Introduction

COE718: Embedded Systems Design

http://www.ee.ryerson.ca/~courses/coe718/

Dr. Gul N. Khan

http://www.ee.ryerson.ca/~gnkhan

Electrical and Computer Engineering

Ryerson University

Overview

- Course Management
- Embedded Systems and Applications
- Characteristics of Embedded Systems
- Course Main Contents

Electrical and Computer Engineering

COE718: Embedded Systems Design

http://www.ee.ryerson.ca/~courses/coe718/

Instructor: **Dr. Gul N. Khan**

Email: gnkhan@ee.ryerson.ca

URL: http://www.ee.ryerson.ca/~gnkhan/

Telephone: 416 979-5000 ext. 6084, Office: ENG448

Consultation Hours: Thursday 2:00-3:00PM or by Appointment

Department of Electrical and Computer Engineering

Ryerson University

Lectures and Projects

Half Notes

- You will need to take notes from lectures and also require text-reference books and some research articles identified by the instructor.

Labs and Project

- Aimed at concept reinforcement and practical experience.

Lectures, Labs, Projects and other support material will be available at the course website:

http://www.ee.ryerson.ca/~courses/coe718/

Assessment and Evaluation

Labs: 20% (4+4+4+4+4) + optional Bonus Lab (2% marks)
Project: 15% + optional Bonus (2-4% marks)
Midterm Exam: 25% (Thursday: 27 October 2016)
Final Exam: 40%

Course Text-Reference Books and other Material

Daniel W. Lewis, Fundamentals of Embedded Software with the ARM Cortex M3, Pearson 2013 ISBN 978-0-13-291654-7

Other Reference Books

- *T. Martin,* The Designer's Guide to the Cortex-M Processor Family: A Tutorial Approach, Elsevier, 2013, ISBN 978-0080982960
- *Marilyn Wolf,* Computer as Components: Principles of Embedded Computing System Design, 3rd Edition Morgan Kaufman - Elsevier Publishers 2013, ISBN 978-0-12-388436-7
- Embedded Core Design with FPGAs by *Z. Navabi,* McGraw Hill 2007, ISBN 978-0-07-147481-8 or ISBN 0-07-147481-1
- *Alan Burns and Andy Wellings*, Real-time Systems & Programming Languages, Addison-Wesley 2001, ISBN 0 201 72988 1

Some Articles, Embedded Processors and other Data Sheets are available at the Course Website: http://www.ee.ryerson.ca/~courses/coe718/

Lecture's Material

- **Introduction to Embedded Systems**
  Text by D. Lewis: Chapter-1, Professional Journal and Magazine Articles
- **Embedded CPUs and Cores**
  ARM Cortex M3 Core Architecture: Chapter 5 of Text by D. Lewis.
- **ARM Cotex M3 ISA and Programming**
  Text by D. Lewis: part of Chapters 6 and 7.
- **Real-time Operating Systems**
  Concurrency, pre-emptive and non-pre-emptive Scheduling. Keil RTX operating system environment: Chapter 9 of Text by D. Lewis
- **Real-time Scheduling**
  FPS and EDF Real-time Scheduling techniques, RTX real-time operating environment. Chapter 10 of Text by D. Lewis and Part of Chap. 7-10, 13 of Text by Burns & Wellings
- **Introduction to Hardware Software Codesign**
  Embedded System Modeling and Co-specification
  Text by M. Wolf, part of Chapter 7; Journal Articles, Manuals
- **Embedded System Partitioning, Accelerator based Embedded Systems**
  Text by M. Wolf: part of Chapters 7 and 8; Review Journal Articles
- **Fault-tolerant Embedded Systems -** Review Articles
- **Embedded System Case Studies –** If time permits

Introduction

- What are Embedded Systems?
- Challenges in Embedded Computing System Design
- Design Methodologies

Main Aim of the Course

- To introduce the basics of embedded system design.
    - Software and hardware components of an embedded system
- Hardware Software Codesign
- To understand real-time operating systems
- Embedded Computer Architecture

Ideally Student should have the knowledge of:

- Basics of Programming C or C++ and Computer Architectures
- Operating Systems

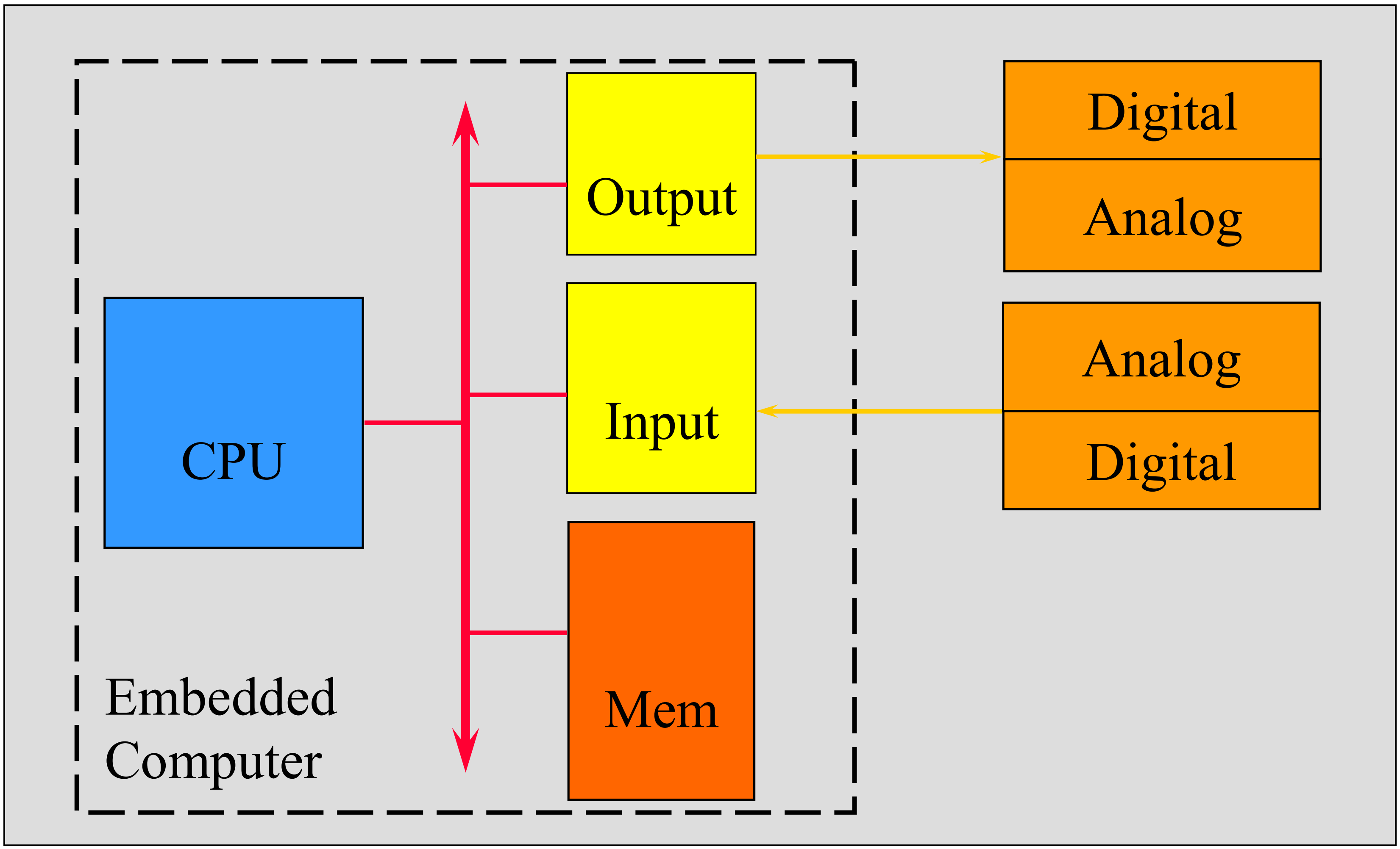
What is an Embedded (Real-time) System?

Most embedded systems are also real-time systems.

- An embedded system is an information processing system that responds to externally generated input stimuli within a finite and specified period.
  - The correctness depends not only on the logical result but also the time it was delivered
  - Failure to respond is as bad as the wrong response!
- Embedded system: any device that includes a programmable computer but is not itself a general-purpose computer.
- Take advantage of application characteristics to optimize the design:

Don't need all the general-purpose bells and whistles.

Embedding a Computer



Embedded Systems

- Electronic devices that incorporate a computer (microprocessor/micro-controller) in their implementation.
- A computer is used in such devices primarily as a means to simplify the system design and to provide flexibility.
- Often the user of the device is not even aware that a computer is present.
- Embedded Systems are every where
  - Embedded processors account for more than 90% of worldwide microprocessor production.
  - Embedded *vs.* desktop = 100:1
  - 99% of all processors are for the embedded systems market.
  - Number of embedded processors in a typical home is estimated at 50-60.

  (a luxury automobile has more than 50 embedded processors)

Embedded Real Time Systems

- Real-time systems process events.
- Events occurring on external inputs cause other events to occur as outputs.
- Minimizing response time is usually a primary objective, or otherwise the entire system may fail to operate properly.

Types of Embedded Real Time System

- Hard real-time — e.g. Flight control systems.
- Soft real-time — e.g. Data acquisition system.
- Real real-time — e.g. Missile guidance system.
- Firm real-time

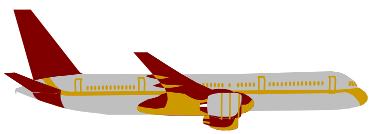
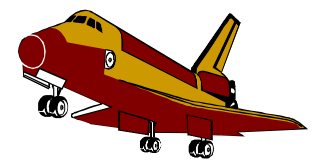
Types of Real Time System

- **Hard real-time —** systems where it is absolutely imperative that the responses occur within the required deadline.
  For example: Flight control systems.
- **Soft real-time —** systems where deadlines are important but which will still function correctly if deadlines are occasionally missed. For example: Data acquisition system.
- **Real real-time —** systems which are hard real-time and their response time is very short.
  For example: Missile guidance system.
- **Firm real-time —** systems which are soft real-time but in which there is no benefit from late delivery of service.

A single system may have all hard, soft and real real-time subsystems.

In reality many systems will have a cost function associated with missing each deadline.

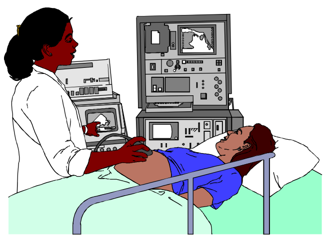
Embedded System Applications

Aerospace Spacecrafts 	Navigation systems, automatic landing systems, flight attitude controls, engine controls, space exploration (e.g., the Mars Pathfinder). 
Automotive	Fuel injection control, passenger environmental controls, anti-lock braking, air bag controls, GPS mapping.
Children's Toys	Nintendo's, Mattel's "My Interactive Pooh", Tiger Electronics' "Furby".
Communi-cations	Satellites; network routers, switches, hubs.

Embedded System Applications

Computer Peripherals	Printers, scanners, keyboards, displays, modems, hard disk drives, CD-ROM drives.
Home	Dishwashers, microwave ovens, HDTV, sound systems, fire/security alarm systems, lawn sprinkler controls, thermostats, cameras, clock digital radios.
Industrial	Elevator controls, surveillance systems, robots.
Instrumen-tation	Data collection, oscilloscopes, signal generators, signal analyzers, power supplies.

Embedded System Applications

Medical 	Imaging systems (e.g., XRAY, MRI, and ultrasound), patient monitors, and heart pacers.
Office Automation	Copiers/ FAX machines, smart telephones, and cash registers.
Personal	Tablets, ipads, cell phones, smart-watches

Early History of Embedded Systems

- First microprocessor was Intel 4004 in early 1970's.
- HP-35 calculator used several chips to implement a microprocessor in 1972.
- Automobiles used microprocessor-based engine controllers starting in 1970's.
    - Control fuel/air mixture, engine timing, etc.
    - Multiple modes of operation: warm-up, cruise, etc.
    - Provides lower emissions, better fuel efficiency.

- Microcontroller: includes I/O devices, on-board memory.
- Digital signal processor (DSP): microprocessor optimized for digital signal processing.

Typical embedded word sizes: 8-bit, 16-bit, and 32-bit.

Automotive embedded systems

Today's high-end automobile may have 100 microprocessors:

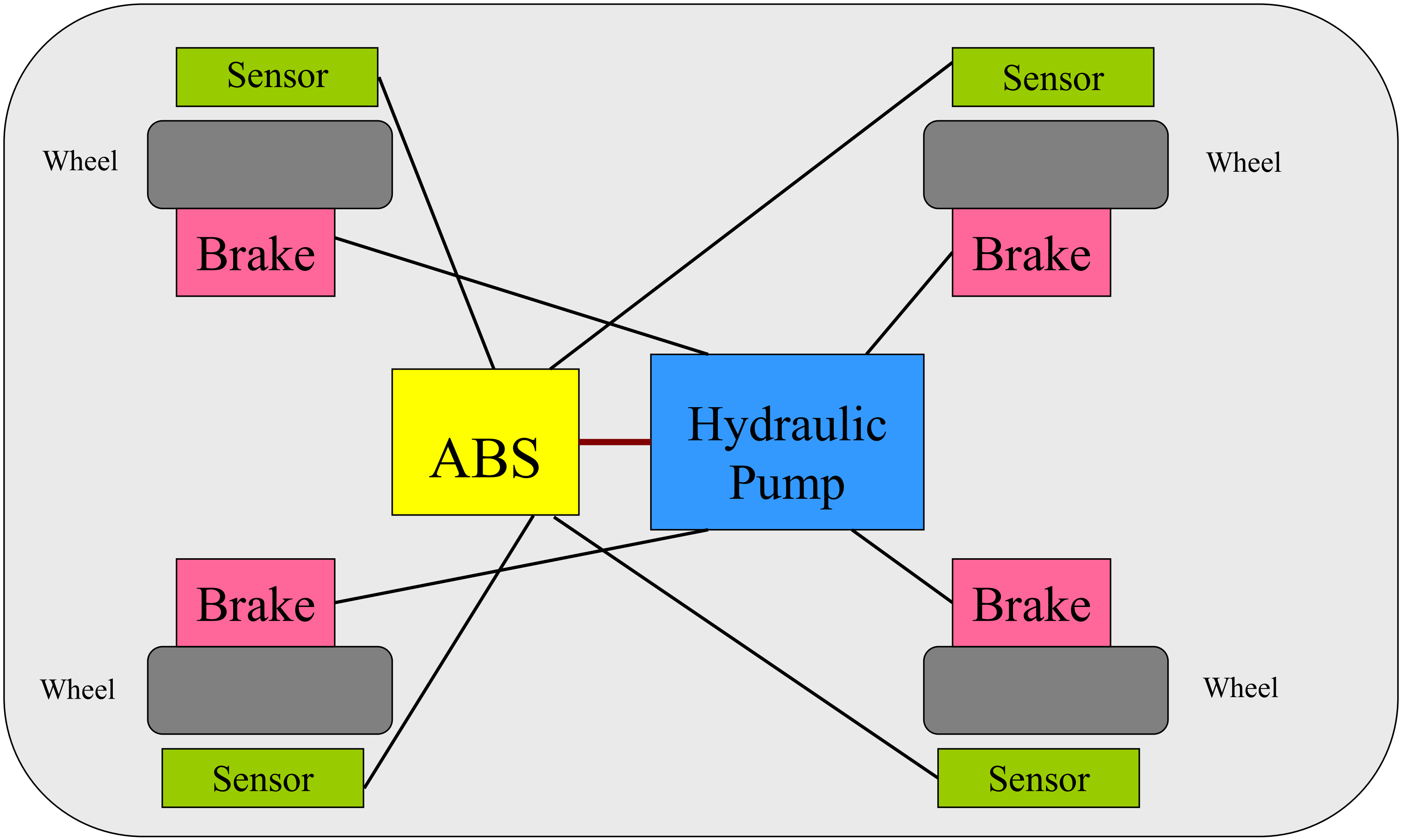
- 4-bit microcontroller checks seat belt
- Microcontrollers run dashboard devices
- 16/32-bit microprocessor controls engine

BMW brake and stability control system

- Anti-lock brake system (ABS): Pumps brakes to reduce skidding.
- Automatic Stability Control (ASC+T): Controls engine to improve stability.
- ABS and ASC+T communicate.

  ABS was introduced first---needed to interface to existing ABS module.

Anti-lock Brake System (ABS)



Embedded System Applications

Programmable Digital Thermostat

Uses: 4-bit Microprocessor

Vending Machine.

Vendo V-MAX 720

Uses
8-bit Motorola 68HC11
Microcontroller

Vitality's GlowCap

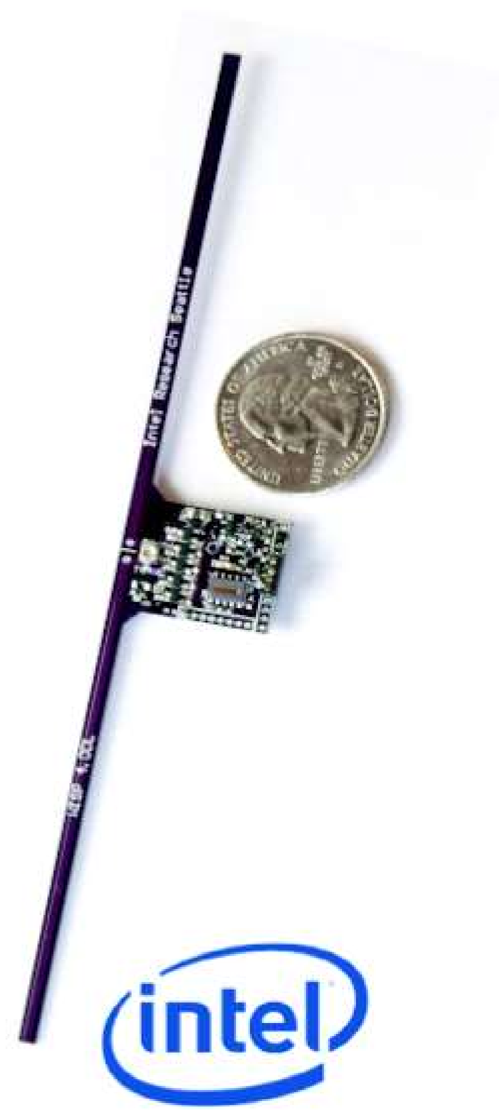

Vitality's GlowCap

- GlowCap has a tiny Amtel 8-bit picoPower AVR Processor
- Help People to take their medication on-time.
- Sense when the bottle is opened.
- Connect to Vitality server and transmit information wirelessly.

Intel WISP RFID

TIMSP430F1232: Low Power Micro-controller

- 16-bit CPU
- 8 KBytes of flash memory
- 256 bytes of RAM
- 10-bit –ADC with 200 kilo-samples/second
- CPU can run at 8MHz with 3.3V supply voltage

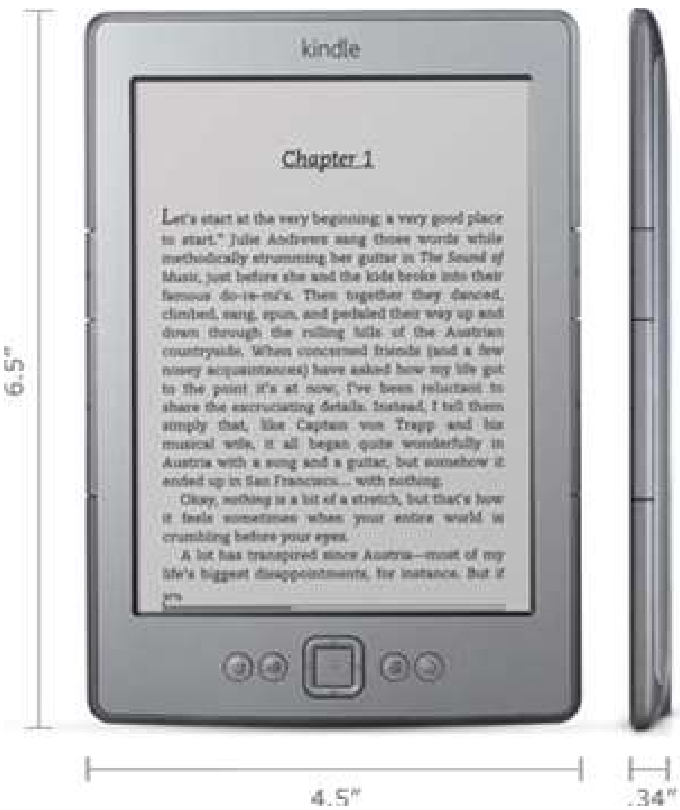


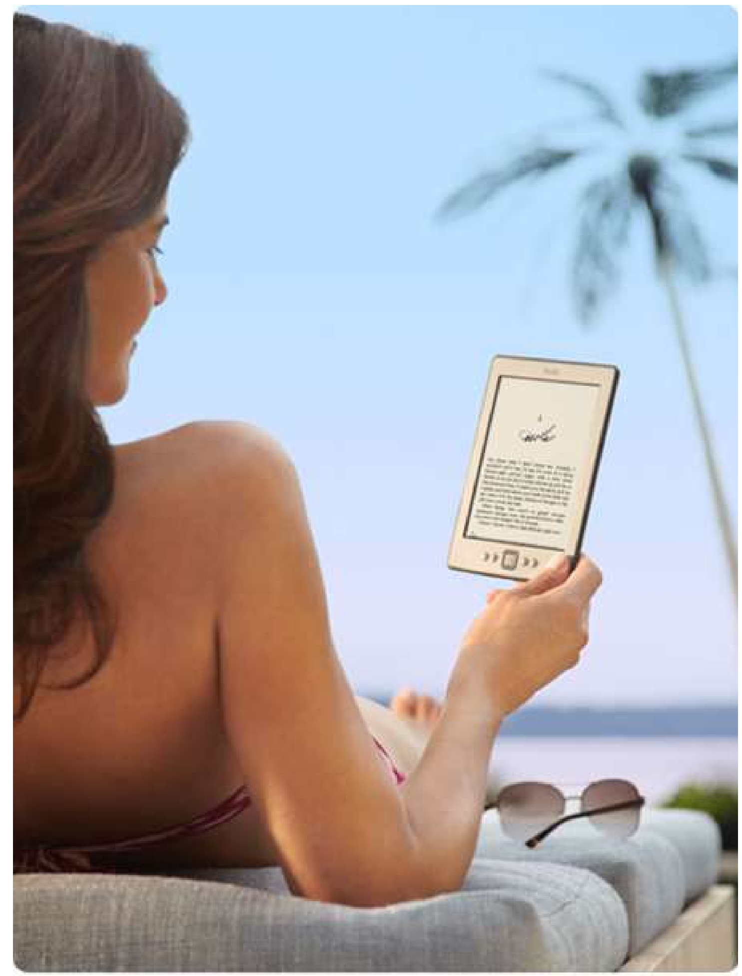
Sonicare Diamond Clean Toothbrush

Toothbrush
8-bit PIC
microprocessor

Amazon Kindel-2

Uses 32-bit ARM processor

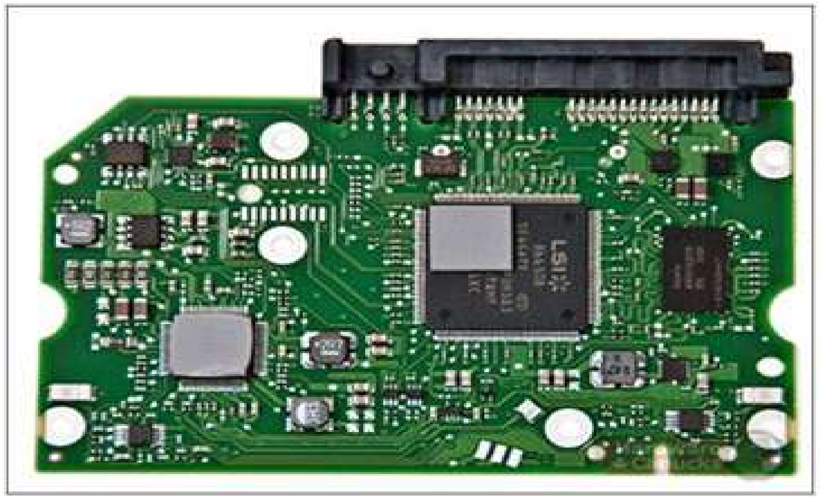




Reads like real paper — no glare, even in bright sunlight

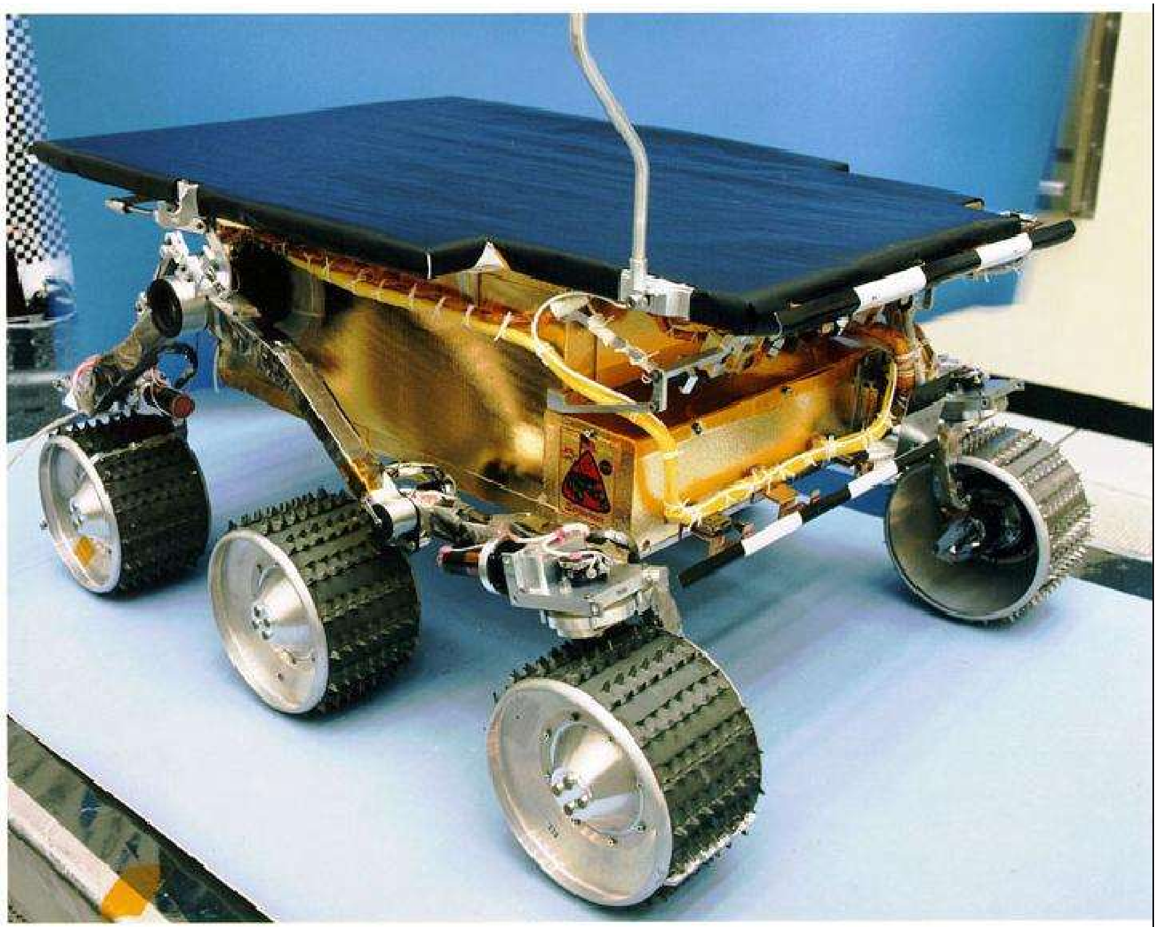
Seagate Barracuda Disk Drive

- Employ two ARM Cortex-R4 Processors
- One for Servo control and 2nd for Command and Data Flow



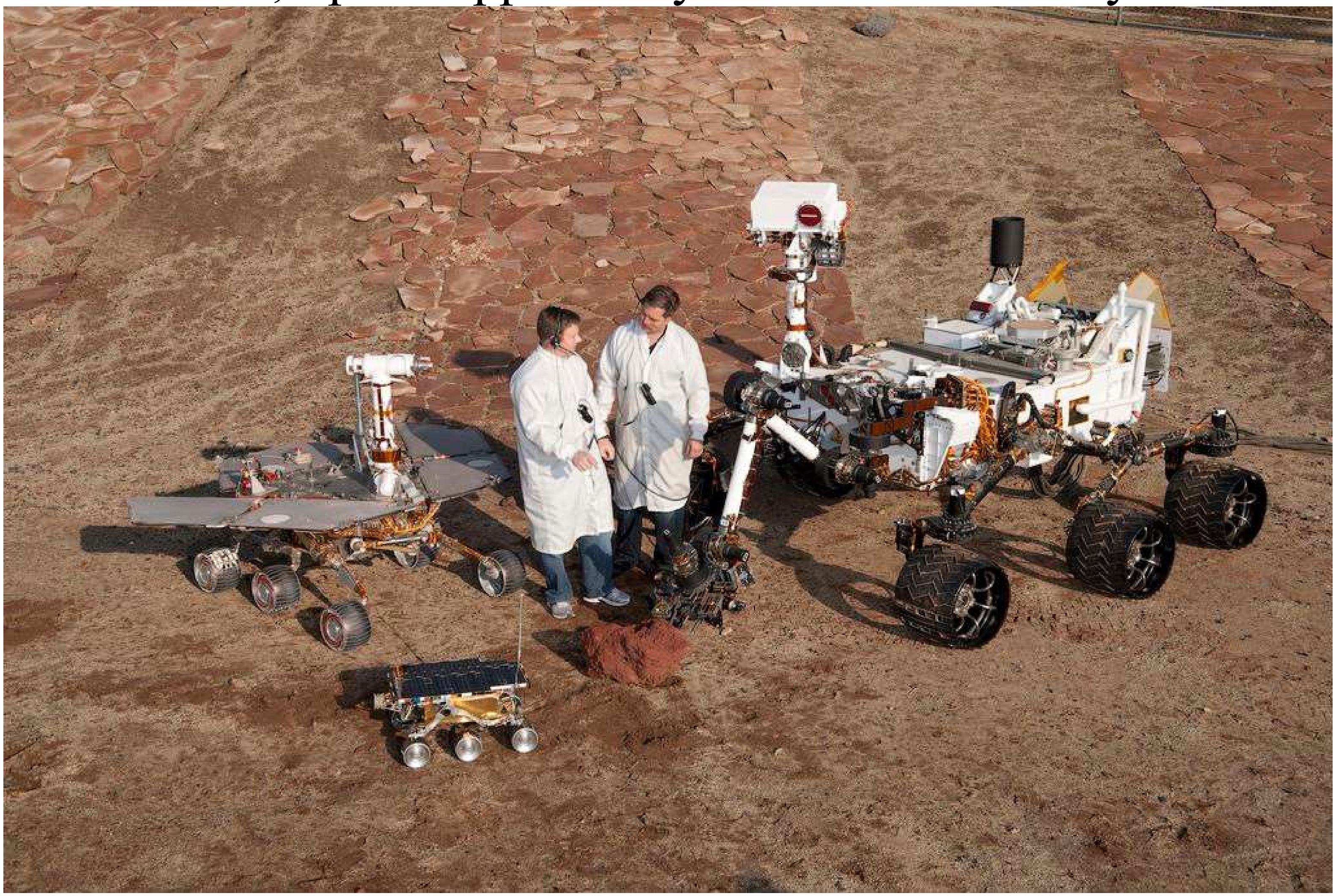
NASA's Mars Rover

PathFinder: Uses Intel 80C85 8-bit Microprocessor



MAR’s Rovers

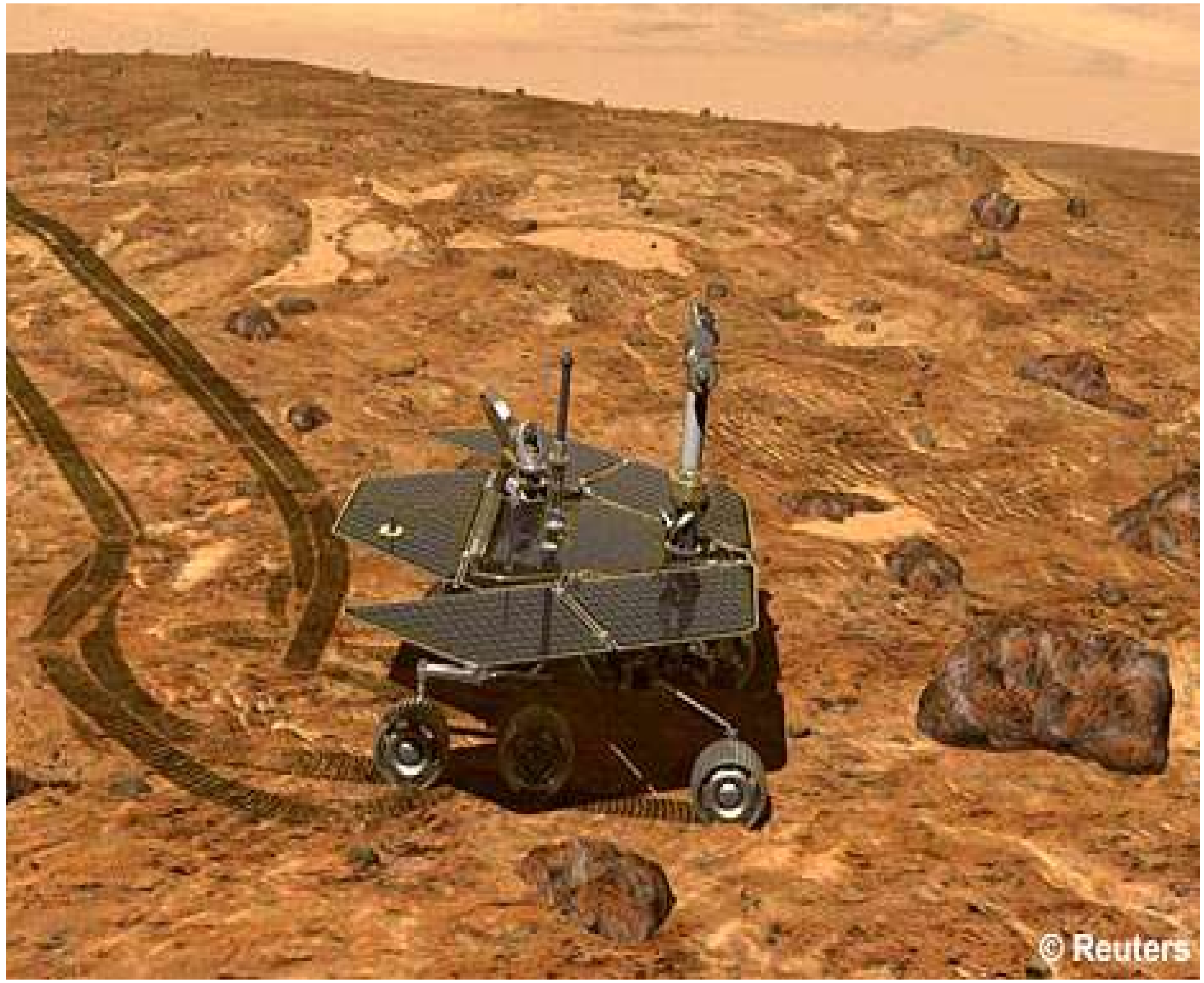
Pathfinder-1997, Spirit/Opportunity-2003 and Curiosity-2012



2003 MAR'S Rover

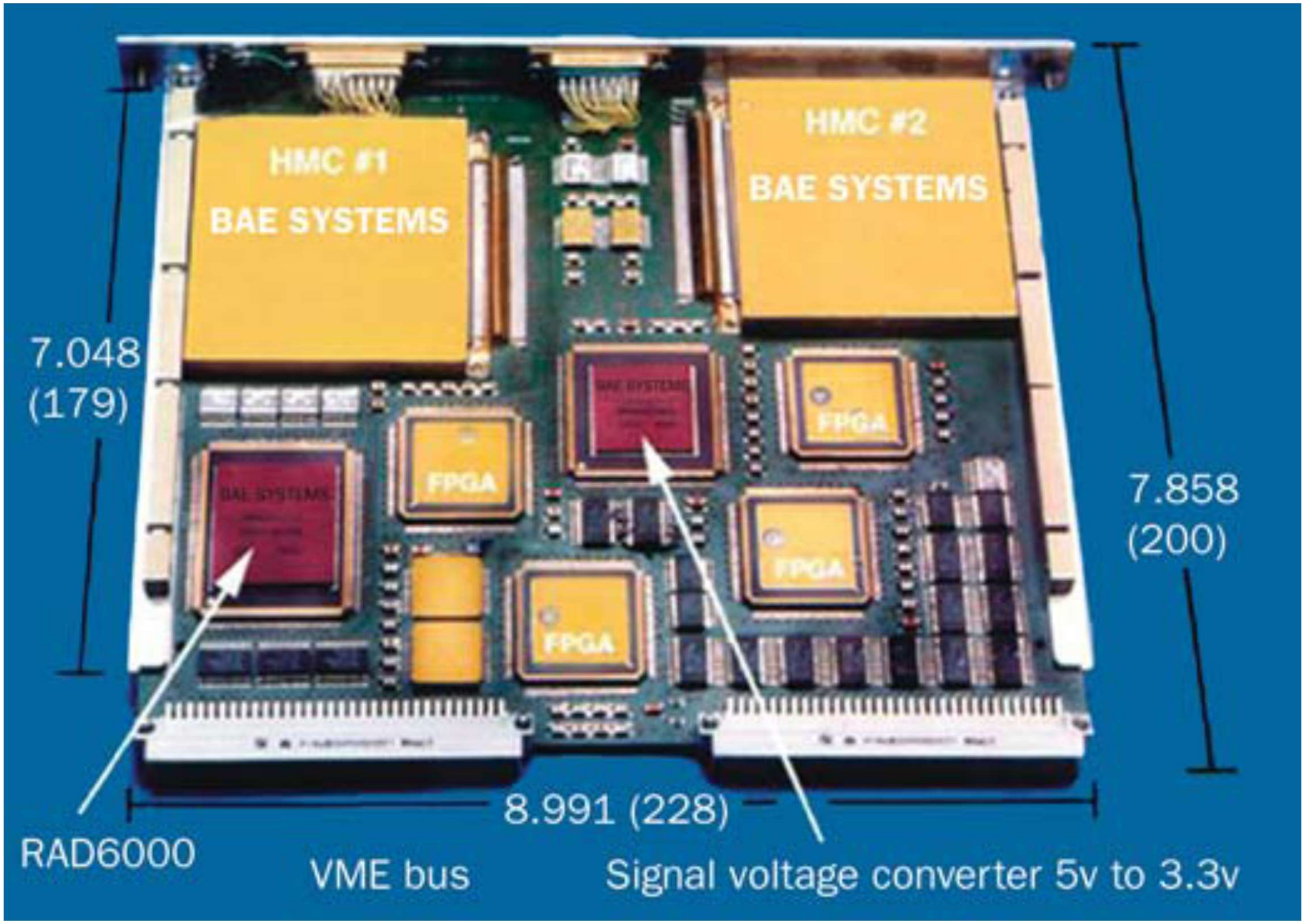
Spirit/Opportunity

- Use BAE Systems RAD6000 32-bit RISC Processor
- Radiation hardened IBM POWER series 6000 CPU
- Employ VxWorks: Embedded Real-time Operating System from Wind River.



Mars Rover RAD6000 Flight Computer,

FPGA-based



Comparison of embedded Computer Systems for Mars Rovers

Rover (mission,year)	CPU	RAM	Storage	Operating system
Sojourner Rover (Pathfinder 1997)	2MHz Intel 80C85	512KB	176 KB	Custom cyclic executive
Pathfinder Lander (1997) (Base station for *Sojourner* rover)	20MHz IBM RAD6000	128 MB	6 MB (EEPROM)	VxWorks (multitasking)
Spirit and ***Opportunity*** (Mars Exploration Rover, 2004)	20 MHz IBM RAD6000	128 MB	256 MB	VxWorks (multitasking)
Curiosity (Mars Science Laboratory, 2011)	200 MHz IBM RAD750	256 MB	2GB	VxWorks (multitasking)

Non-functional Requirements

- Many embedded systems are mass-market items that must have low manufacturing costs.
- Limited memory, microprocessor power, etc.
- Power consumption is critical in battery-powered devices.
- Excessive power consumption increases system cost even in wall-powered devices.

Power

- Custom logic is a clear winner for low power devices.
- Modern microprocessors offer features to help control power consumption.
- Software design techniques can help reduce power consumption.

Where are we heading?

- **Embedded CPUs and Cores**
  ARM Cortex M3 Core ISA and Architecture, Chapter 5
- **ARM Cotex M3 ISA and Programming**
  Part of Chapters 6 and 7.
- **Real-time Operating Systems**
  Concurrency, pre-emptive and non-pre-emptive Scheduling.
  Keil RTX operating system environment: Chapter 9
- **Real-time Scheduling**
  FPS and EDF Real-time Scheduling techniques, RTX real-time operating environment manual and details. Chapter 10
- **Introduction to Hardware Software Codesign**
  Embedded System Modeling and Co-specification
  Text by M. Wolf, part of Chapter 7; Journal Articles, etc.
- **System Partitioning, Accelerator based Embedded Systems**
  Text by M. Wolf: part of Chapters 7 and 8; Review Journal Articles
- **Fault-tolerant Embedded Systems -** Review Articles
- **Case Studies –** if time permits