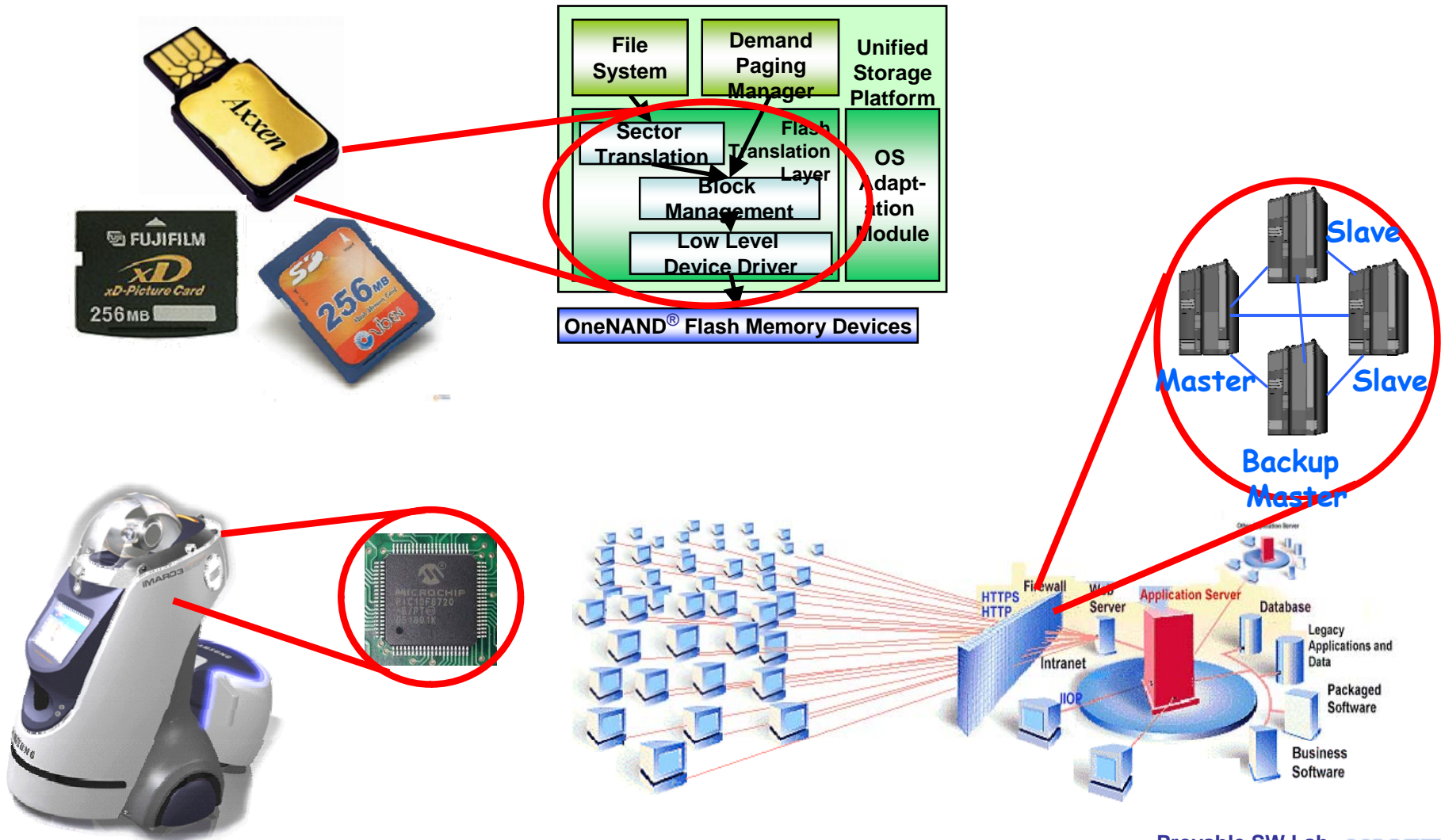# CS453: Formal Software Verification Techniques

## Moonzoo Kim

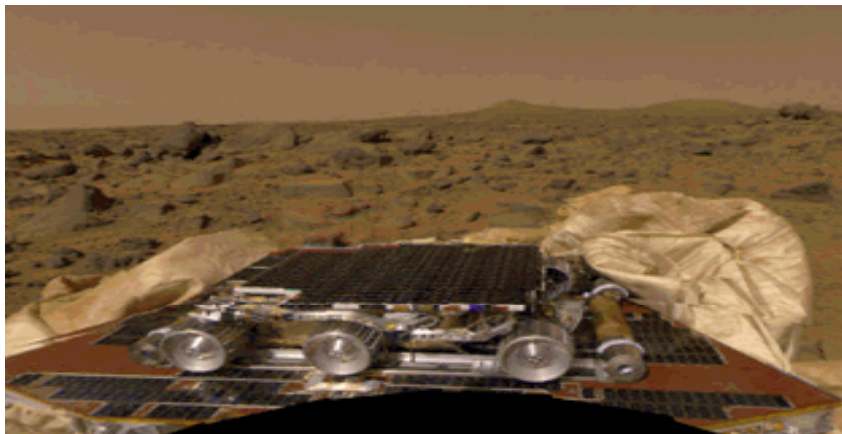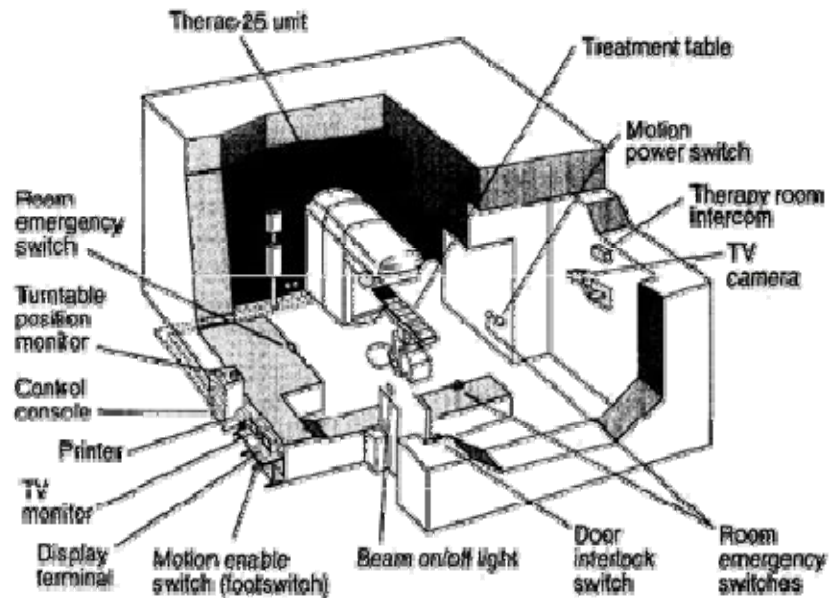*Provable Software Laboratory*

*CS Dept. KAIST*

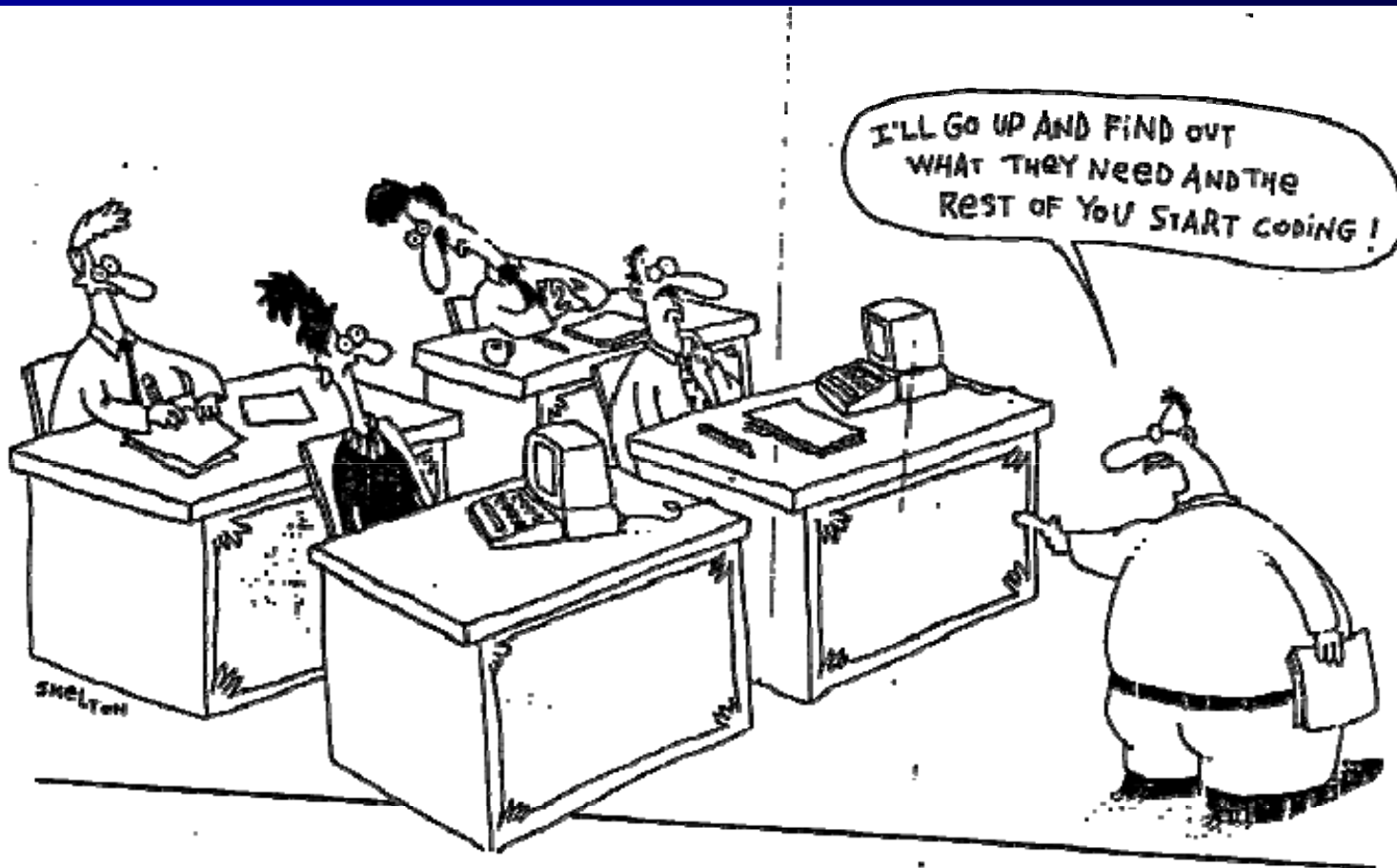# PROVABLE SW LAB



File System

Demand Paging Manager

Unified Storage Platform

Sector Translation

Flash Translation Layer

Block Management

OS Adapt-ation Module

Low Level Device Driver

OneNAND® Flash Memory Devices

Slave

Master

Slave

Backup Master

HTTPS HTTP

Firewall

Web Server

Application Server

Database

Legacy Applications and Data

Intranet

IIOP

Packaged Software

Business Software

**Provable SW Lab.**
**CS Dept. KAIST**
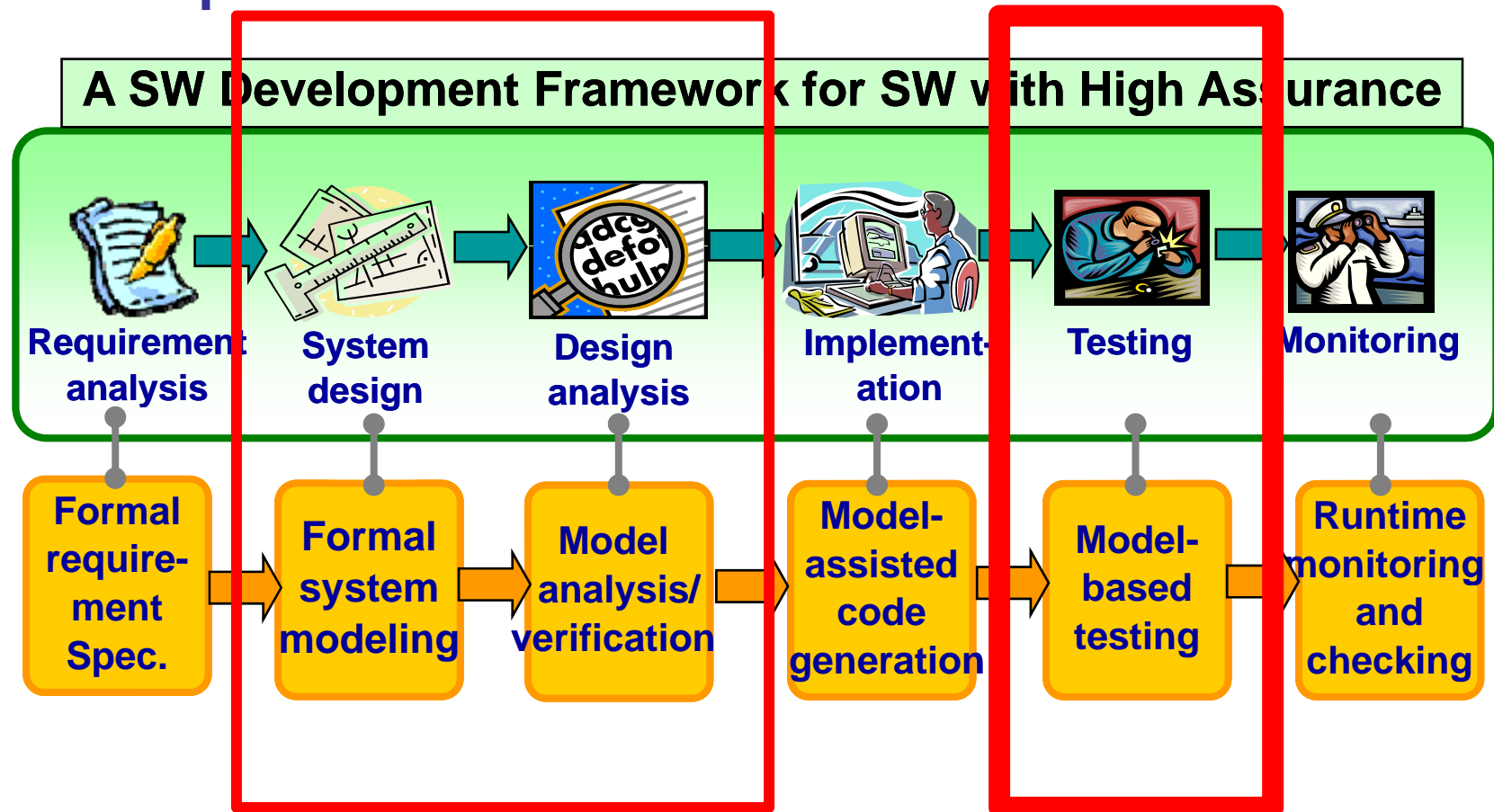
2

# Motivation: Poor Quality of SW

# Current Practice for SW



- **SW developers have to follow scientific disciplines for building and analyzing software with high quality**
  - This class focuses on the analysis activities
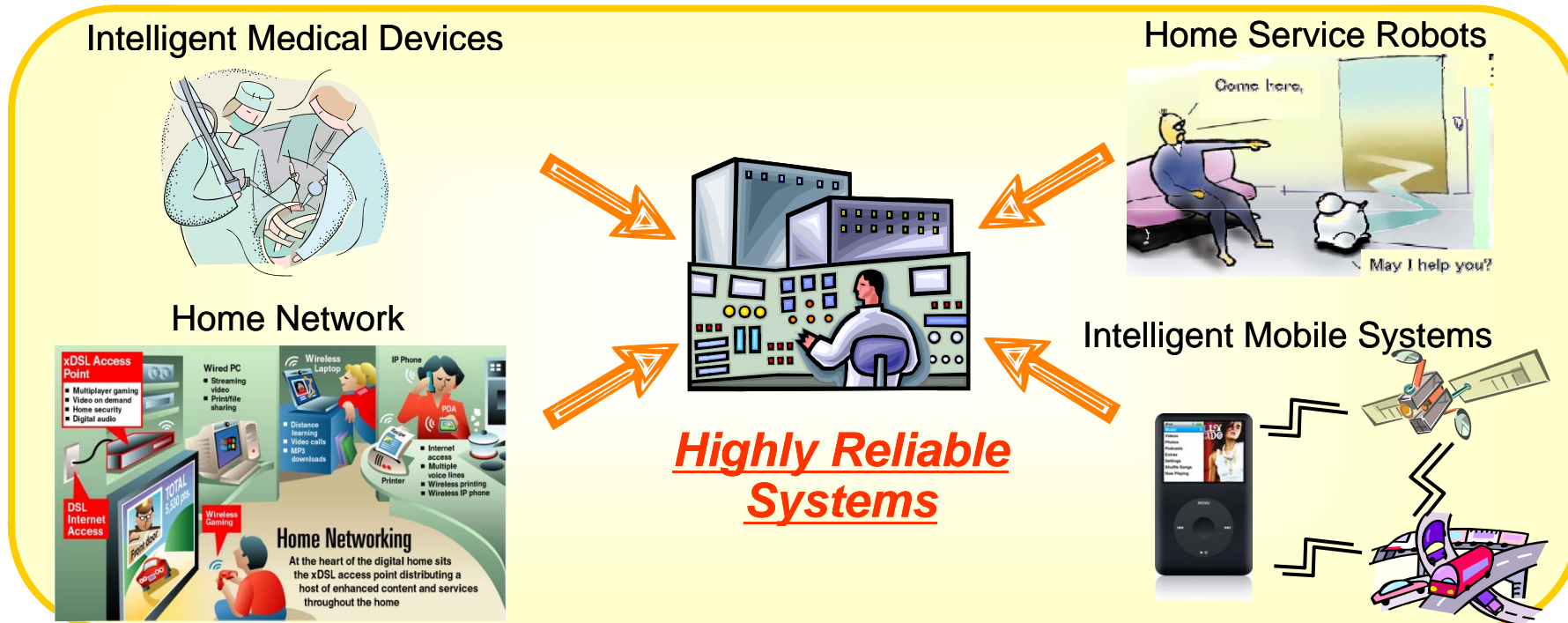
# Software Development Cycle

- **A practical end-to-end formal framework for software development**

**A SW Development Framework for SW with High Assurance**

| Requirement analysis | System design | Design analysis | Implement-ation | Testing | Monitoring |
|---|---|---|---|---|---|
| Formal require-ment Spec. | Formal system modeling | Model analysis/ verification | Model-assisted code generation | Model-based testing | Runtime monitoring and checking |

# Main Target Systems

- **Embedded systems where <span style="color:red">highly reliable SW technology</span> is a key to the success**
  - The portion of SW in commercial embedded devices increases continuously
  - More than 50% of development time is spent on SW testing and debugging



Intelligent Medical Devices

Home Service Robots

Home Network

Intelligent Mobile Systems

*Highly Reliable Systems*

# How to Improve the Quality of SW

1.  **Systematic testing**
    -   Coverage criteria
    -   Mutation analysis

2.  **Debugging through automated analysis tools**
    -   Scientific treatment of SW with computing power
    -   Useful tools are available

3.  **Formal verification**
    -   Guarantee the absence of bugs!!!

Provable SW Lab.
CS Dept. KAIST  KAIST

# Questions???

- **Is formal analysis really feasible in industry?**
  - Yes, several case studies even in Korea
- **Is formal analysis academically significant?**
  - Yes, 3 Turing awardees in '07
- **Is formal analysis too hard to learn and use?**
  - No, there are tools available

# Verification of High-Availability Protocol

- **We develop a formal model of high-availability protocol used in commercial security appliances**
  - HA protocol coordinates a group of firewalls
- **We found several problems in HA regarding a master election procedure**
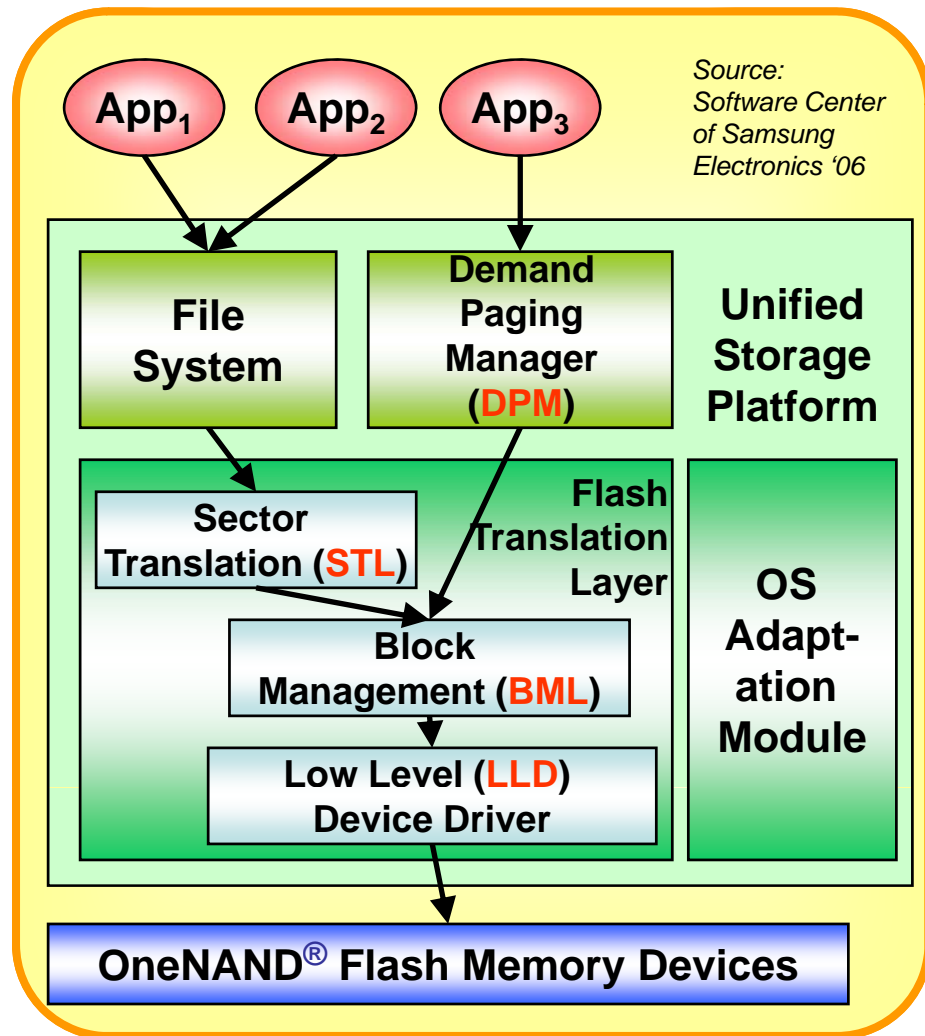


9
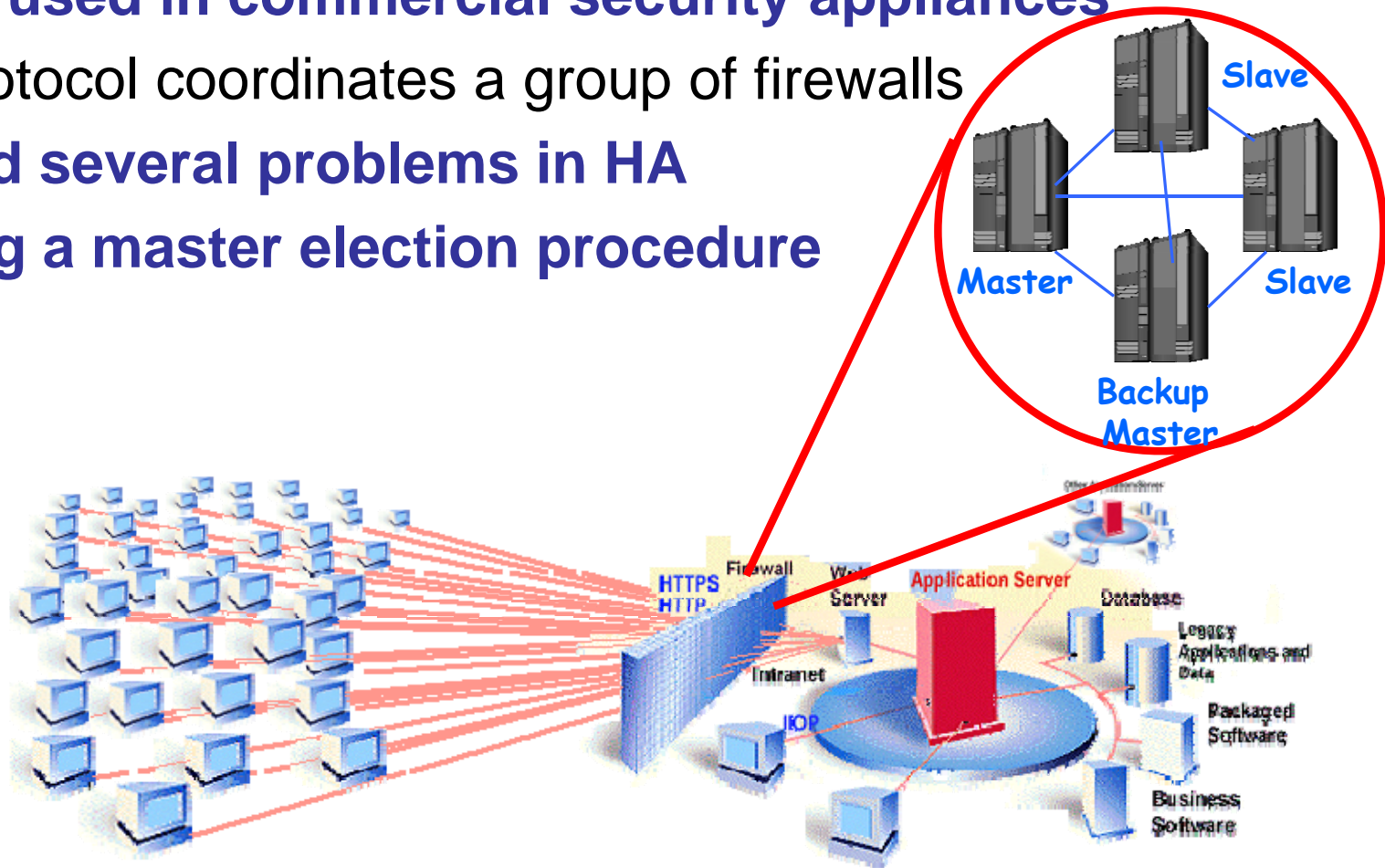
# Home Service Robot

■ Designed for providing various services to human user
 - Service areas : home security, patient caring, cleaning, etc
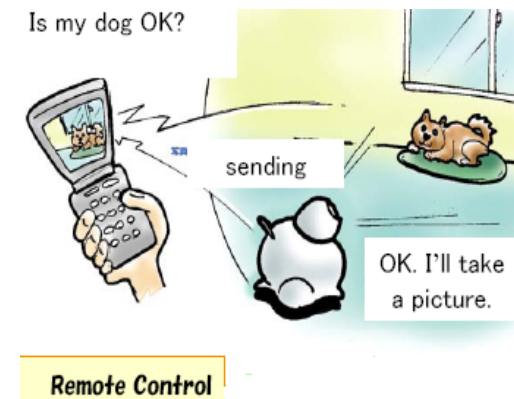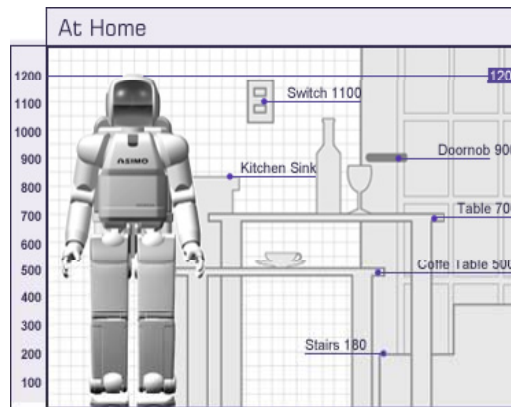
# OneNAND® Flash Memory Verification

- **Each memory cell can be written limited number of times only**

- **XIP by emulating NOR interface through demand-paging scheme**

- **Performance enhancement**



Source: Software Center of Samsung Electronics '06

App₁ App₂ App₃

File System

Demand Paging Manager (DPM)

Unified Storage Platform

Sector Translation (STL)

Flash Translation Layer

Block Management (BML)

Low Level (LLD) Device Driver

OS Adapt-ation Module

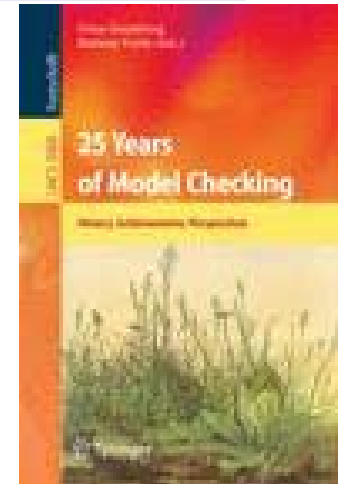OneNAND® Flash Memory Devices

# Research Trends toward Quality Systems

- **Academic research on developing embedded systems has reached stable stage**
  - just adding a new function to a target system is not considered as an academic contribution anymore
- **Research focus has moved on to the quality of the systems from the mere functionalities of the systems**
  - Energy efficient design, ez-maintenance, dynamic configuration, etc
- **Software reliability is one of the highly pursued qualities**
  - NSDI 2007 Best paper
    - "Life, Death, and the Critical Transition: Finding Liveness Bugs in Systems Code" @ U.C. San Diego
      - Heuristic application of model checking to detect liveness bug
  - OSDI 2004 Best paper
    - "Using Model Checking to Find Serious File System Errors" @ Stanford
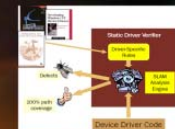      - Application of software model checking to find FS bugs

# Formal Verification as a Foundational and Promising CS Research

- **2007 ACM Turing Awardees**
  - Prof. Edmund Clarke
  - Dr. Joseph Sipfakis
  - Prof. E. Allen Emerson
- **For the contribution of migrating from pure research to industrial reality**
- **One of the four Microsoft Research main areas**

# Class Schedule

- wk1: overview on formal SW analysis techniques
- wk2: conventional testing techniques
- wk3: background on Propositional logic and SAT (Satisfiability) solvers
- wk4: SAT solver heuristic and tool application 1: MiniSAT
- wk5: background on First order logic
- wk6: Satisfiability Modulo Theory (SMT) basic
- wk7: tool application 2: Z3

- wk8: midterm exam
- wk9: advanced application of SMT solvers
- wk10: directed automated random testing
- wk11: tool application 3: CUTE, CalFuzzer
- wk12: basic temporal logic for requirement property
- Wk13-14: tool application 4: Spin & NuSMV
- wk15: state space minimization techniques
- wk16: final exam

# Administrative Stuff

- **Instructor: Prof. Moonzoo Kim**
- **Class time: Tue/Thr 10:30 -12:00**
- **Office hour: Tue 4:30-6:30 PM**
- **Grade policy**
  - HW 50%
  - Attendance & quiz 20%
  - Mid exam 15%
  - Final exam 15%
- **TA: Yunho Kim (Rm#2438)**
- **Web page: http://pswlab.kaist.ac.kr**

# Final Remarks

- **For undergraduate students:**
  - Highly recommend URP studies or independent studies
    - Ex. 이준희 (05학번) got a silver award and macbook air notebook ☺
      - Debugging Linux kernel through model checking to detect concurrency bugs


- **For graduate students:**
  - Welcome research discussions to apply formal analysis techniques
    - Ex. Systematically debugging C programs or prove the correctness of algorithms, etc