

ECE 1175
Embedded Systems Design
Design Methodology I

Wei Gao

System Characteristics

- Non-functional requirements
 - 1. Real-time
 - 2. Low power
 - 3. Low cost
- Performance constraints
- Marketing issues
 - Short development cycles
 - Small yet efficient teams

1. Real-Time

- System works properly only if it meets its timing constraints
- Timing constraints
 - Deadline: complete a task within D millisec
 - E.g., ABS, GPS
 - Throughput / rate: complete N tasks per sec
 - E.g., portable DVD player

1. Real-Time

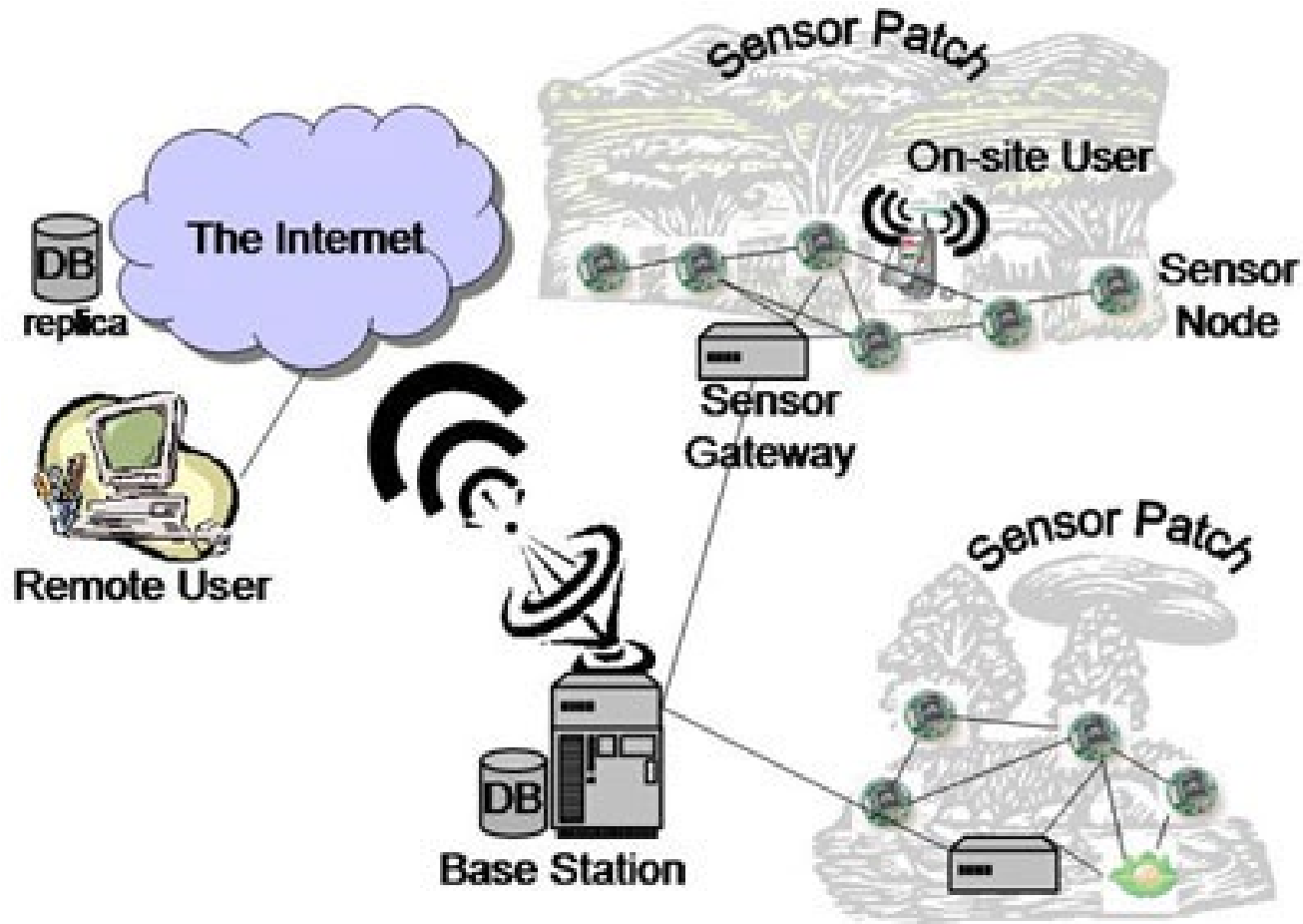
- **Hard real time:** violating timing constraints causes failure
 - Anti-lock Brake System (ABS)
 - CD burner
 - Software modem
- **Soft real time:** missing deadline results in degraded performance
 - Video
 - GPS map
 - Audio (MP3 player)?

2. Low Power

- Battery-powered devices
 - Cell phone, iPod, wireless sensors...
- Requirement on the battery life
 - Widely ranging from hours to months/years
- Wall-powered devices: excessive power consumption increases system cost
 - Cooling
 - Energy bills

Great Duck Island

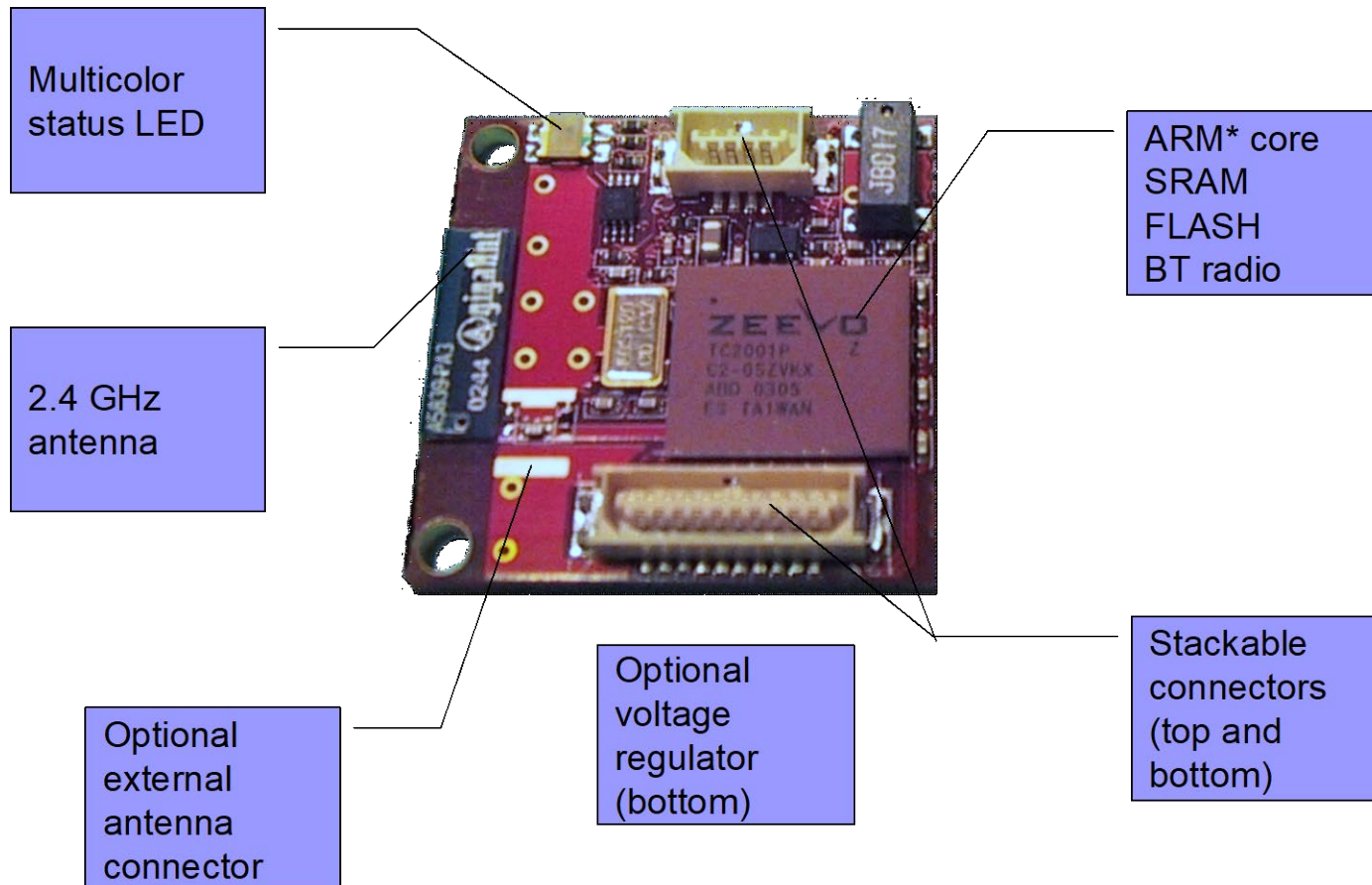
- Require 9 month lifetime on batteries!



3. Low Cost

Source: Lama Nachman, presentation at
TinyOS Technology Exchange

- Constraints on memory size, processor speed, I/O interfaces...
 - Example: Intel iMote²



iMote² Price Sheet

Source: Lama Nachman, presentation at
TinyOS Technology Exchange

Item	Description	Cost
Intel PXA 270/271/273 processor	Discrete / 32M F+S / 32M F	\$17.75 / \$32.44 / \$23
PMIC	Dialog power management IC	\$3.75
CC2420	ChipCon 802.15.4 Radio	\$3.6
Crystals	13 MHz, 16 MHz, 32KHz	~\$2.8
Antenna	Giga Ant surface mount	\$1.5
LED + Driver	Agilent	\$1
Connectors	Basic (2) + Advanced (2)	\$1 + \$1.6
Passives		~ \$9
Fab + Assembly	Rough estimate from iMote ¹	~ \$10
Total	With PXA 270 / 271 / 273	\$52 + F / \$67 / \$57

- Prices above assume 1K units, except for PXA and PMIC (special Intel pricing)
- PXA 270 configuration will need an external flash + bringing out addr/data bus

iMote²: Cost Consideration

*Source: Lama Nachman, presentation at
TinyOS Technology Exchange*

- No A/D on mote board → Reduce cost
 - Digital sensors don't need it
 - Applications have different A/D requirements (number of channels, sampling rate, filtering...)
 - A/D will be integrated into the sensor board
- SDIO (Secure Digital I/O)
 - Connector is too big (30x30 mm) and costly (\$1.74) to be included on Imote² board
 - Pins will be exposed through basic connector

Performance Constraints

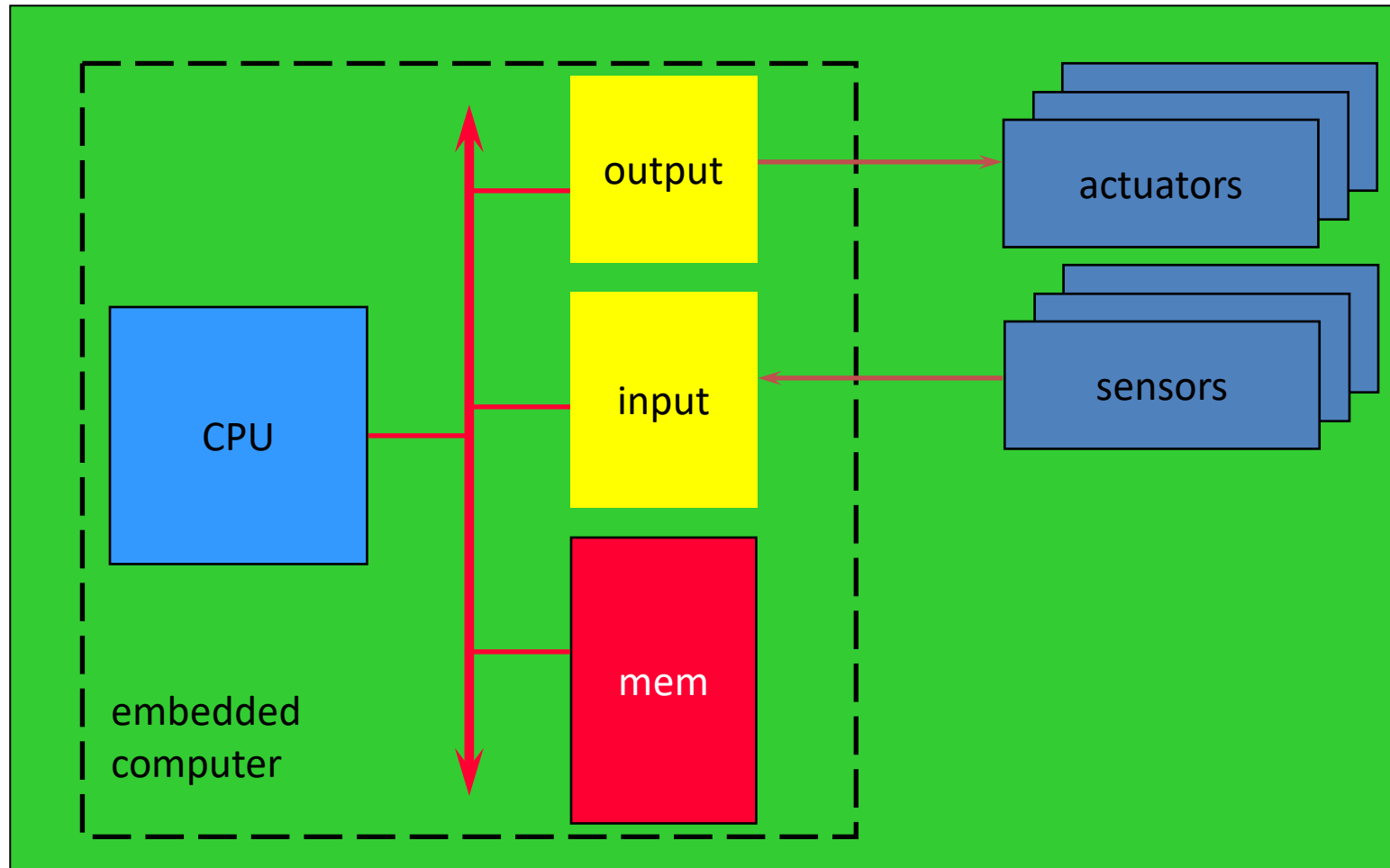
- Due to the requirements of **low power** and **low cost**
- Example: Crossbow TelosB
 - http://www.willow.co.uk/TelosB_Datasheet.pdf
 - Processing capability
 - 8 MHz TI MSP430 microprocessor
 - Memory size
 - 10kB RAM
 - 48kB Flash memory for program
 - 1Mb external flash for data logging
 - Data transmission
 - ZigBee: 250kbps data rate



Design Teams

- Often designed by a small team
- Often have to meet tight deadlines
 - 6 month market window is common.
 - Example: Can't miss back-to-school window for calculator.
- Compare to teams for
 - Pentium
 - Windows XP, Vista

Embedding A Computer



Alternative Technology for CPU

- Application-Specific Integrated Circuits (ASICs)
- Microprocessors
- Field-Programmable Gate Arrays (FPGAs)
- Why do we use a microprocessor?

ASIC

Example: Digital baseband processing for cell phones

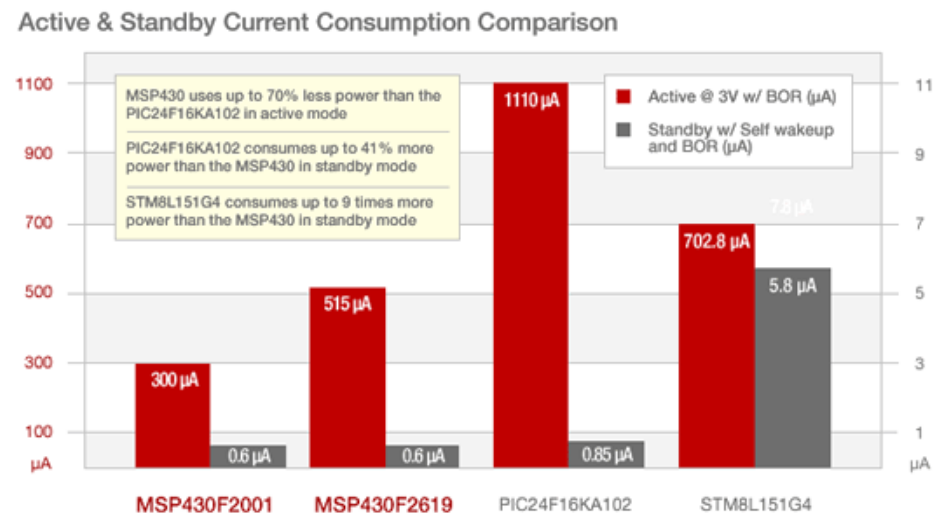
- ✓ **Performance:** Fast!
- ✓ **Power:** Fewer logic elements → low power
- ✗ **Development cost:** Very high
 - 2 million \$ for starting production of a new ASIC
 - Needs a long time and a large team
- ✗ **Reprogrammability:** None!
 - Single-purpose devices
 - Difficult to upgrade systems

Microprocessors – Performance Paradox

- Von Neumann (or Harvard) architecture is fundamentally slow!
 - Fetch, decode instructions
- But engineering optimization often fixes it!
 - heavily pipelined
 - cache
 - clock frequency
 - circuit density
 - aggressive VLSI technology
 - multi-core processor

Microprocessors – Power

- However, microprocessors improve performance at the cost of power!
 - Performance/watt remains low.
- Solution
 - Modern microprocessors offer features to help control power consumption.
 - Software can help reduce power consumption.



Microprocessors – Reprogrammability and Development Cost

Let software do the work!

- Fast and cheap development
- Easy to upgrade, patch, and reuse
- Simplify the design of families of products
- Multi-purpose device: same hardware can run multiple applications (e.g., camera phones)

FPGA

- Programmable hardware
- Combine the benefits of ASIC and microprocessor
 - Hardware implementation → good performance/watt
 - Reprogrammable → lower development cost

FPGA - Limitations

- Current FPGAs are not ready as a standalone platform
- Many overhead transistors → waste power!
 - Configurable connections
 - Personalization memory: contribute to 70% of the transistors on a chip
- More difficult to program than software
- More commonly used for prototyping

State of The Practice

- Microprocessor is the dominant player
 - Flexibility and low development cost >> performance/watt consideration
 - Power management techniques are crucial
 - To be covered in this class.
- Microprocessor + ASIC is common
 - Ex: cell phone
- FPGA is expected to play an increasingly important role

The State of the Art

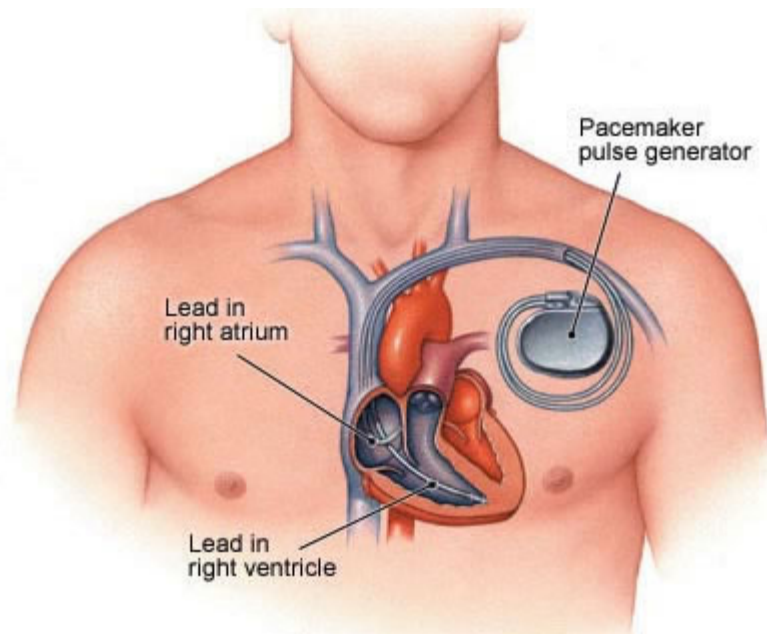
- Revival of ASICs
 - Domain-specific chip design
 - AI-specialized processors
- Utilizing the FPGA's parallelism
 - Graphical processing
 - Neural network computations

Design Challenges

- Non-functional constraints
 - How do we meet deadlines?
 - Faster hardware or better software?
 - How do we minimize power?
 - Turn off unnecessary logic? Reduce memory accesses? Slow down CPU?
 - Cost considerations
- Trade-offs among constraints
 - Several constraints may coexist at the same time
- Optimization & analysis are important!
 - Will be covered in this course

Some Examples

- Artificial pacemaker



Some Examples

- Self-driving cars



Some Examples

- Street surveillance camera

