Introduction to
“Embedded System Design”

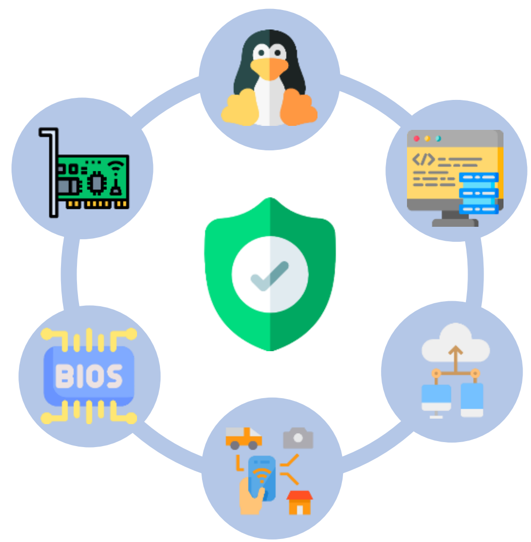
Lecture 1

Yeongpil Cho

Hanyang University

About me

- 조영필 (Yeongpil Cho)
  - A system security researcher
  - Designing new SW/HW techniques for better security
    - OS kernels
    - Hypervisor
    - Firmware
    - Applications
    - etc.



Course information

- Goal
  - We are going to discuss hardware and software architectures of embedded systems
- Class time & location
  - Theory classes
    - Thu. 1-3 pm @ ITBT 813
  - Practice classes
    - Fri. 9-11 am @ ITBT 609
- Course materials
  - (main) Lecture notes
  - (auxiliary) Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C: Third Edition

Course information

- Grading policy
  - Midterm: 20%
  - Final: 20%
  - Lab assignments: 20%
  - Term project: 30%
  - Attentance:10%
- Office hour
  - Make an appointment at any time
    - ypcho@hanyang.ac.kr
  - Location: IT/BT Bld. 1208
- TA
  - Taewook Kim (김태욱)
    - qkenr7895@hanyang.ac.kr
  - Seokhyun Hong (홍석현)
    - ghazard8572@hanyang.ac.kr

Course outline

- We will explore
  - ARM's Microcontrollers
  - ARM's Application Processors
- Theory Lectures
  - Deal with only microcontrollers.
- Practice Lectures
  - Deal with both the application processors and microcontrollers.
  - Developing a simple OS on application processors.
  - Developing a maze solving robot on microcontrollers.

Tentative Syllabus

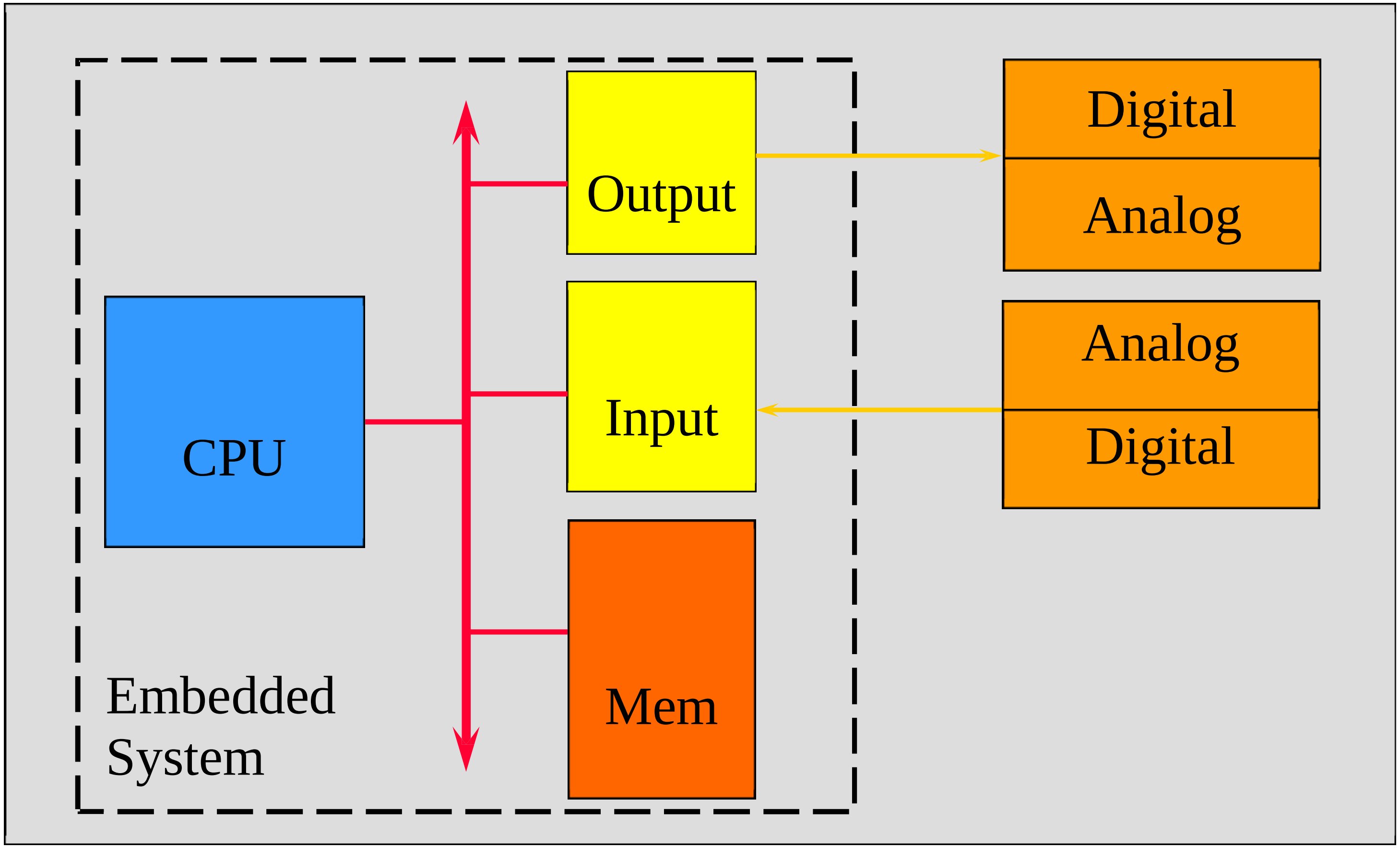
Week	Date	Lecture
1	3/3	Course Overview
2	3/10	Basics in Computer Architecture
3	3/17	Introduction to ARM Architecture & Overview of Cortex-M processors
4	3/24	Memory system in Cortex-M processors
5	3/31	ARM Assembly Language
6	4/7	ARM Assembly Language
7	4/14	Midterm Exam
8	4/21	ARM Assembly Language
9	4/28	Cortex-M's subroutine mechanism
10	5/5	Cortex-M's interrupt mechanism **(make-up class will be scheduled)**
11	5/12	Cortex-M's interrupt mechanism
12	5/19	Cortex-M's timer and GPIO
13	5/26	Final Exam
14	6/2	Term Project
15	6/9	Term Project
16	6/16	Term Project (Optional)

What is Embedded systems?

Embedded system

- Any device that includes a computer but is not itself a general-purpose computer.
    - Of course, it can handle general purpose computations, but …
- Application specific
    - The design is specialized and optimized for specific application.
    - Don’t need all the general-purpose bells and whistles (additional functions).

Embedding a Computer



Early History of Embedded Systems

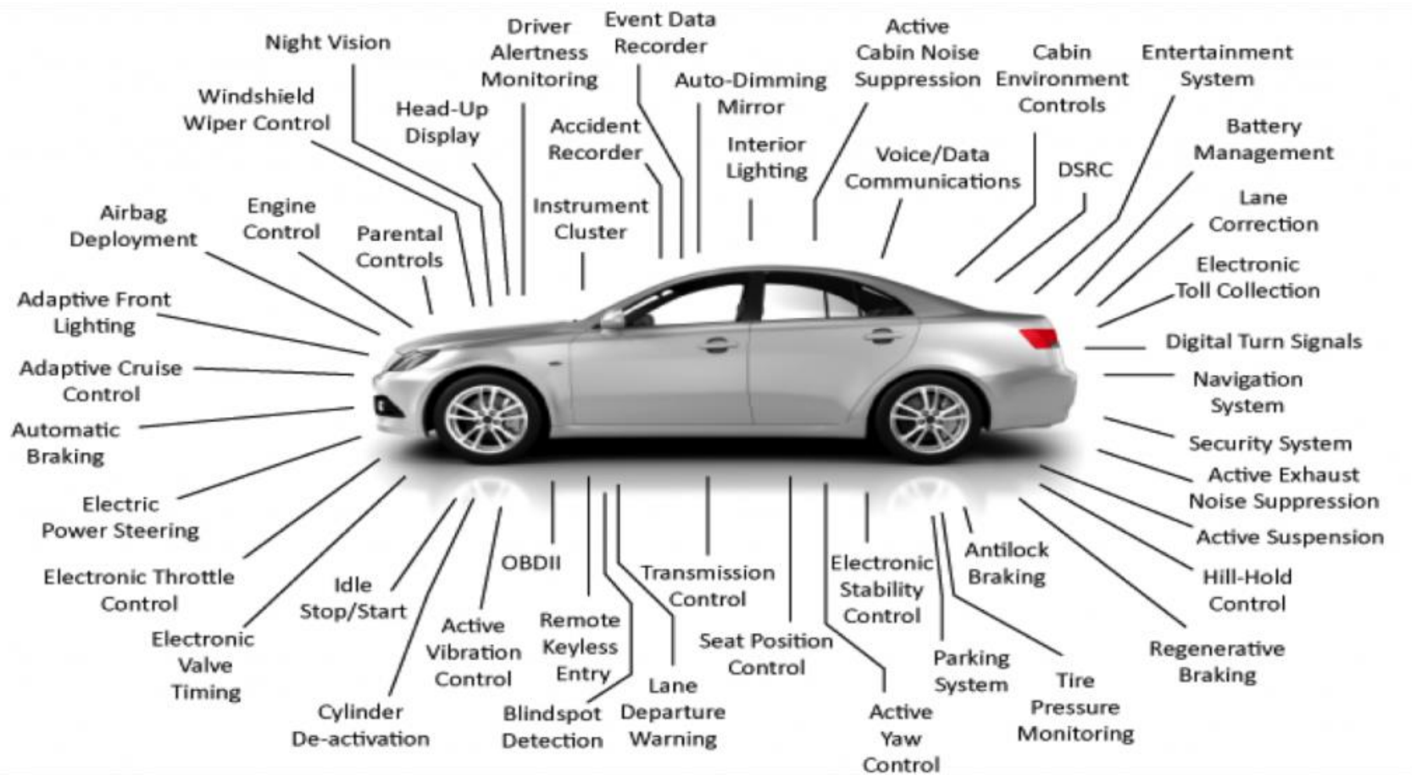
- First microprocessors are
  - Intel 4004 (4-bit CPU) at 1971.
  - Intel 8008 (8-bit CPU) at 1972.
- HP-35 was the first portable calculator which used microprocessors in 1972.
- Automobiles used microprocessor-based engine controllers starting in 1970’s.

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.

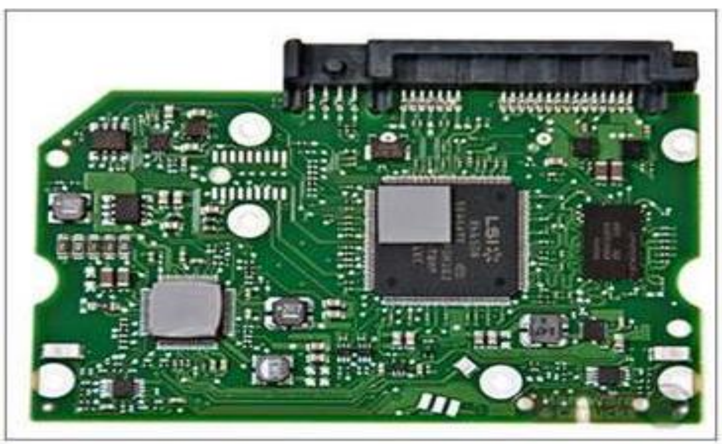
Automotive embedded systems

- Today's high-end automobile may have 100 microprocessors:
  - 4-bit microcontroller checks seat belt.
  - Microcontrollers run dashboard devices.
  - 16/32-bit microprocessor controls an engine.



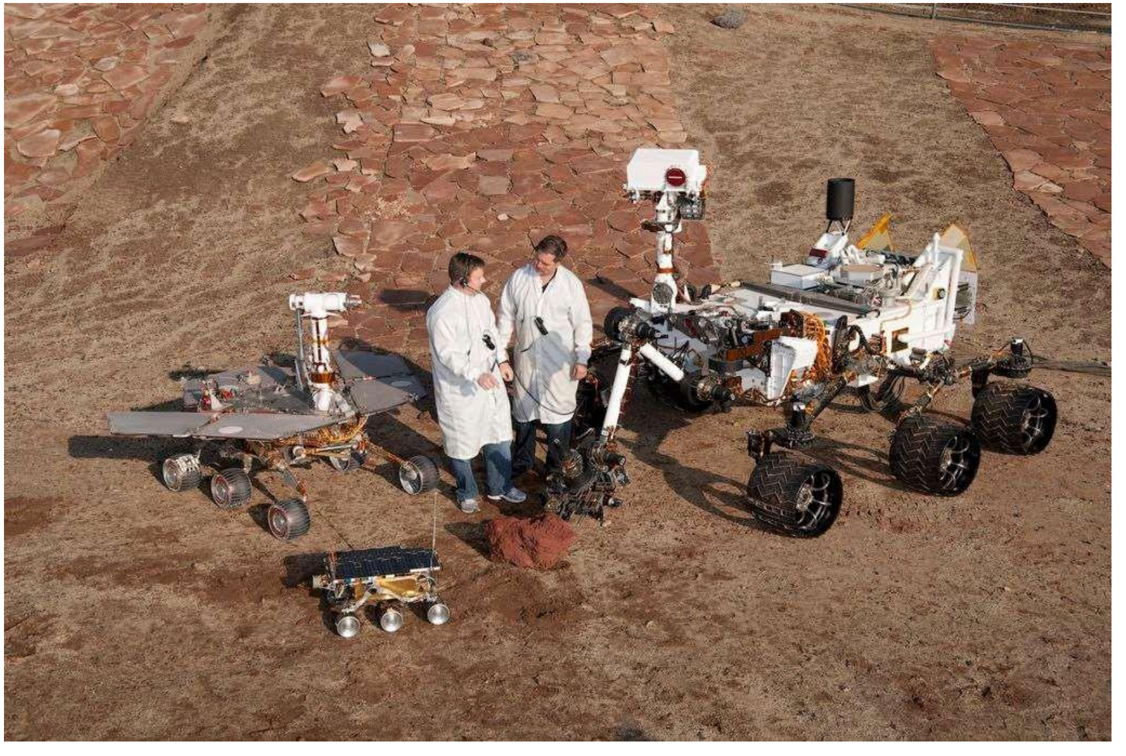
Seagate Barracuda Disk Drive

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



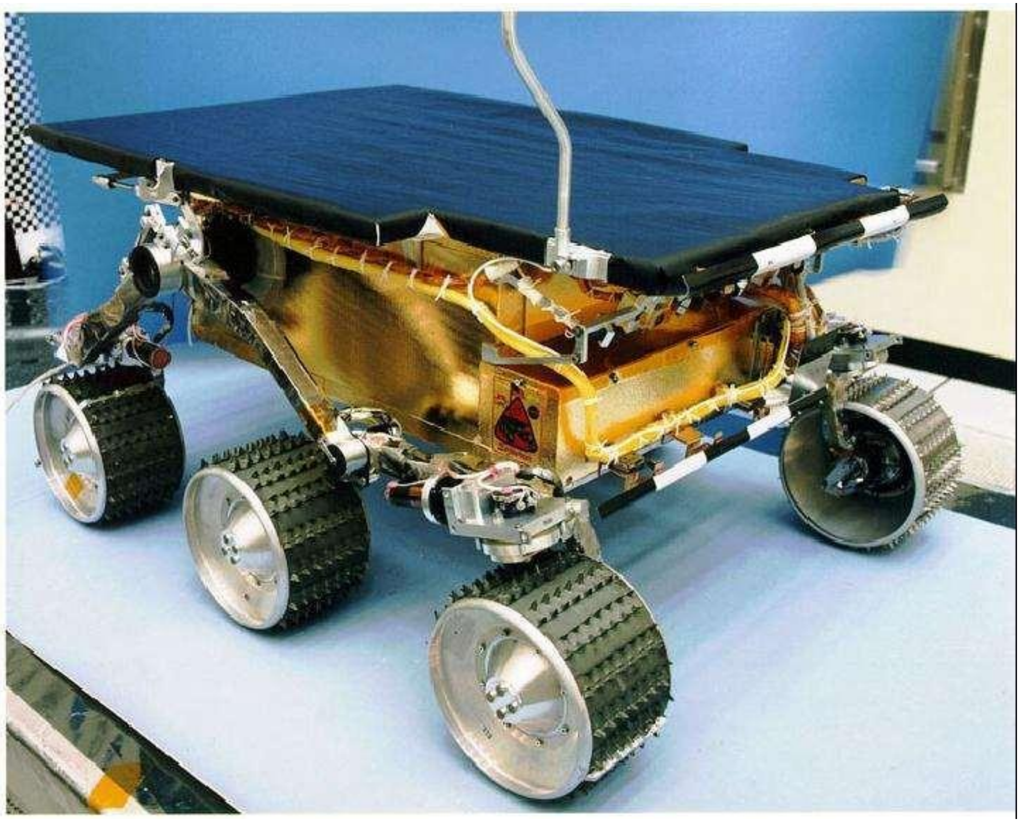
NASA's Mars Rover

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



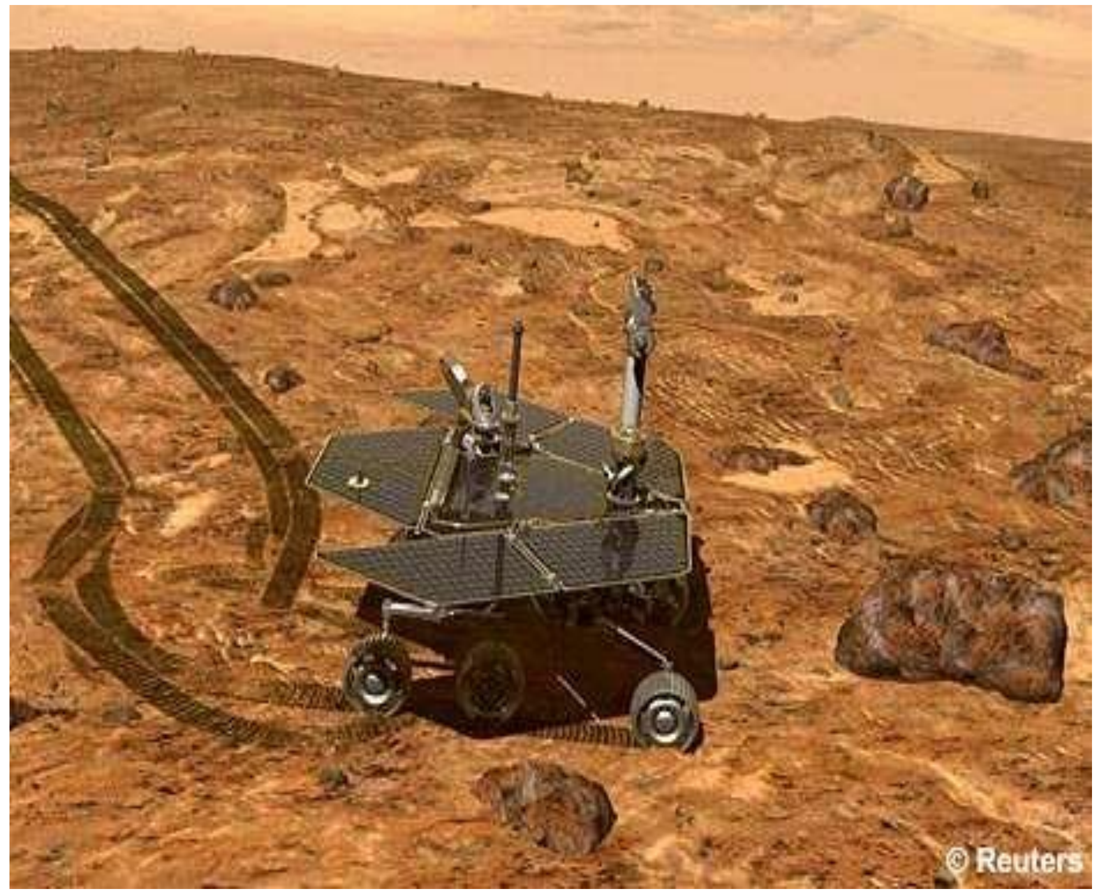
PathFinder

- Uses Intel 80C85 8-bit Microprocessor



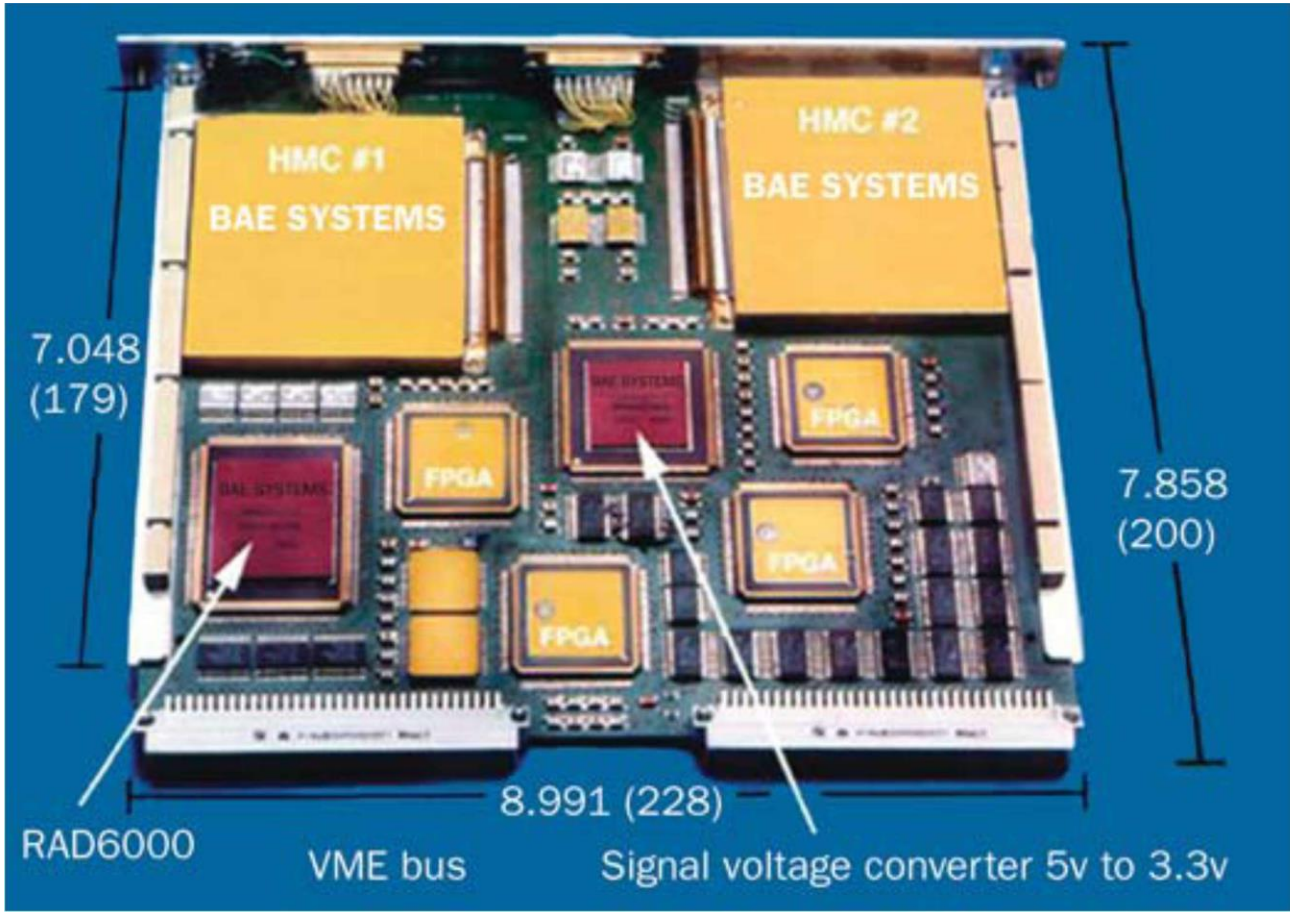
Spirit/Opportunity

- 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.



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	128MB	6 MB (EEPROM)	VxWorks (multitasking)
Spirit and Opportunity (Mars Exploration Rover, 2004)	20 MHz IBM RAD6000	128MB	256MB	VxWorks (multitasking)
Curiosity (Mars Science Laboratory, 2011)	200 MHz IBM RAD750	256MB	2GB	VxWorks (multitasking)

Embedded System Design Constraints

- **Cost**
  - Competitive markets penalize products which don’t deliver adequate value for the cost
- **Performance**
  - Perform required operations (throughput)
  - Meet real-time deadlines (latency)
- **Size and weight limits**
  - Mobile (aviation, automotive) and portable (e.g. handheld) systems
- **Power and energy limits**
  - Battery capacity
  - Cooling limits
- **Environment**
  - Temperatures may range from -40°C to 125°C, or even more

Power Considerations

- Custom logic typical in low power devices.
  - Low-power electronic circuit design methods
- Modern microprocessors offer features to help control power consumption.
  - Turn off unnecessary logic/modules
  - Reduce clock rates (CMOS)
  - Use “sleep modes”
- Software design techniques can also help reduce power consumption.
  - Reduce memory accesses
  - Reduce external communication

Safe, secure systems

- **Security**: system's ability to prevent malicious attacks.
  - Traditional security is oriented to IT and data security.
  - Insecure embedded computers can create unsafe cyber-physical systems.
  - Internet of Things presents special security challenges!
- **Safety**: system's ability to maintain stable state without crashes, harmful releases of energy, etc.
  - We need to combine safety and security:
    - Identify security breaches that compromise safety.
  - Safety and security can't be bolted on (added afterward)
    --- they must be baked in (included from the beginning).

Computing units for embedded systems

- ASIC (Application Specific Integrated Circuit)
- FPGA (Field Programmable Gate Array)

	Implementation	Design Cost	Unit Cost	Upgrades & Bug Fixes	Size	Weight	Power	System Speed
Dedicated Hardware	Discrete Logic	low	mid	hard	large	high	?	very fast
	ASIC	high ($500K/mask set)	very low	hard	tiny - 1 die	very low	low	extremely fast
	Programmable logic – FPGA, PLD	low	mid	easy	small	low	medium to high	very fast
Software Running on Generic Hardware	Microprocessor + memory + peripherals	low to mid	mid	easy	small to med.	low to moderate	medium	moderate
	Microcontroller (int. memory & peripherals)	low	mid to low	easy	small	low	medium	slow to moderate
	Embedded PC	low	high	easy	medium	moderate to high	medium to high	fast

Embedded Real Time Systems

- Real-time systems process (response to) events.
- Minimizing response time is usually a primary objective, or otherwise the entire system may fail to operate properly.

- Three types of Embedded Real Time Systems
  - Hard real-time
  - Soft real-time
  - 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, Nuclear plant control. Pacemaker
- **Soft real-time** — systems where deadlines are important, but which will still function correctly if deadlines are occasionally missed.
  - For example: Data acquisition system, Mobile communication.
- **Firm real-time** — systems which are soft real-time but in which there is no benefit from late delivery of service.
  - For example: Video playing, Voice calls.

- A single system may have all hard, soft and firm real-time subsystems.
- In reality, systems will have a cost function associated with missing each deadline.