Embedded Operating Systems

Che-Wei Chang
chewei@mail.cgu.edu.tw
Department of Computer Science and Information Engineering, Chang Gung University

Flexible Embedded Systems

- Features of Embedded Systems
  - Customized hardware with high scalability
  - Heterogeneous devices with unified interface
  - Application-aware designs for energy saving

Hardware and System Software Integration





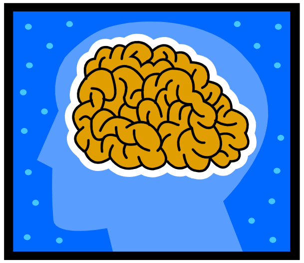
Key Technologies for Systems

- Fast System Initialization
  - Keep the system and application states for users
  - Reduce the Initialization time of applications
- Energy-Efficient Designs
  - Adjust the frequencies of processors
  - Change the states of peripheral devices
- Performance Tuning Tools
  - Understand the hot spots of applications
  - Exploit the advantages of hardware

Course Roadmap

Basic Concepts

- Embedded System Design Concepts
- Embedded System Developing Tools and Operating Systems
- Embedded Linux and Android Environment



Core Technology

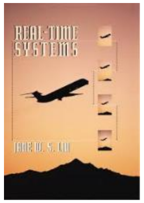
- Real-Time System Design and Scheduling Algorithms
- System Synchronization Protocols

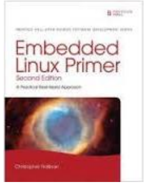
Real Exercises

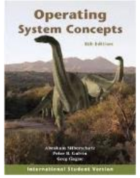
- System Initialization and Memory Management
- Power Management Techniques and System Routine
- Embedded Linux Labs and Exercises on Linux

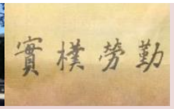
Syllabus

- **Lecturer**: Che-Wei Chang (張哲維)
- **Lecture Hours**: Tuesday 9:10 a.m. – 12:00 p.m.
- **Office Hours**: Wednesday 1:00 p.m. – 3:00 p.m.
- **Classroom**: Seminar Room 1
- **Reference Books and Slides:**
  - Jane Liu, “Real-Time Systems,” Prentice Hall, 2000.
  - Christopher Hallinan, “Embedded Linux Primer,” 2nd Edition, Prentice Hall, 2011.
  - Silberschatz, Galvin, and Gagne, “Operating System Concepts,” 9th Edition, John Wiley & Sons, 2013.









Grading and Resources

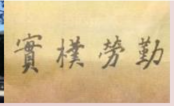
- Midterm: 30%
- Discussion and Quiz: 20%
- Exercises: 20%
- Final Project Presentation and Report: 30%
- Course Website: https://icechewei.github.io/webpage/teaching.html

Embedded Systems — Overview

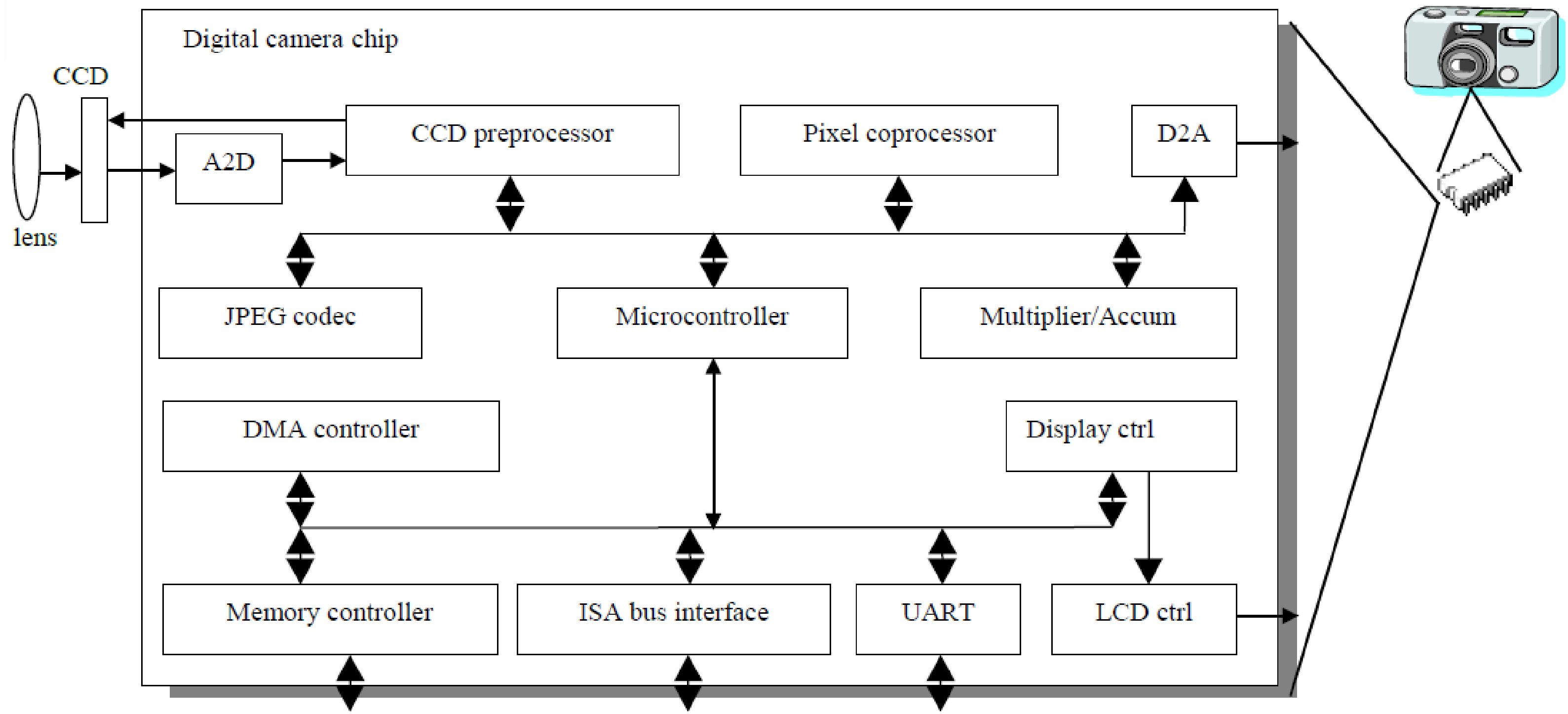
Definition of Embedded Systems

- It is difficult to define embedded systems
  - An embedded system is a digital system
  - An embedded system has computing processors
  - An embedded system runs dedicated functions
  - An embedded system is frequently used as a controller

- An embedded system is a computer system with some dedicated functions within a larger mechanical or electrical system, often with real-time computing constraints. - From Wikipedia

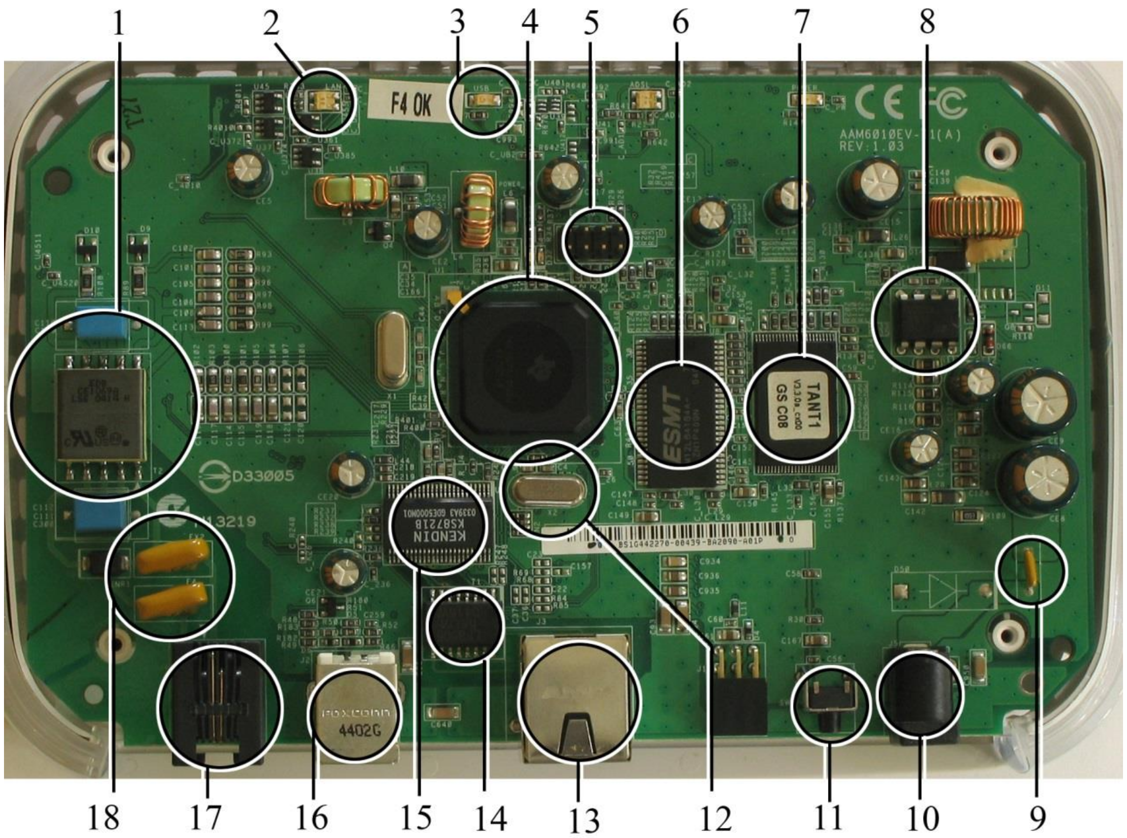


An Embedded System Example — A Digital Camera



- Single-Functioned: always a digital camera
- Tightly-Constrained: low cost, low power, small
- Reactive and Real-Time: short response time

An Embedded System Example — An ADSL Modem – From Wikipedia



- Microprocessor (4), RAM (6), and Flash Memory (7), ...

Design Challenge—Optimizing Performance Metrics

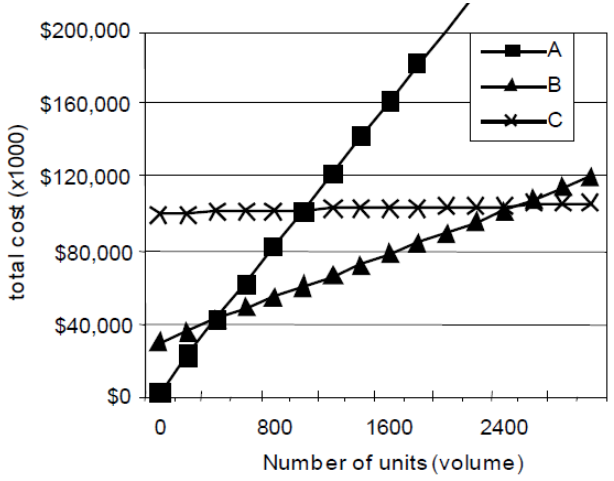
- Obvious Design Goal
  - Construct an implementation with desired functionality
- Performance Metrics
  - Performance metrics are the measurable features of a system's implementation
  - Simultaneously optimizing numerous design metrics is a challenging issue

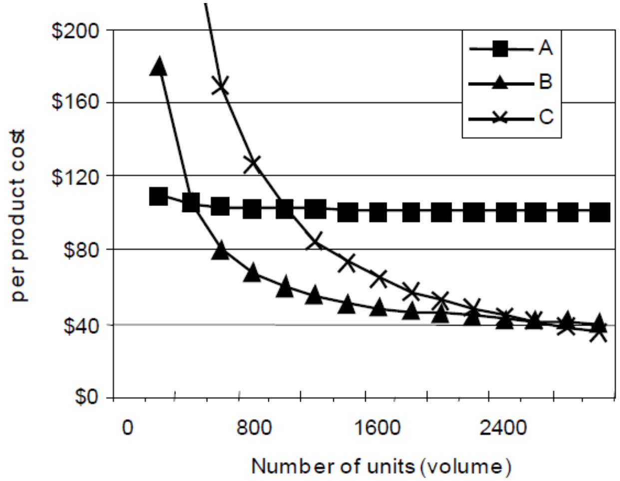
Common Performance Metrics

- Unit Cost: the monetary cost of manufacturing each copy of the system
- NRE Cost (Non-Recurring Engineering cost): the one-time monetary cost of designing the system
- Size: the physical space required by the system
- Performance: the execution time or throughput of the system
- Power: the amount of power consumed by the system
- Flexibility: the ability to change the functionality of the system without incurring heavy NRE cost
- Time-to-Market: the time required to develop a system to the point that it can be released and sold to customers
- Maintainability: the ability to modify the system after its initial release

NRE and Unit Cost

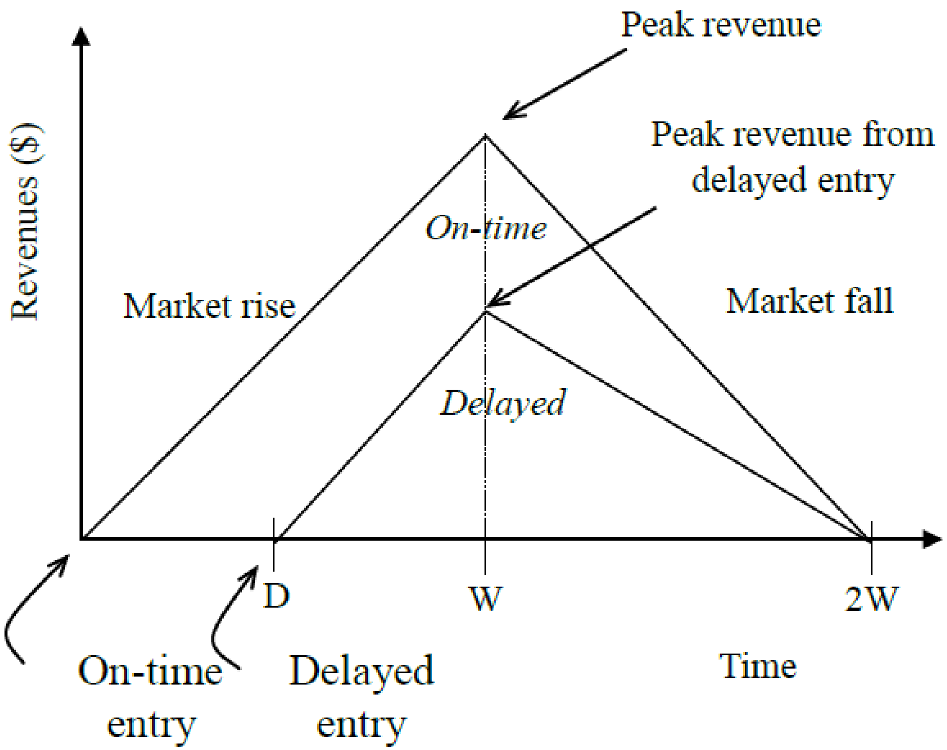
- Compare Technologies by Costs— the best solution depends on quantity of the product
  - Technology A: NRE=$2,000, unit=$100
  - Technology B: NRE=$30,000, unit=$30
  - Technology C: NRE=$100,000, unit=$2





  - We must also consider time-to-market

Delayed Market Entry



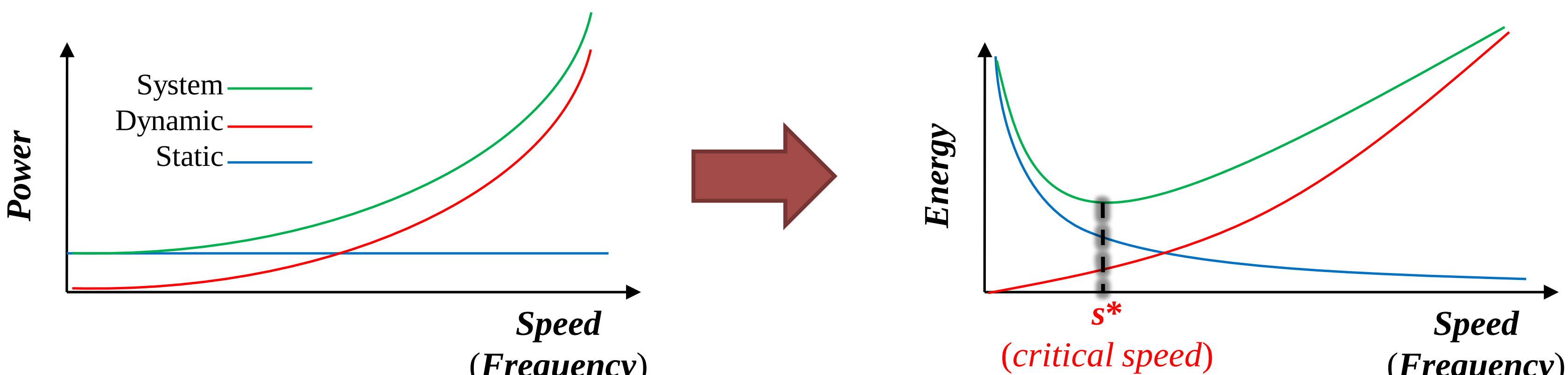
- A Simplified Revenue Model
  - Product life = 2W, peak at W
  - The time of market entry defines a triangle, representing the market penetration
  - The triangle area represents the revenue
- Loss
  - The difference between the on-time and delayed triangle areas

Performance Design Metrics

- Latency (response time)
  - Time between task start and end
    - e.g., Cameras A and B process an image in 0.5 seconds
- Throughput
  - Number of tasks per second
    - e.g. Camera A processes 2 images per second
  - Throughput can be more than latency seems to imply due to concurrency
    - e.g. Camera B may have two cores to process 4 images per second (by pipelining or multithreading on multiple cores)

Performance and Power

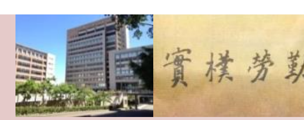
- Dynamic Power Consumption
  - Switching power and short-circuit power
  - Dynamic Voltage Frequency Scaling (DVFS)
- Static Power Consumption
  - Leakage power
  - Dynamic Power Management (DPM)



Department of Computer Science and Information Engineering, Chang Gung University

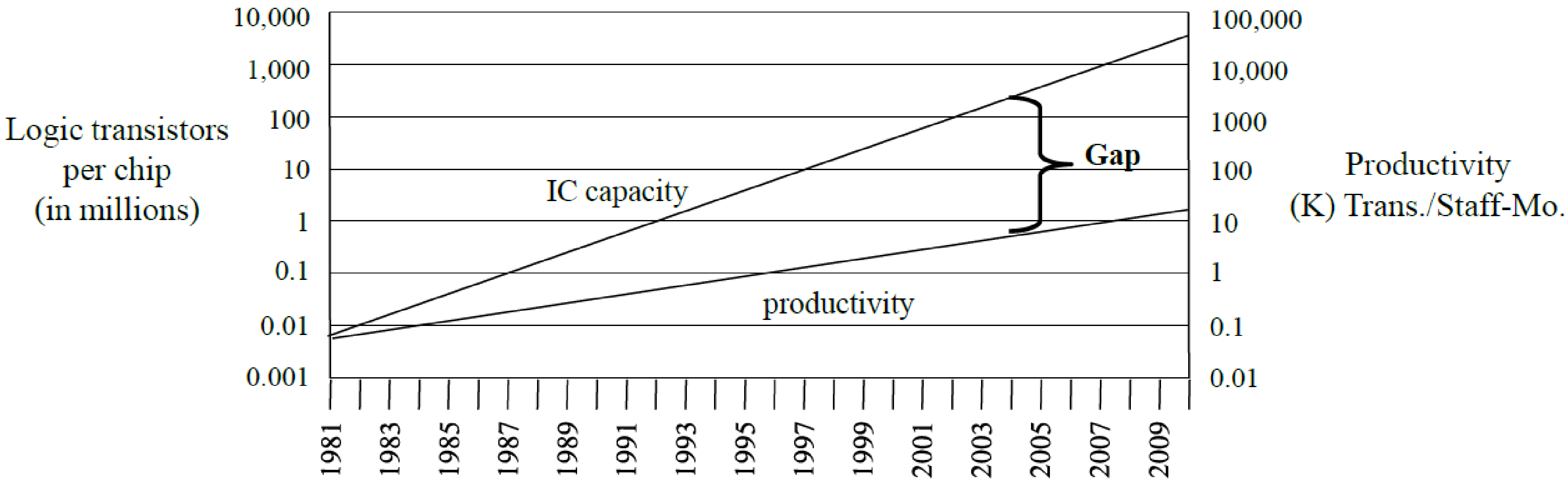
Solutions for a Specific Function

- General Purpose Processor
  - A flexible software solution
- Special Purpose Processor
  - Specialized processors for some application area, e.g., graphics processing units for 2D and 3D rendering
- Field Programmable Gate Array (FPGA)
  - An integrated circuit designed to be configured by customers after manufacturing
- Application-Specific Integrated Circuit (ASIC)
  - An integrated circuit customized for a particular use



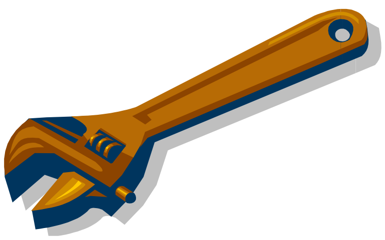
Design-Productivity Gap

- While designer productivity has grown at an impressive rate over the past decades, the rate of improvement has not kept pace with chip capacity



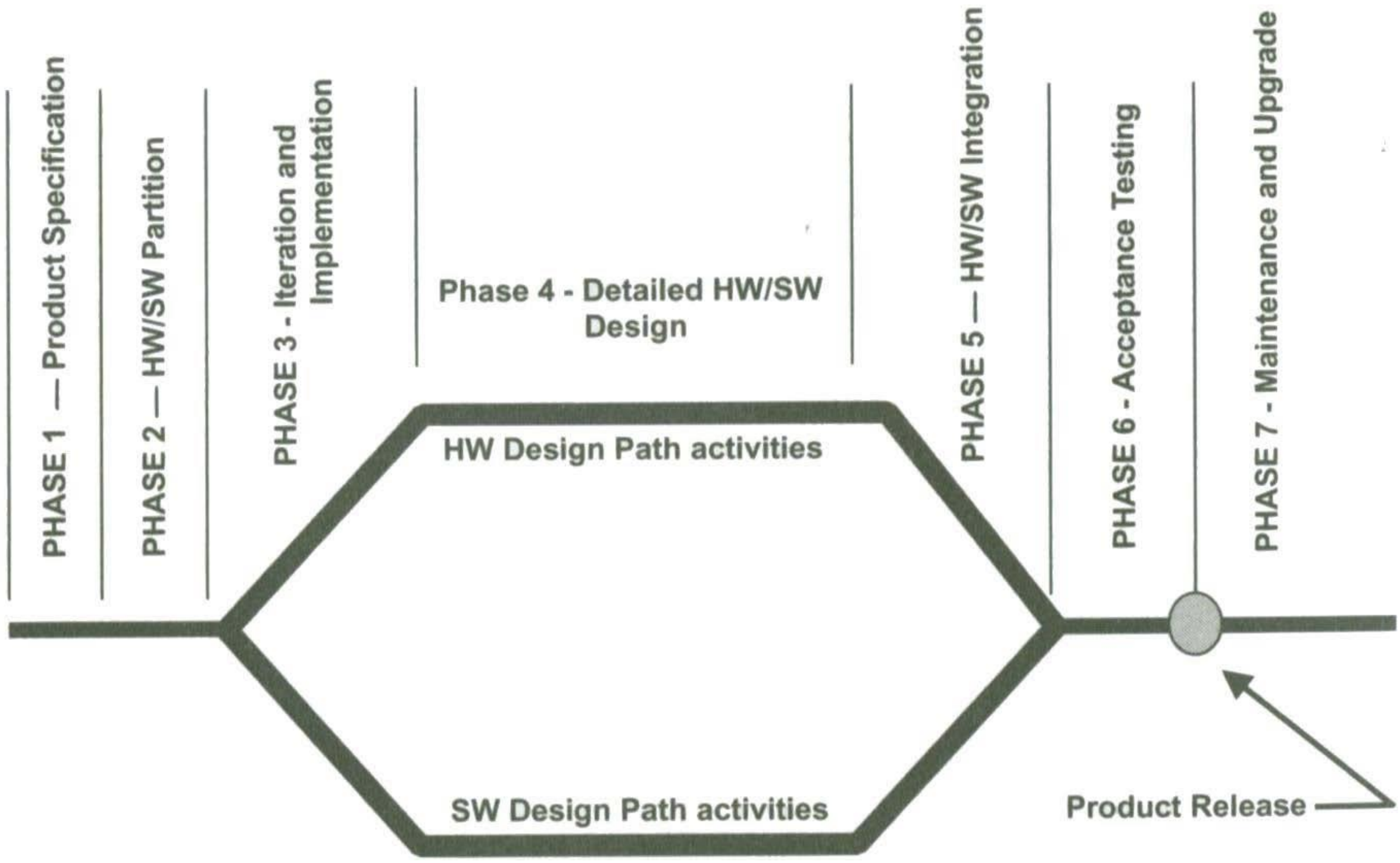
Building an Embedded System

- Understand the embedded system design life cycle
- Be familiar with the real-time requirements of embedded systems
- Use embedded system developing tools to work more efficiently
- Include an embedded operating system with driver support for a complicated environment of applications



The Embedded Design Life Cycle

Workflow of Embedded Designs



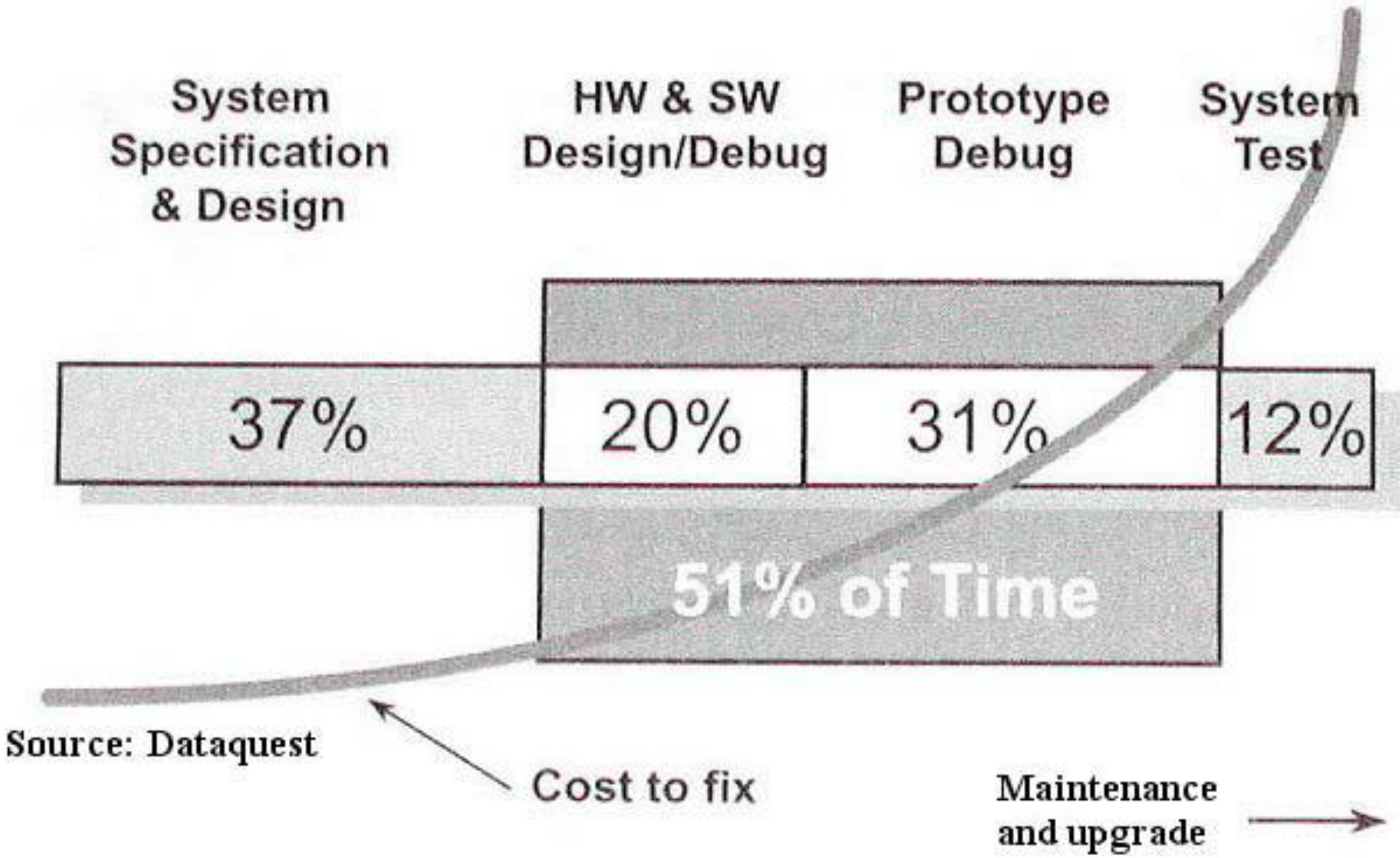
Product Specification

- R&D Engineer View
- Marketing and Sale Department View
- Customer View
- Questionaries
- Marketing Specialists

Partitioning Decision

- Complex optimization problem
- Many embedded system designs are required to be :
  - Price sensitive
  - Leading-edge technology
  - Non-standard
  - Market competitive
- These conflicting requirements make it difficult to create an optimal design

Detailed Hardware and Software Design



Debugging an Embedded System

- General Requirements
  - Run control
  - Memory substitution
  - Real time analysis
  - On-Chip Hooks
  - OS Supports
- The Holy Grail of Embedded System Design
  - Real time nature
  - Accurately model or simulate is difficult

The Concept of Real-Time Embedded Systems

Introduction to Real-Time Systems

- What is a real-time system?
  - Any system where a timely response by the computer to the external environment is vital
- Examples:
  - Multimedia systems, virtual reality, games
  - Avionics, air traffic control, robots, automobiles, nuclear power plant
  - Stock market, trading system, information access, etc.

Real-Time Issues and Research

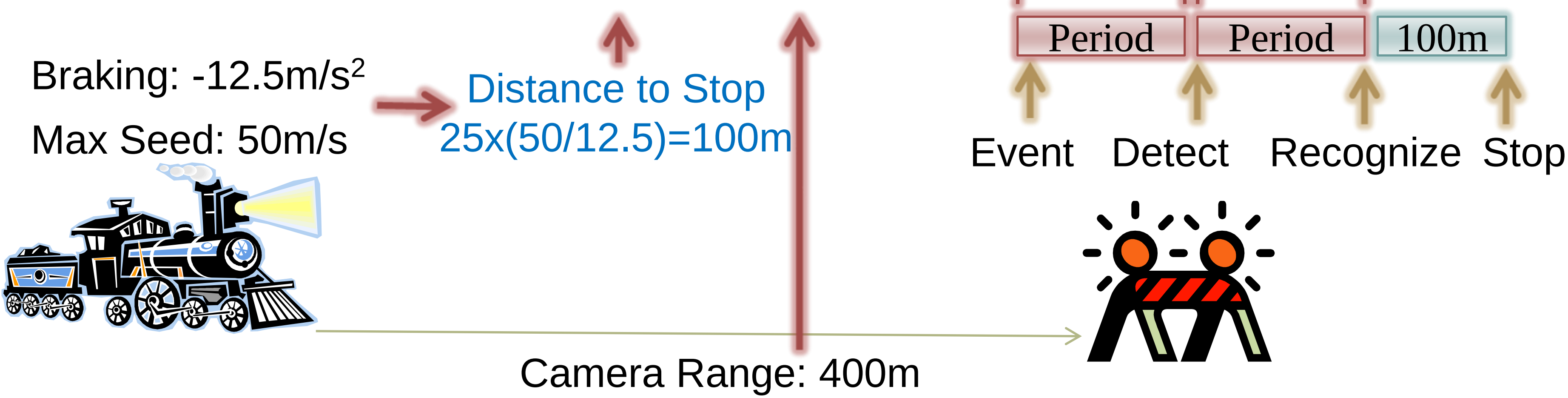
- Software Engineering
  - System Design Methodologies
  - Toolchain Designs
- Operating Systems
  - Many/Multi-Core Task Management and Scheduling
  - Task Synchronization
  - Energy-Efficient System Designs
  - File Systems and Storage Systems
- Programming Models
  - Heterogeneous/homogeneous Multiprocessor Programming
  - Better control over timing

An Example of Real-Time Designs

- A camera periodically takes a photo
- The image recognition result will be produced before the next period
- If there is an obstacle, the train automatically brakes

Time of a Period = 150/50 = 3s

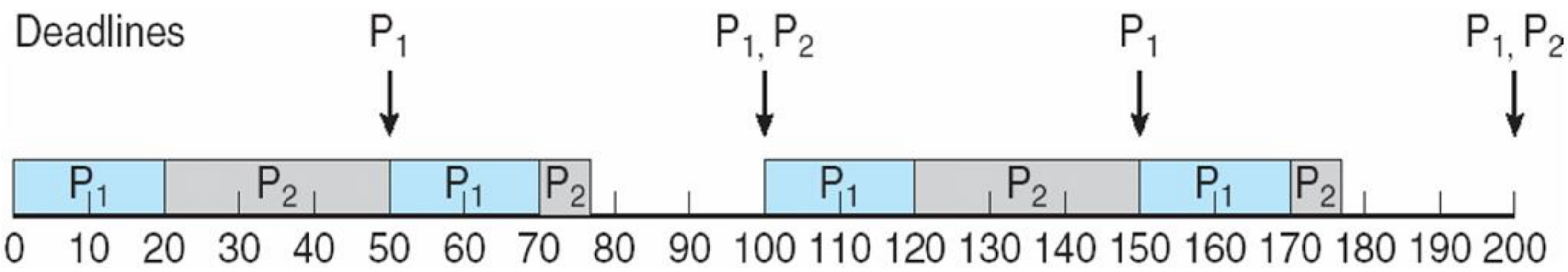
Distance of a Period = (400 -100)/2 = 150m



Rate Monotonic Scheduling

- A scheduler is needed if there is more than one real-time task
- Rate monotonic scheduler: A static priority is assigned to each task based on the inverse of its period
  - A task with shorter period ➔ higher priority
  - A task with longer period ➔ lower priority
  - For example:
    - P_1 has its period 50 and execution time 20
    - P_2 has its period 100 and execution time 37

    ➔P_1 is assigned a higher priority than P_2



Challenges of Real-Time Systems

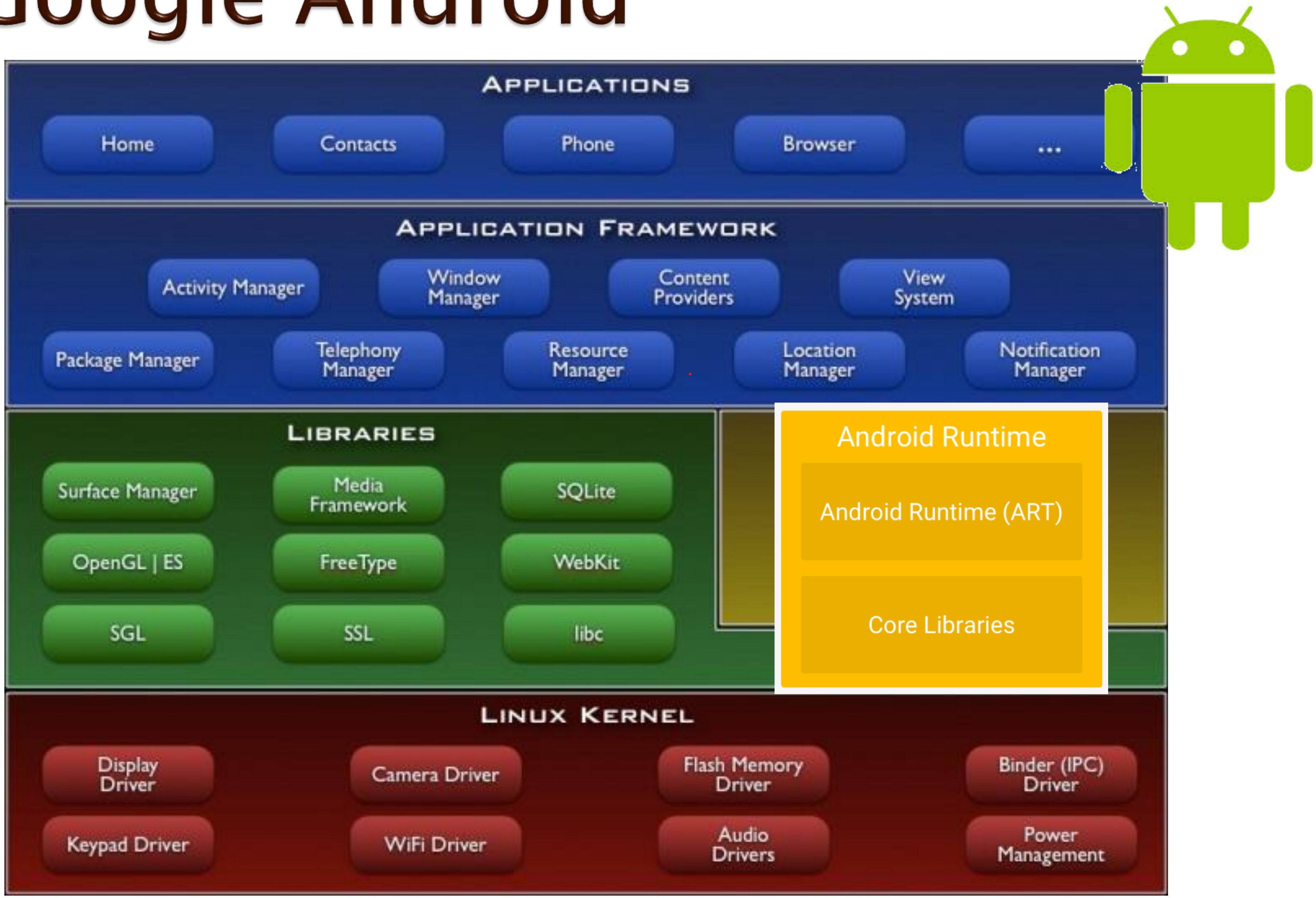
- Are there other scheduling algorithms?
  - Yes, analysis should also be provided to choose a proper scheduling algorithm.
- Are there some aperiodic tasks?
  - Yes, we then need to jointly consider periodic and aperiodic tasks.
- Can tasks share some resources or data?
  - Of course, the synchronization protocols should be provided to protect the access to the shared resources and data.

Embedded Operating Systems and Developing Tools

Apple iOS

- iOS is the operating system that runs iPhones, iPod Touches, iPads, and Apple TVs.
- The language used to develop the software for iOS is Objective-C (and Swift)
- Features
  - Home Screen
  - Multi-Touch
  - Not Fully Open-Sourced

Google Android



Real-Time Operating Systems

- A RTOS is an abstraction from hardware and software programming
  - Shorter development time
  - Less porting efforts
  - Better reusability
- Choosing an RTOS is important
  - High efforts when porting to a different OS
  - The chosen OS may have a high impact on the amount of resources needed
- Example: eCos, Nucleus, VxWork, QNX, OSE, RT-Linux, uC/OSII

Embedded System Development Tools

- Compiler Tools
  - Quality of code generator
  - Support particular features of the target hardware
- Hardware and Software Debugging Tools
  - ICE (In-Circuit Emulator), ROM emulator, logic analyzer, performance analyzer
  - GNU debugger, ARM RealView development suite, …
- Performance Measuring Tools
  - Development suites
  - Power meters
  - Data acquisition devices

What is an embedded system?

Do you have your answer?