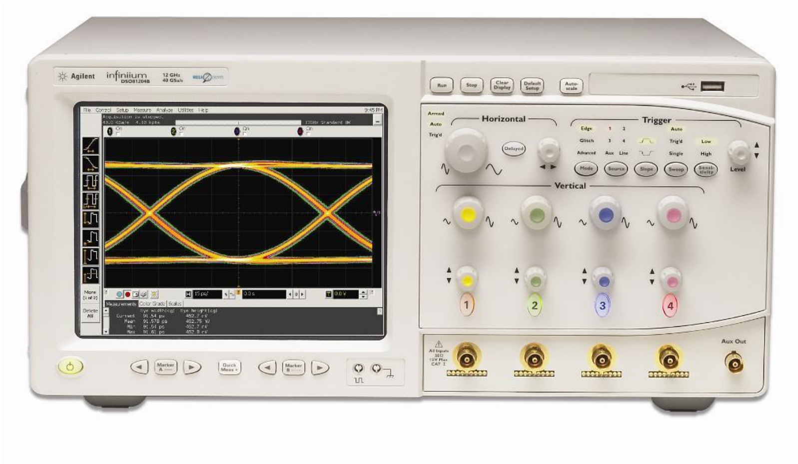
Embedded Systems - Examples (8)



Photograph courtesy of NASA/JPL CALTECH

- NASA's Twin Mars Rovers.
- Microprocessor: Radiation Hardened 20MHz PowerPC.
- Commercial Real-time OS.
- Software and OS was developed during multi-year flight to Mars and downloaded using a radio link.

Embedded Systems - Examples (9)



- Agilent Oscilloscope.
- Microprocessor: X86.
- OS: Windows XP.

Embedded Systems - Examples (10)

- Product: Atronic Slot Machine.
- Microprocessor: X86.
- OS: Windows CE.

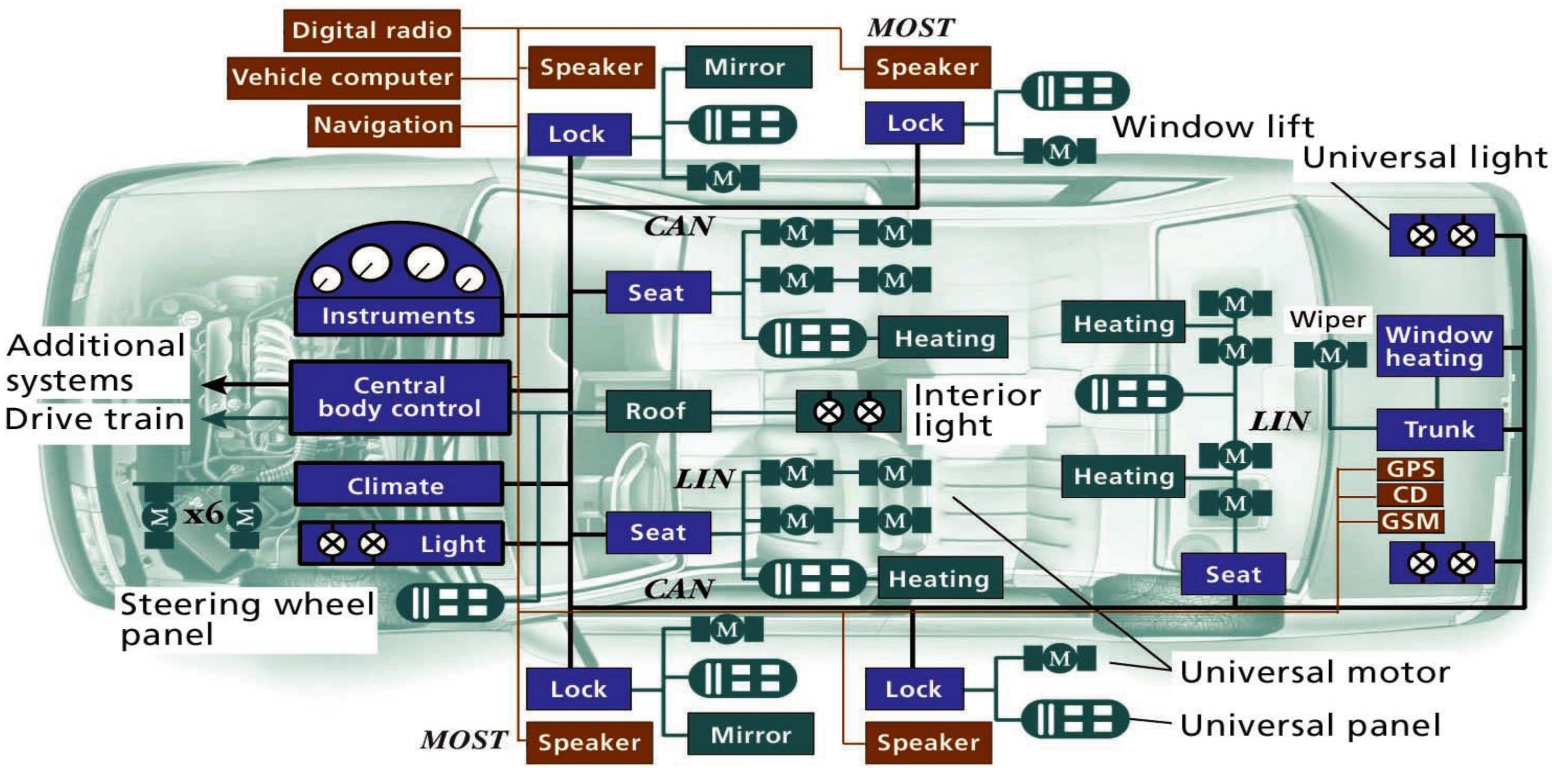
Embedded Systems - Examples (11)

- Sony Aibo robotic dog.
- Microprocessor: 64-bits MIPS RISC.

Networks and Embedded Systems

- An increasing number of embedded systems connect to the Internet.
  - Resource management.
  - Security.
- Many specialized networks have been developed for embedded systems:
  - Automotive.
  - Device control.

Cars as Distributed Embedded Systems



CAN Controller area network
GPS Global Positioning System
GSM Global System for Mobile Communications
LIN Local interconnect network
MOST Media-oriented systems transport

Embedded Systems - Examples (12)

- BMW 745i
  - Windows CE OS.
  - 53 8-bit μP.
  - 11 32-bit μP.
  - 7 16-bit μP.
  - Multiple Networks.

Relevance of Embedded Systems

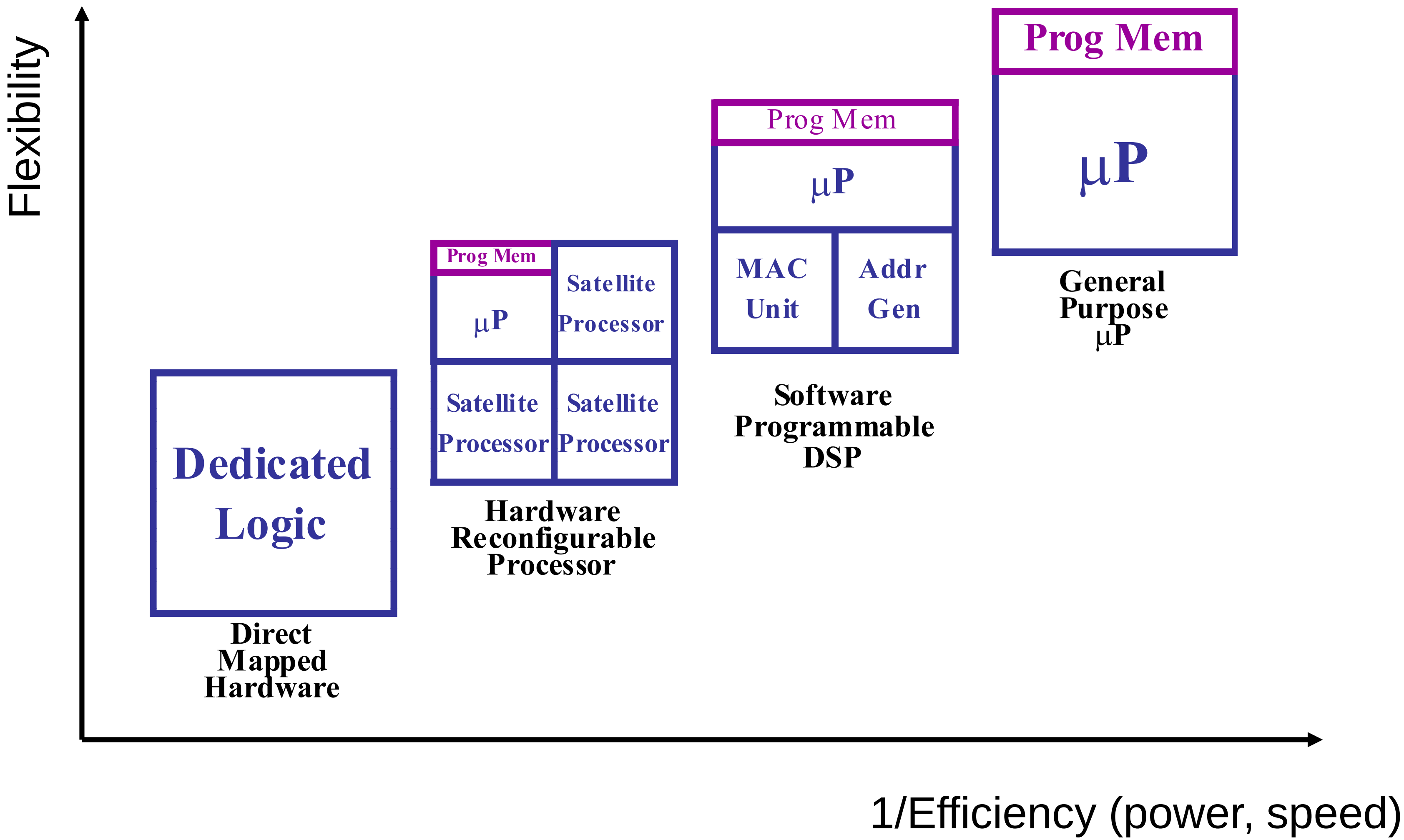
- Ratio of Embedded Devices / Desktop PCs > 100
- A typical house may contain over 50 embedded processors
- A high-end car can have over 100 embedded processors
- Embedded systems account for the most of the world’s production of microprocessors.

Summary

- Embedded devices can be found everywhere in large numbers!
- Most new devices are using 32-bit processors.
- The C family (C, C++, C#) is the most widely used language for embedded systems.
  - Simpler systems often are matched by simpler languages…

Implementation Fabrics

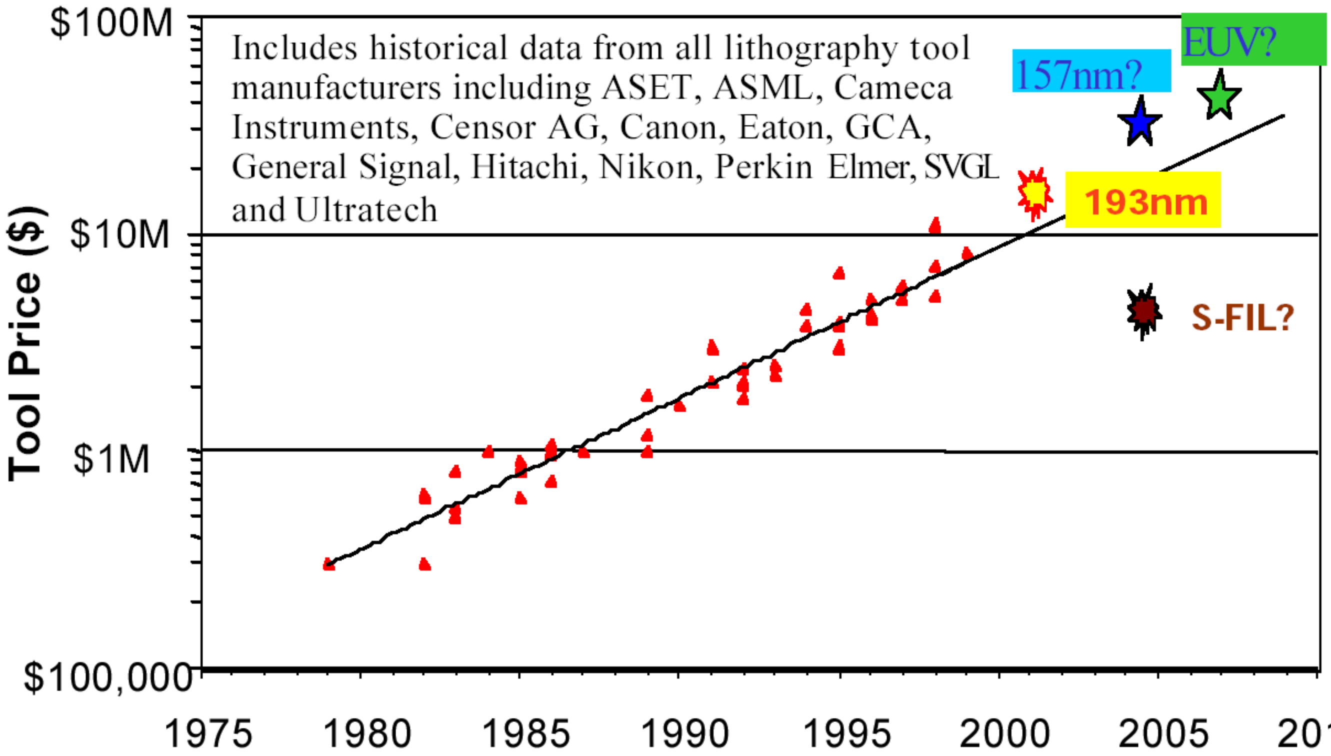
Architectural Choices



Challenges for implementation in hardware

– Lack of flexibility (changing standards).

– Mask cost for specialized HW becomes very expensive

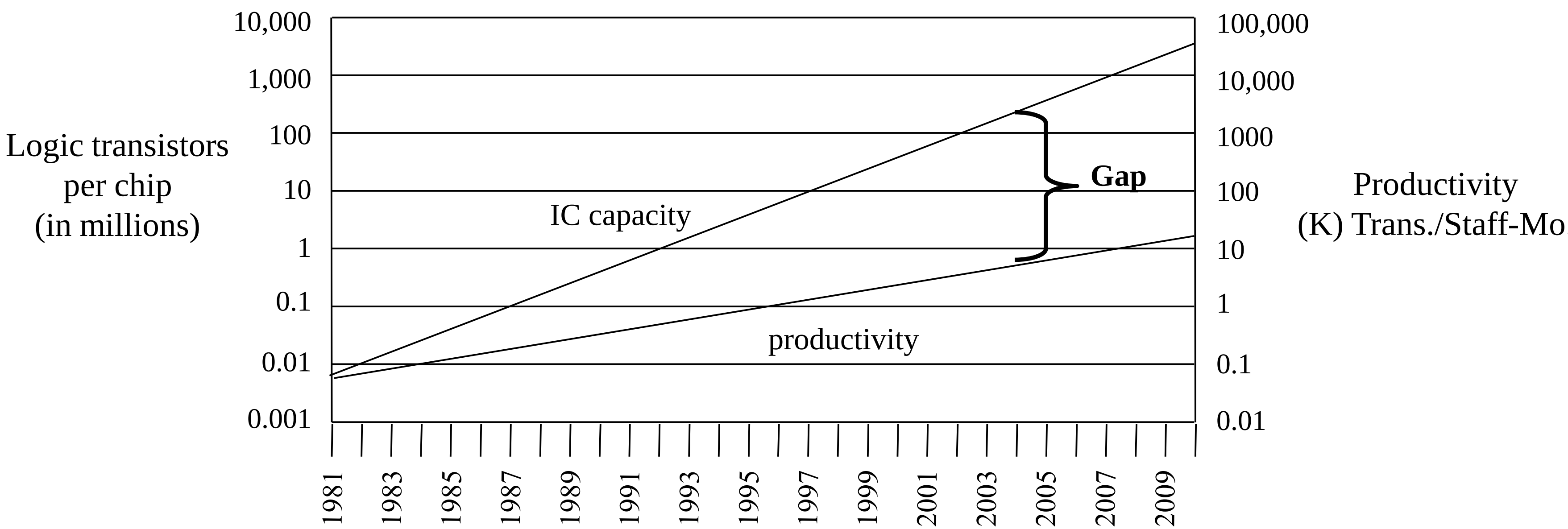


☞Trend towards implementation in Software

[http://www.molecularimprints.com/Technology/tech_articles/MII_COO_NIST_2001.PDF9]

Design productivity gap

- 1981 leading edge chip required 100 designer months
  - 10,000 transistors / 100 transistors/month
- 2002 leading edge chip requires 30,000 designer months
  - 150,000,000 / 5000 transistors/month
- Designer cost increase from $1M to $300M



The performance paradox

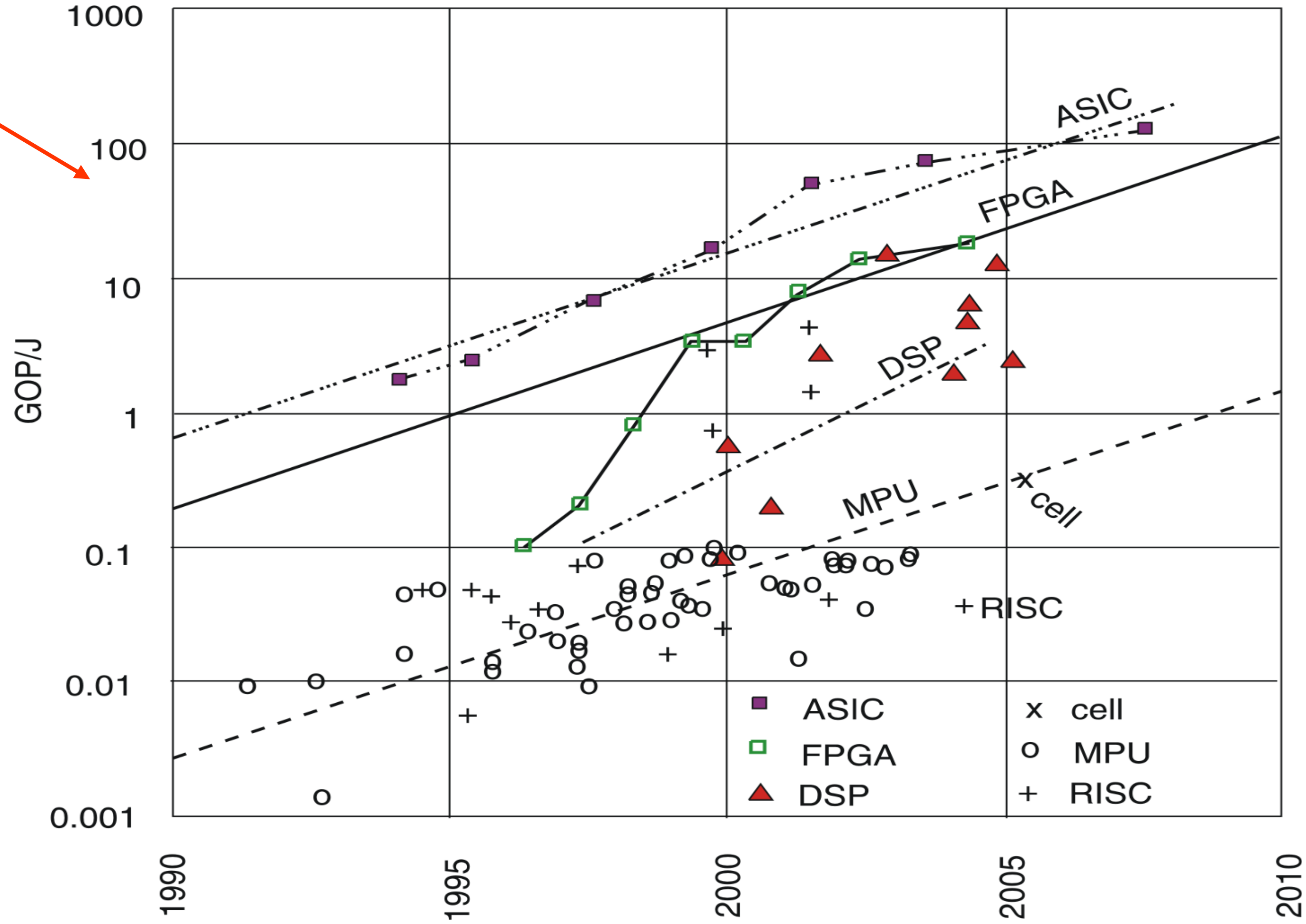
- Microprocessors use much more logic to implement a function than does custom logic.
- But microprocessors are often at least as fast:
  - heavily pipelined;
  - large design teams;
  - aggressive VLSI technology.

Power

- Custom logic uses less power, but CPUs have advantages:
  - Modern microprocessors offer features to help control power consumption.
  - Software design techniques can help reduce power consumption.
- Heterogeneous systems: some custom logic for well-defined functions, CPUs+software for everything else.

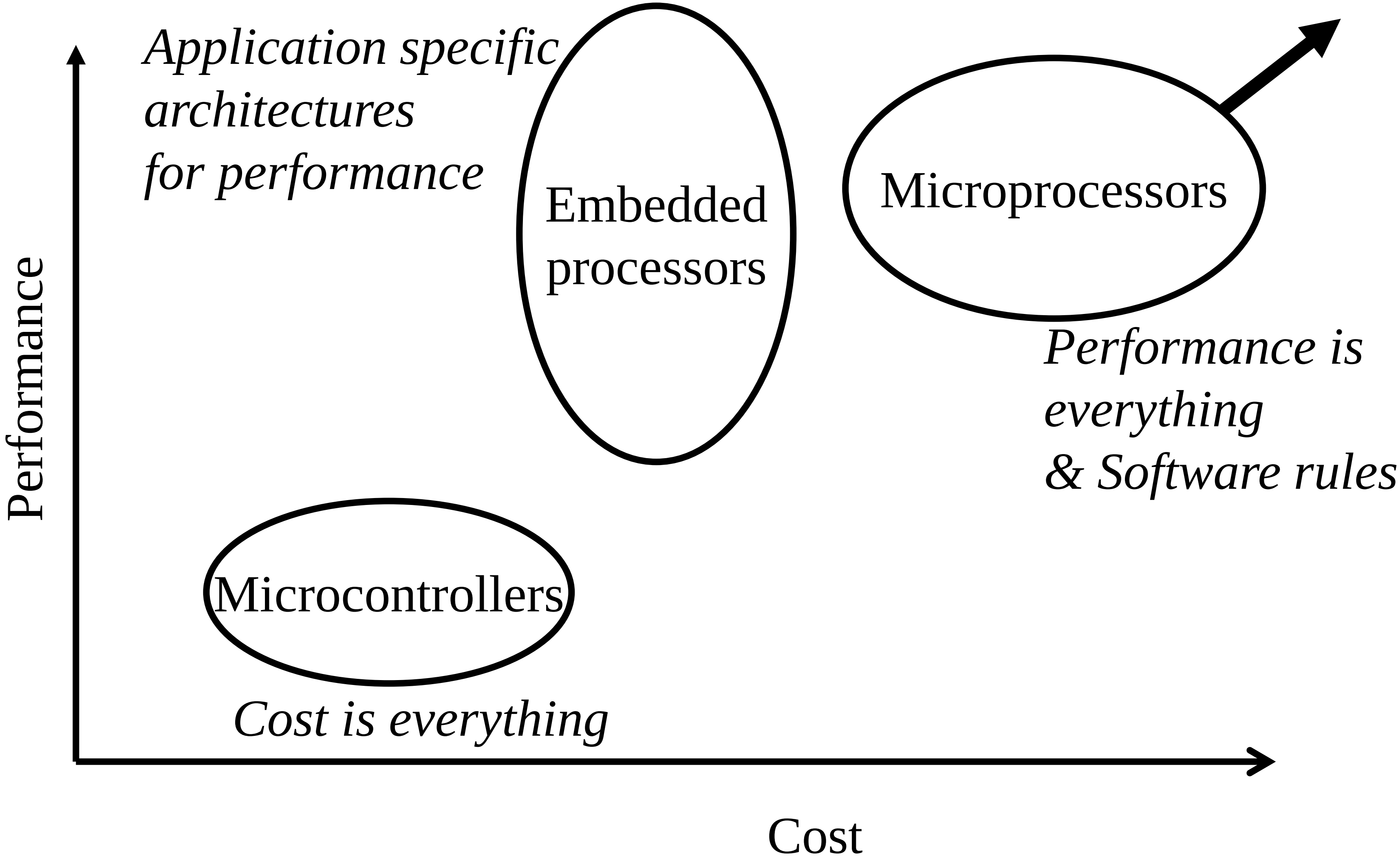
Importance of Energy Efficiency

“inherent power efficiency of silicon“

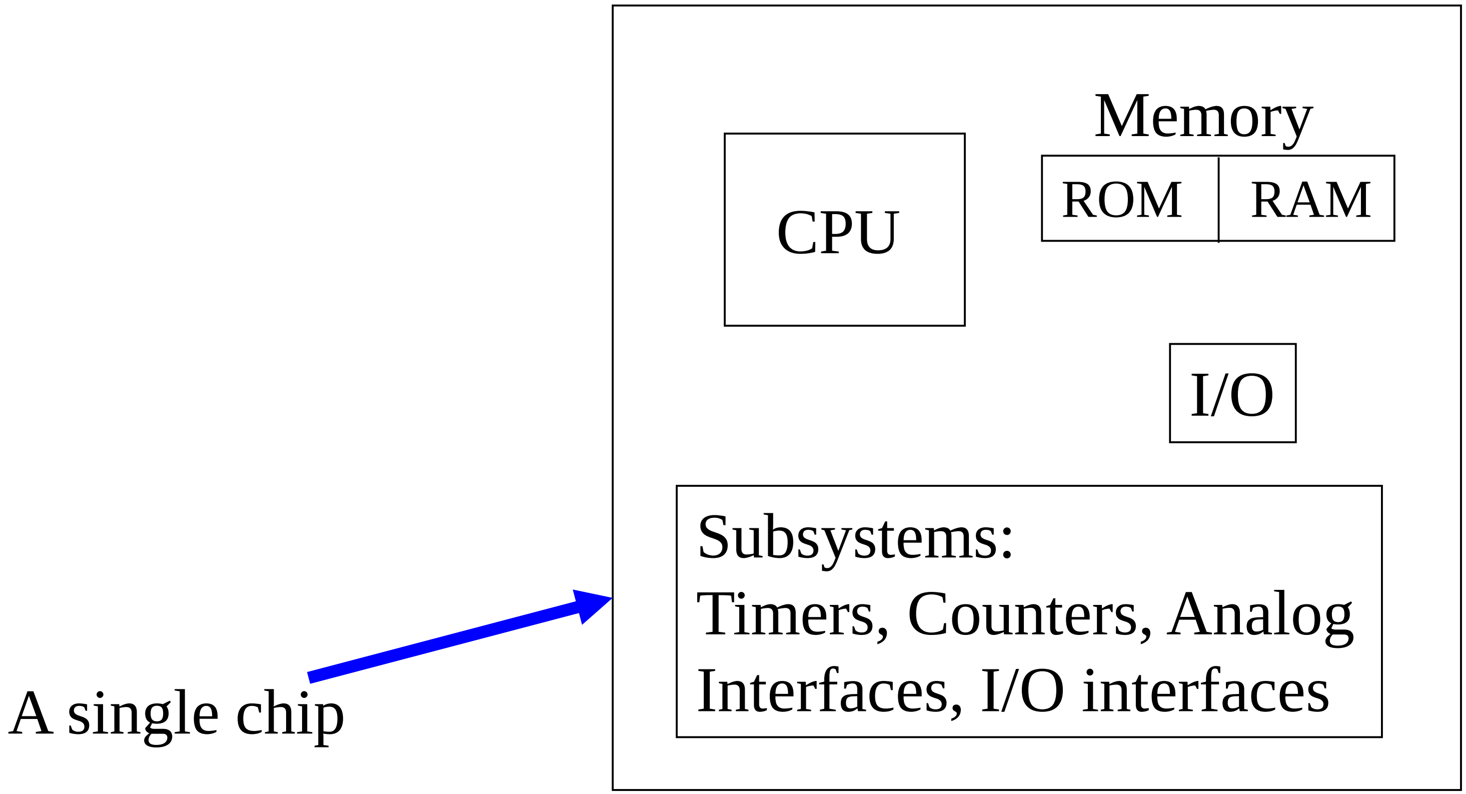


Efficient software design needed, otherwise, the price for software flexibility cannot be paid.

The Processor Design Space



Microcontrollers



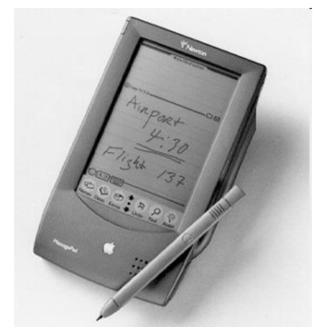
The ARM Processor Architecture

- ARM stands for “Advanced RISC Machine”
- – based on Reduced Instruction Set Computer (RISC) architecture
    - o trading simpler hardware circuitry with software complexity (and size)
    - o but latest ARM processors utilize more than 100 instructions

A Bit of ARM History

- Originally conceived to be a processor for the desktop system (Acorn®)
- – now entrenched in embedded markets

- First well-known product:
- – Apple®'s Newton™ PDA (1993)
  based on an ARM6™ core



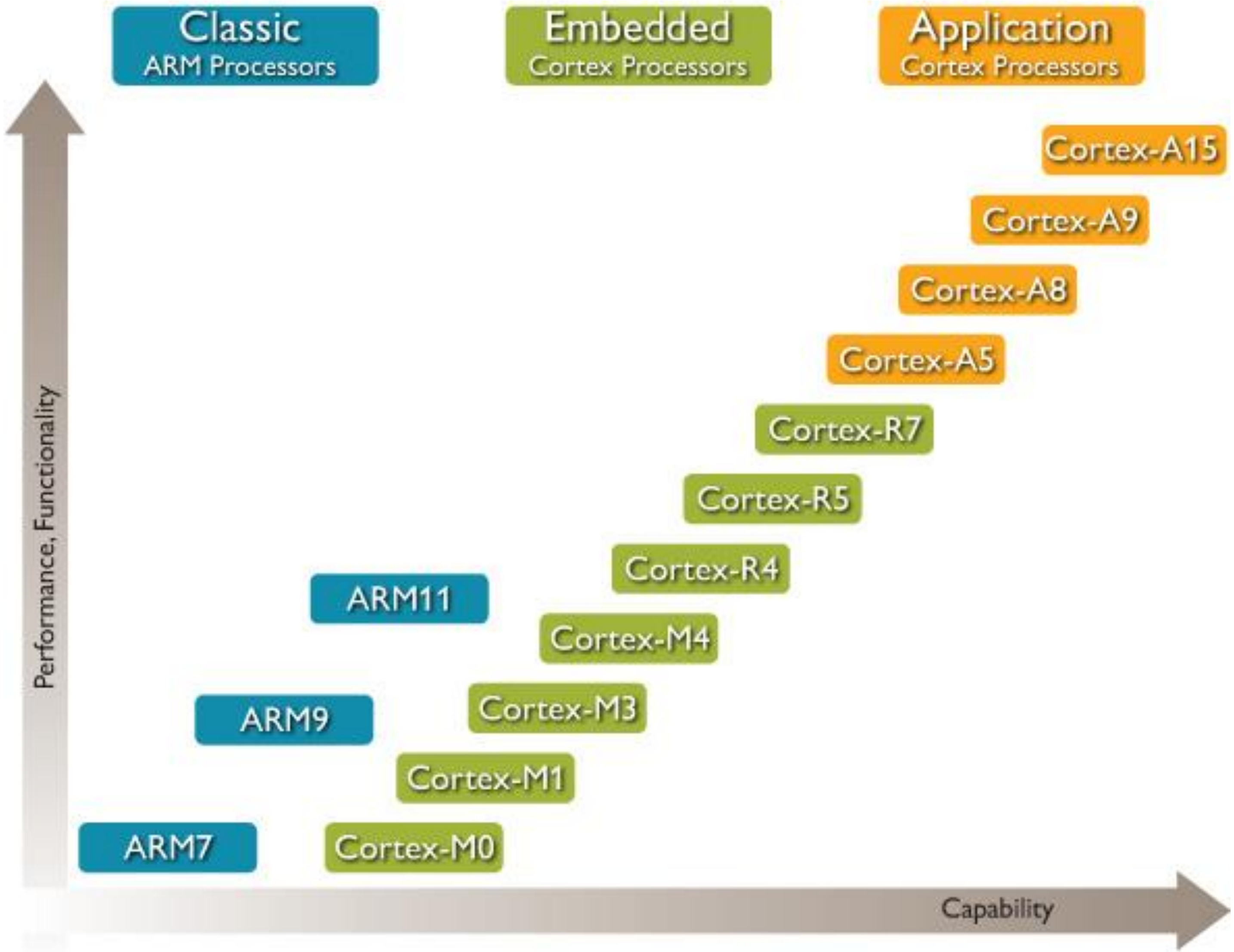
- Significant breakthrough:
- – Apple®'s iPod® (2001)
  based on an ARM7™ core

The Microprocessor Market

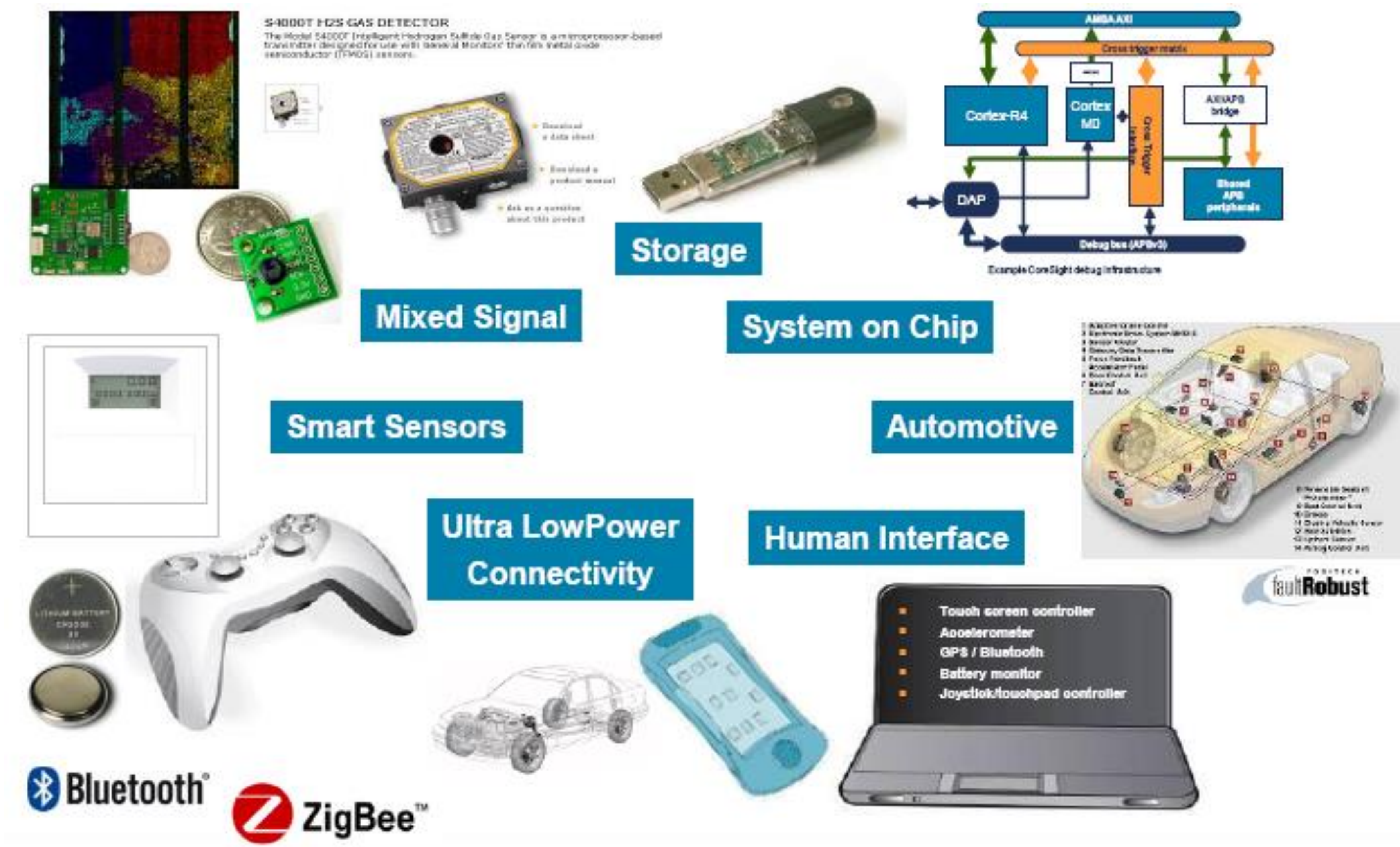
- In 2007,
  - 13 billion microprocessors were shipped
  - 3 billion were embedded processors based on the ARM architecture
  - 150 million were for the PC, notebook, and workstation
- By February 2008,
  - 10 billion ARM-based processors have been produced

ARM Partnership Model

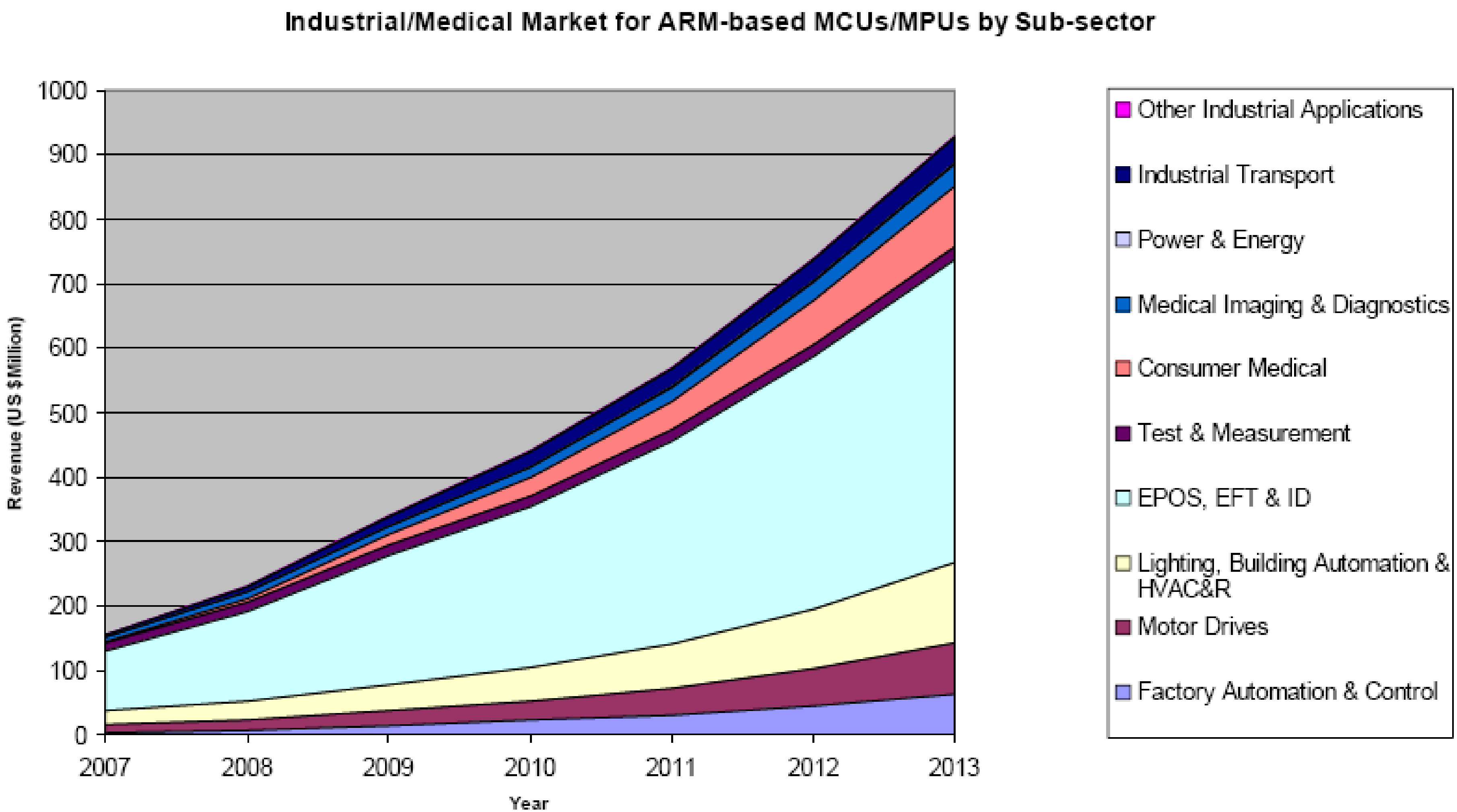
ARM Processors Families



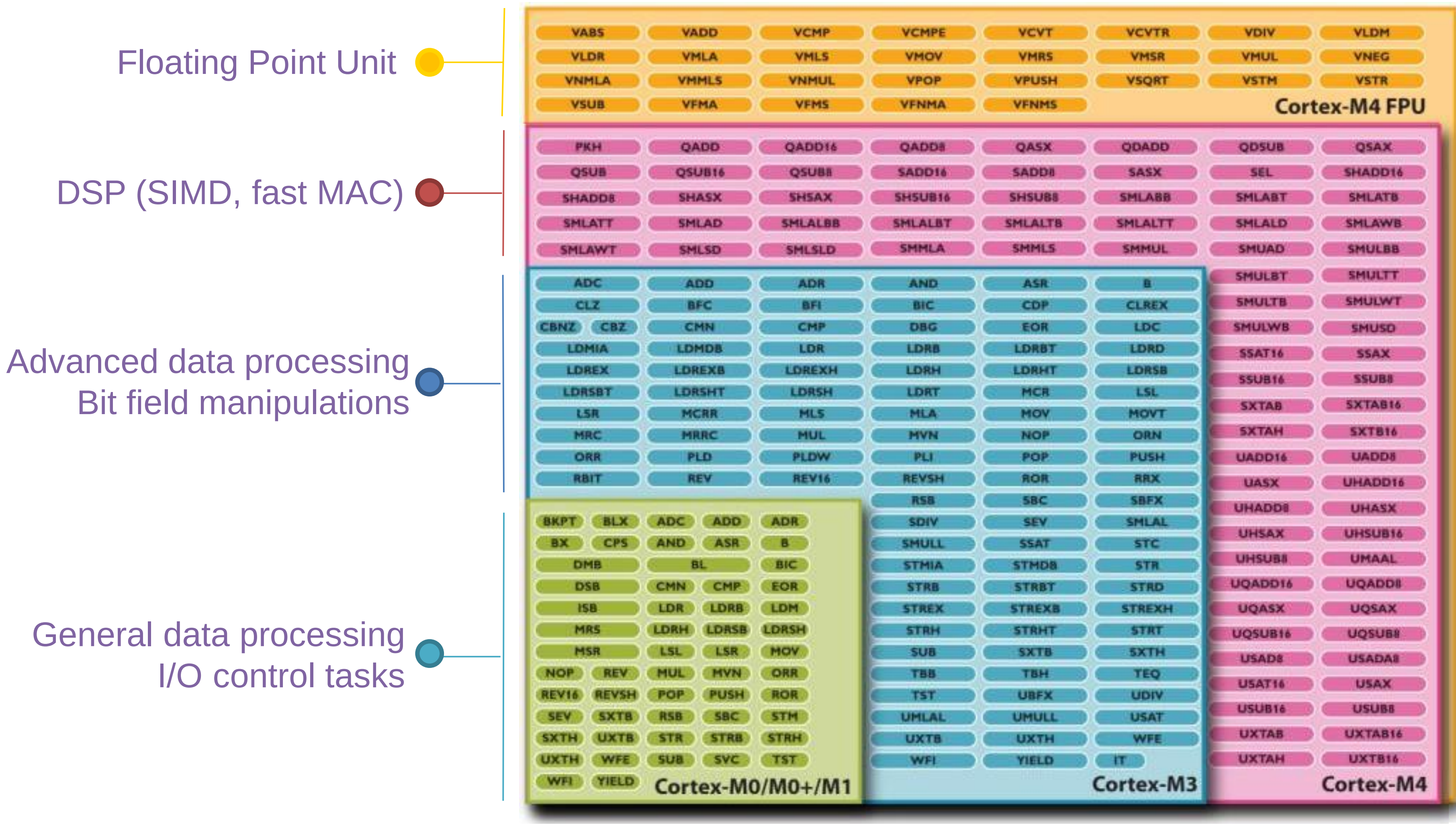
ARM Cortex-M Family



ARM Cortex-M market



Cortex-M processors

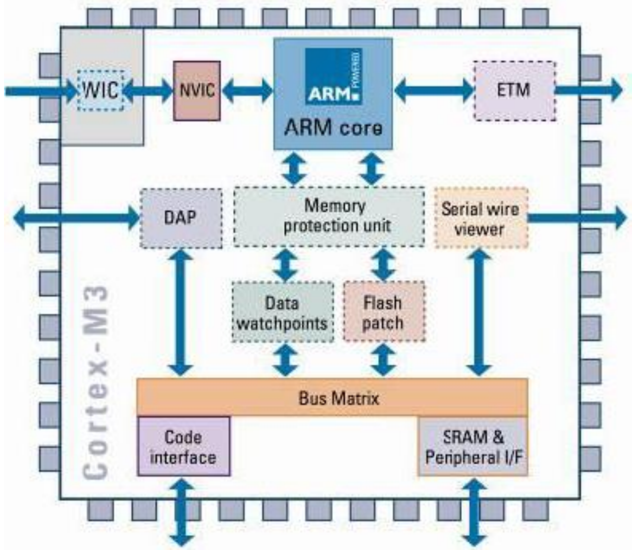


Binary compatible

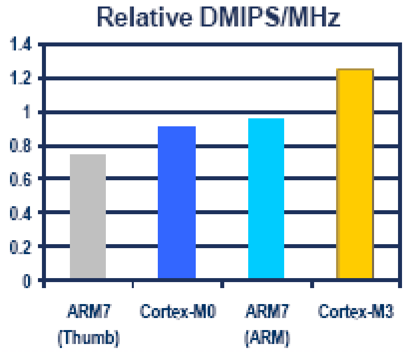
Source: ARM

ARM Cortex-M3 Processor

- Cortex-M3 architecture
- Harvard bus architecture
  - 3-stage pipeline with branch speculation
- Configurable nested vectored interrupt controller (NVIC)
- Wake-up Interrupt Controller (WIC)
  - Enables ultra low-power standby operation
- Extended configurability of debug and trace capabilities
  - More flexibility for meeting specific market requirements
- Optional components for specific market reqs.
  - Memory Protection Unit (MPU)
  - Embedded Trace Macrocell™(ETM™)
- Support for fault robust implementations via configurable observation interface
  - EC61508 standard SIL3 certification
- Physical IP support
  - Power Management Kit™(PMK) + low-power standard cell libraries and memories enable0.18µm Ultra-Low Leakage (ULL) process



Comparison	Cortex-M0	Cortex-M3
DMIPS/MHz	0.9	1.25
Gate count	12k	43k
Number interrupts	1-32 + NMI	1-240 + NMI
Interrupt priorities	4	256
Breakpoints, Watchpoints	4/2, 2/1	8/4, 2/1
MPU, integrated trace option	No	Yes
Hardware Divide	No	Yes



STM32 ARM® Cortex™-M3

STM32F10x Product Lines

All lines include:

- Multiple communication peripherals
  Up to 5 x USART, 3xSPI, 2xI²C
- ETM*
- FSMC**
- Dual 12-bit DAC***
- Multiple 16-bit Timers
- Main Osc 4-16MHz (25MHz on 105/107)
- Internal 8 MHz RC
  and 40 kHz RC
- Real Time Clock with Battery
  domain & 32KHz ext osc
- 2 x Watchdogs
- Reset circuitry and
  Brown Out Warning
- Up to 12 DMA channels

* Performance/Access Lines 256KB Flash or more, Value Line with 100+pins and ALL Connectivity devices

** Performance and Access and Value devices with 256KB Flash or more.

*** ALL Value line devices and Performance/Access devices with 256KB Flash or more

+

Connectivity Line: STM32F107

72MHz CPU	Up to 256 KB Flash / 64KB SRAM	2x12-bit ADC (1μs) TempSensor	USB 2.0 OTG (FS)	2 x Audio Class I2S	2 x CAN	PWM timer	Ethernet IEEE1588

Connectivity Line: STM32F105

72MHz CPU	Up to 256 KB Flash / 64KB SRAM	2x12-bit ADC (1μs) TempSensor	USB 2.0 OTG (FS)	2 x Audio Class I2S	2 x CAN	PWM timer

Performance Line: STM32F103

72MHz CPU	Up to 1MB Flash / 96KB SRAM	2/3x12-bit ADC (1μs) TempSensor	USB–FS Device	SDIO*	I2S*	CAN	PWM timer

USB Access Line: STM32F102

48MHz CPU	Up to 128KB Flash / 16KB SRAM	1x12-bit ADC (1μs) Temp sensor	USB–FS Device

Access Line: STM32F101

36MHz CPU	Up to 1MB Flash / 80KB SRAM	1x12-bit ADC (1μs) Temp sensor

Value Line: STM32F100

24MHz CPU	Up to 512KB Flash / 32KB SRAM	1x12-bit ADC (1.2μs) Temp sensor	HDMI-CEC	PWM timer

STM32 Value line applications

Industrial

Electricity meters

Home automation

Low-end UPS

Timers
Communication peripherals

Home appliances

Home appliances, motor control, power tools

DAC
Timers
Communication peripherals

Consumer appliances

A/V receivers, TVs,
Blu-ray disk players

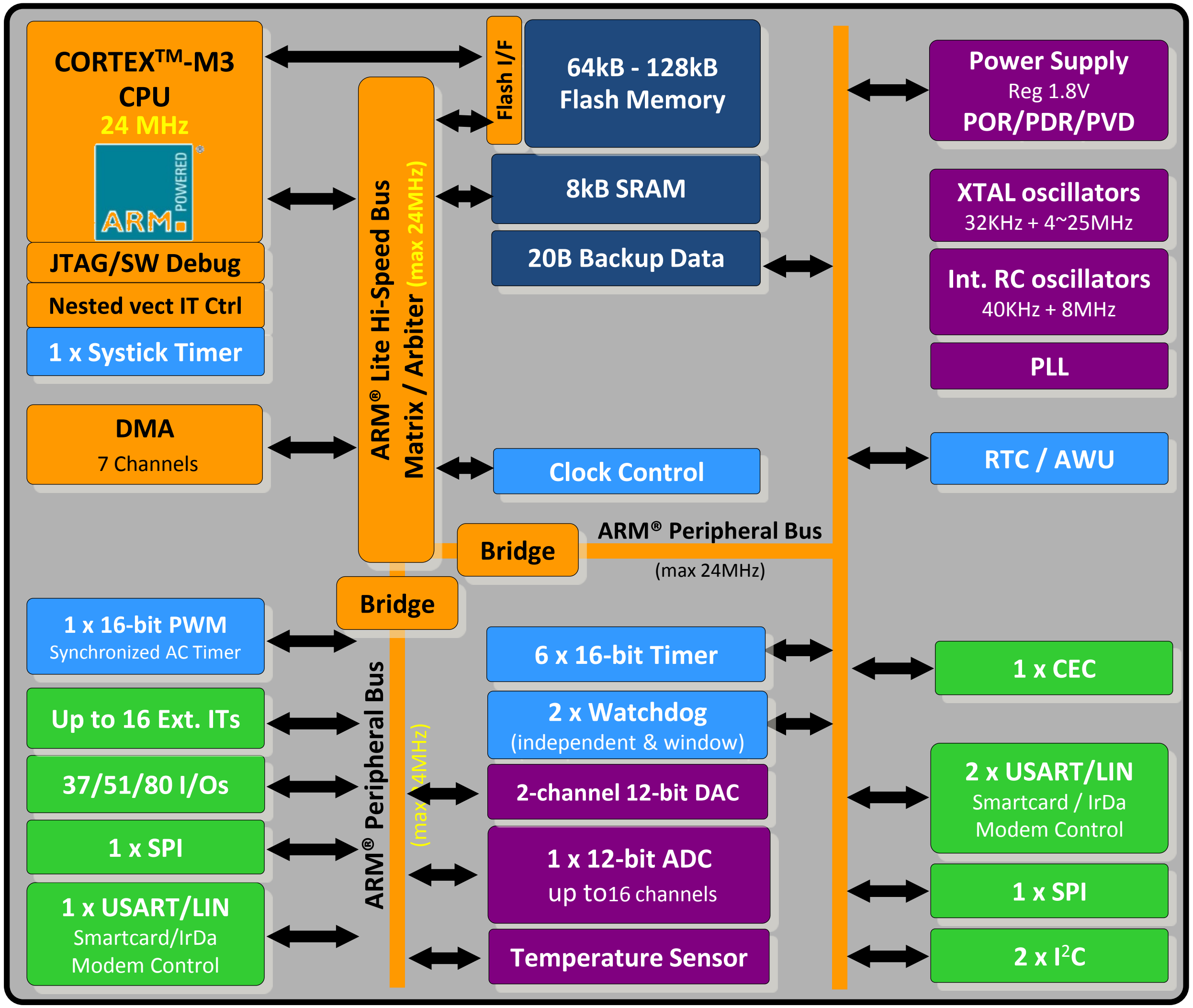
Printers

Cost competitive
Communication peripherals
CEC, DAC

Gaming

STM32 Value line 64K-128KBytes System Diagram

- **Core and operating conditions**
  - ARM® Cortex™-M3
  - 1.25 DMIPS/MHz up to 24 MHz
  - 2.0 V to 3.6 V range
  - -40 to +105 °C
- **Rich connectivity**
  - 8 communications peripherals
- **Advanced analog**
  - 12-bit1.2 µs conversion time ADC
  - Dual channel 12-bit DAC
- **Enhanced control**
  - 16-bit motor control timer
  - 6x 16-bit PWM timers
- **LQFP48, LQFP/BGA64, LQFP100**



STM32 Discovery Kit

- STM32F100RBT6B microcontroller,
  - 128 KB Flash, 8 KB RAM in 64-pin LQFP
- On-board ST-Link
  - Can be used as standalone ST-Link with SWD for programming and debugging
- On-board / Standalone configurable
- Multiple Power Supply Options
  - USB
  - External 5 V
  - External 3.3 V
- Two user LEDs
  - LD3 (green)
  - LD4 (blue)
- Two push buttons (User and Reset)
  - User / application
  - Reset
- 2.54mm (0.1”) Extension header for all MCU pins
  - Quick connection to prototyping board
  - Easy probing