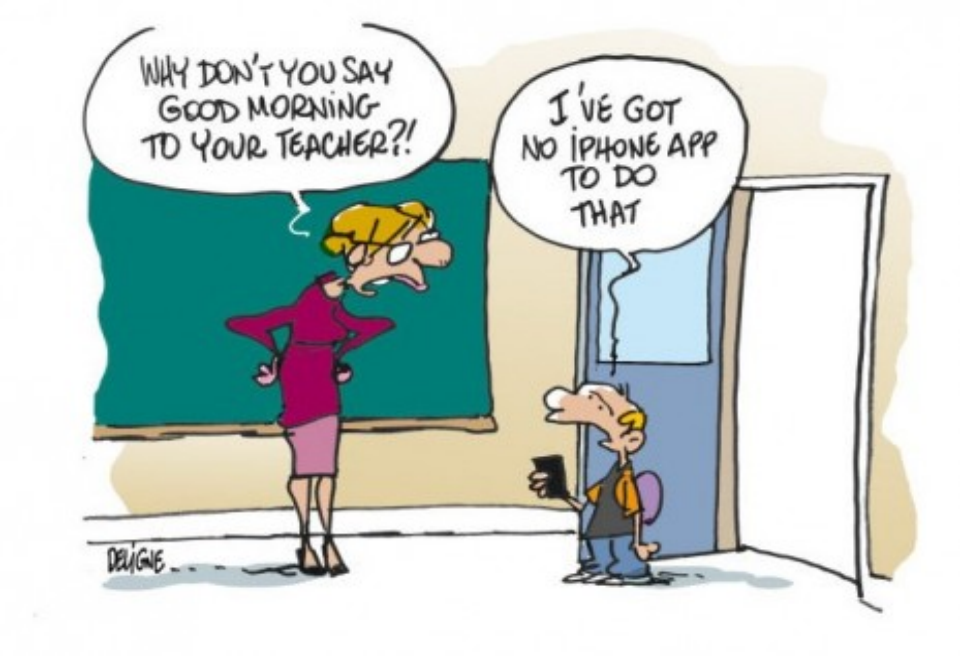
INFORMATION CODING
Linköping University

Software development for Smartphones/tablets



Motivation

- In a few years time tablets and smartphones will be running PC applications / on PC hardware

Motivation

- In a few years time tablets and smartphones will be running PC applications / on PC hardware
  - Promised for a long time

Motivation

- In a few years time tablets and smartphones will be running PC applications / on PC hardware
  - Promised for a long time
  - Progress, but a limit is reached (or will be soon)

Motivation

- In a few years time tablets and smartphones will be running PC applications / on PC hardware
  - Promised for a long time
  - Progress, but a limit is reached (or will be soon)
  - At the same time, the performance of embedded systems increases significantly

Motivation

- In a few years time tablets and smartphones will be running PC applications / on PC hardware
  - Promised for a long time
  - Progress, but a limit is reached (or will be soon)
  - At the same time, the performance of embedded systems increases significantly
- Smartphones / tablets will replace PCs in normal households

Motivation

- In a few years time tablets and smartphones will be running PC applications / on PC hardware
  - Promised for a long time
  - Progress, but a limit is reached (or will be soon)
  - At the same time, the performance of embedded systems increases significantly
- Smartphones / tablets will replace PCs in normal households
  - Already happening

Motivation

- In a few years time tablets and smartphones will be running PC applications / on PC hardware
  - Promised for a long time
  - Progress, but a limit is reached (or will be soon)
  - At the same time, the performance of embedded systems increases significantly
- Smartphones / tablets will replace PCs in normal households
  - Already happening
  - PCs in future: only at work, probably for hardcore gaming

Market

- Android
- iOS (iPhone, iPad)
- Other
  - Firefox, Ubuntu
  - Blackberry OS
  - Windows

Market

- Android (>50%)
- iOS (iPhone, iPad)
- Other (<10%)
  - Firefox, Ubuntu
  - Blackberry OS
  - Windows

Market

- **Android (>50%)**
- **iOS (iPhone, iPad)**
- Other (<10%)
  - **Firefox, Ubuntu**
  - **Blackberry OS**
  - Windows

Market

- CPUs: Arm

Market

- CPUs: Arm
- GPUs
  - Imagination Technologies PowerVR (market leader)
  - Arm Mali
  - Qualcomm Adreno (former: ATI)
  - NVIDIA Tegra

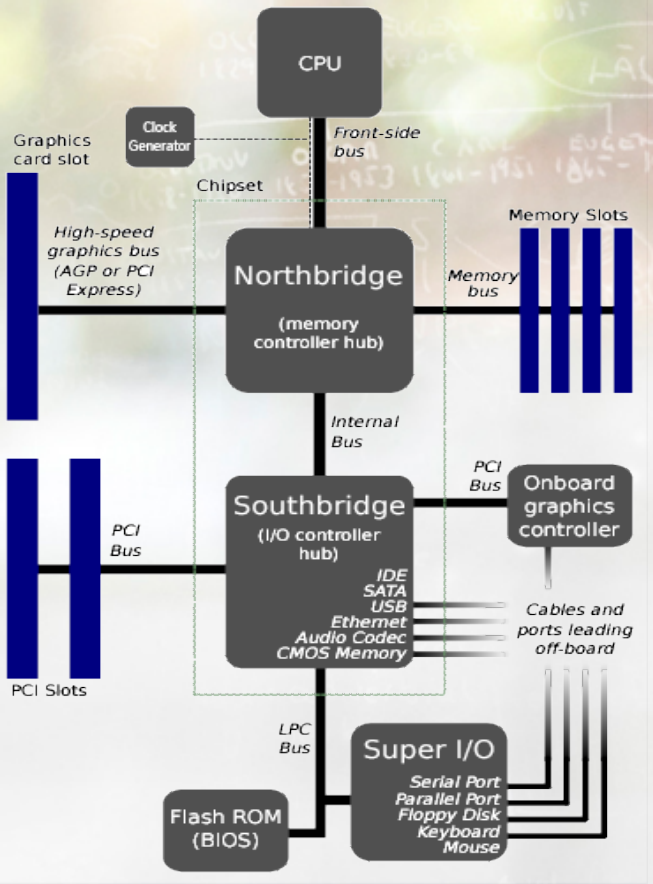
Market

- Chip-producer
  - Texas Instruments
  - Qualcomm
  - ST Ericsson (?)
  - Samsung
  - NVIDIA (ARM+Tegra)
  - Apple (ARM+PowerVR)

Market

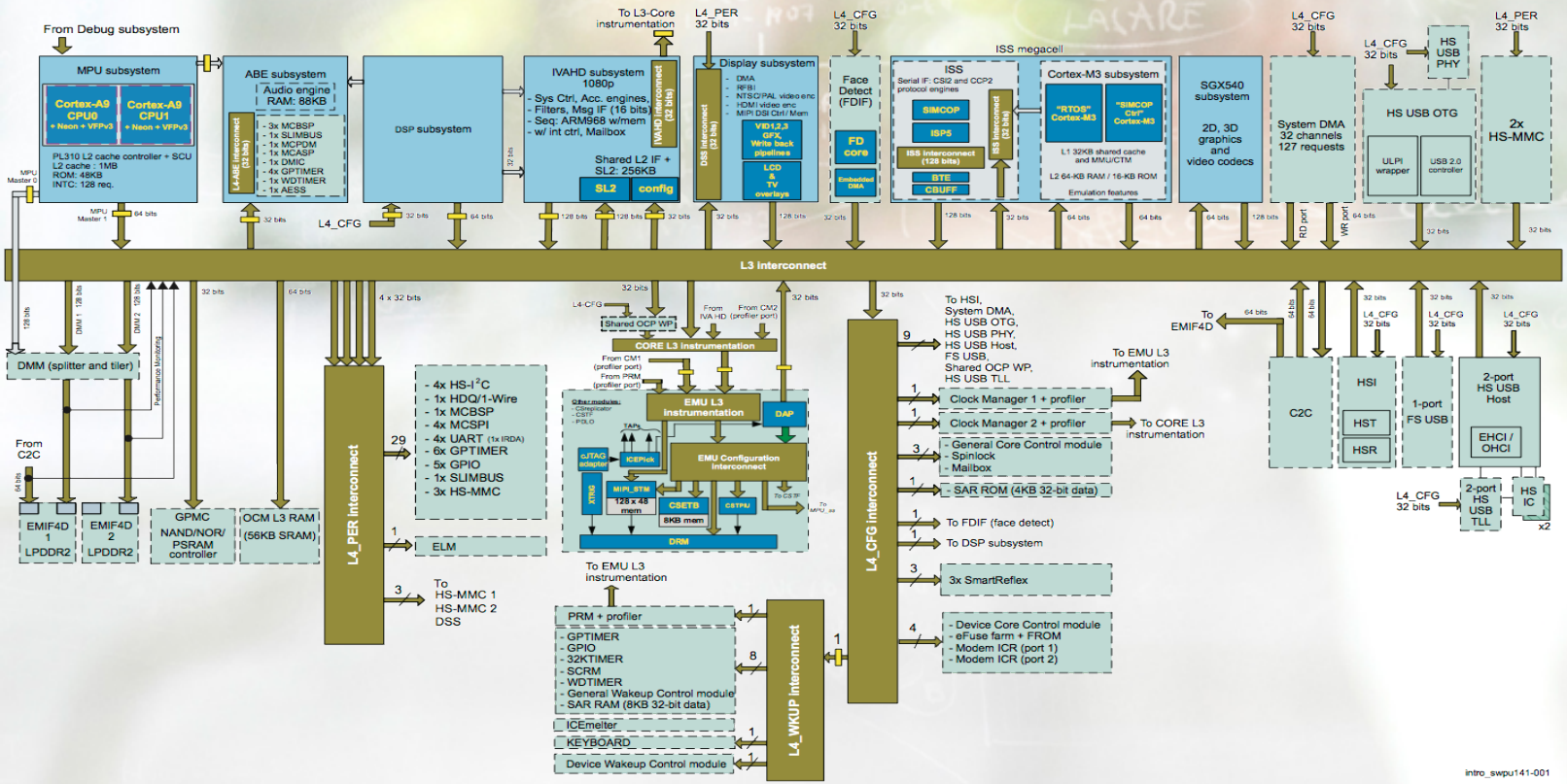
- Chip-producer
  - Texas Instruments
  - Qualcomm
  - ST Ericsson (?)
  - Samsung
  - NVIDIA (ARM+Tegra)
  - Apple (ARM+PowerVR)
- Current Maximum: 1.8 Ghz, 2 Mbytes of memory

PC architecture



Embedded System architecture

Figure 1-2. OMAP4430 Block Diagram



PC vs Embedded System

- PCs
  - Several, star-organized busses
- Mobile Gaming
  - One central bus

PC vs Embedded System

- PCs
  - Several, star-organized busses
  - Distributed memory and caches
- Mobile Gaming
  - One central bus
  - Centralized memory, small caches (if at all)

PC vs Embedded System

- PCs
  - Several, star-organized busses
  - Distributed memory and caches
  - Might be multicore, might have accelerators (GPU, modem) connected at periphery
- Mobile Gaming
  - One central bus
  - Centralized memory, small caches (if at all)
  - Multicore, contains tightly coupled accelerators (System-on-a-Chip

PC vs Embedded System

- PCs
  - Several, star-organized busses
  - Distributed memory and caches
  - Might be multicore, might have accelerators (GPU, modem) connected at periphery
- Optimized for Performance

- Mobile Gaming
  - One central bus
  - Centralized memory, small caches (if at all)
  - Multicore, contains tightly coupled accelerators (System-on-a-Chip
- Optimized for Efficiency

Embedded Systems Programming

- Generally
  - Code optimization very important

Embedded Systems Programming

- Generally
  - Code optimization very important
  - Bus and memory are bottlenecks, more so than in PCs

Embedded Systems Programming

- Generally
  - Code optimization very important
  - Bus and memory are bottlenecks, more so than in PCs
  - Only 32 (or 16) bit: avoid double precision

Embedded Systems Programming

- Generally
  - Code optimization very important
  - Bus and memory are bottlenecks, more so than in PCs
  - Only 32 (or 16) bit: avoid double precision
  - Use fixed-point instead of float wherever possible

Embedded Systems Programming

- Generally
  - Code optimization very important
  - Bus and memory are bottlenecks, more so than in PCs
  - Only 32 (or 16) bit: avoid double precision
  - Use fixed-point instead of float wherever possible
  - Use the multicores & accelerators

Embedded Systems Programming

- Generally
  - Code optimization very important
  - Bus and memory are bottlenecks, more so than in PCs
  - Only 32 (or 16) bit: avoid double precision
  - Use fixed-point instead of float wherever possible
  - Use the multicores & accelerators
  - Low-level programming if possible

Embedded Systems Programming

- iOS: Objective-C
- Android: JAVA

Embedded Systems Programming

- iOS: Objective-C
- Android: JAVA
- Both object-oriented

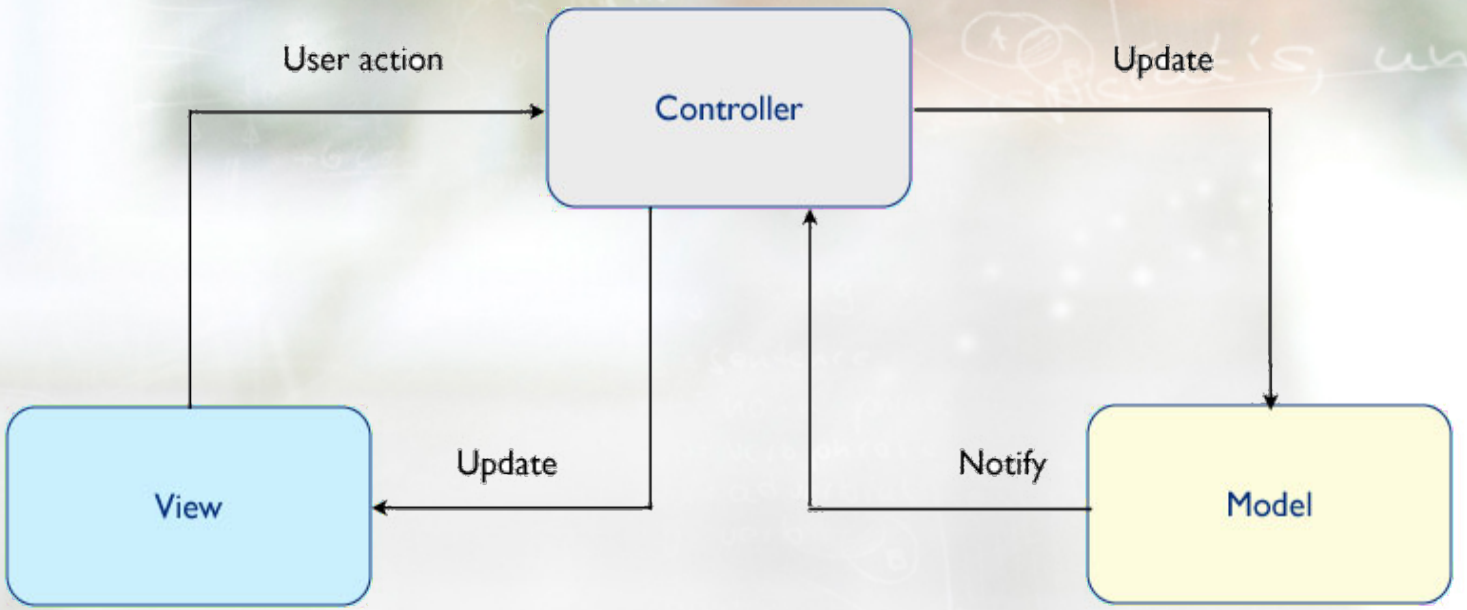
Embedded Systems Programming

- iOS: Objective-C
- Android: JAVA
- Both object-oriented
- Apps run in sandboxes (due to security reasons)

iOS

- Model: manages and modifies data
- View: renders to screen
- Controller: handles inputs and outputs

iOS



iOS

- Delegates
  - Do a task on behalf of another

iOS

- Advantages
  - Low-level programming possible
  - Easy-to-use toolchain, e.g. editor for screen layout, ready-to-use objects for user interaction

iOS

- Advantages
  - Low-level programming possible
  - Easy-to-use toolchain, e.g. editor for screen layout, ready-to-use objects for user interaction
- Disadvantages
  - Limited freedom

Android

- Services
- Content providers
- Broadcast receivers
- **Activities**

Android

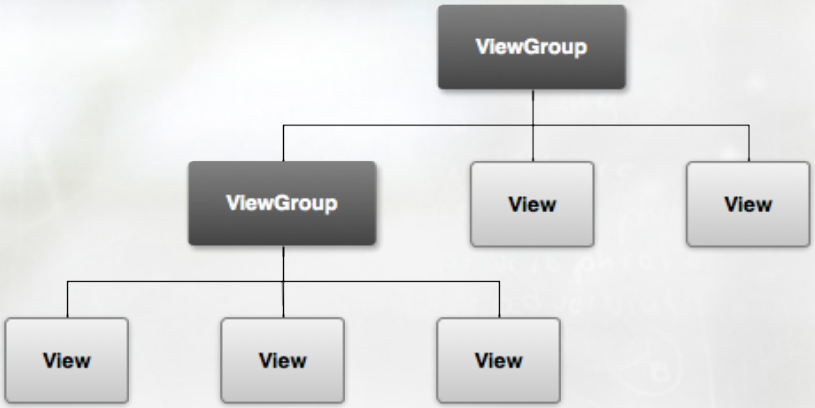
- Services
  - Run in background
  - Mostly computational

Android

- Content providers
  - Access to shared resources, e.g. file-system, network
- Broadcast receivers
  - e.g. rotation, energy-saving mode

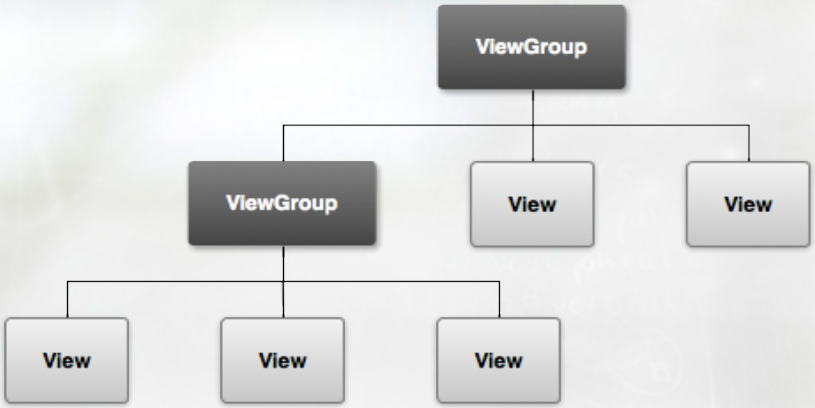
Android

- Activities
  - Control user in- and output
  - Several per screen possible
  - Contain at least on view



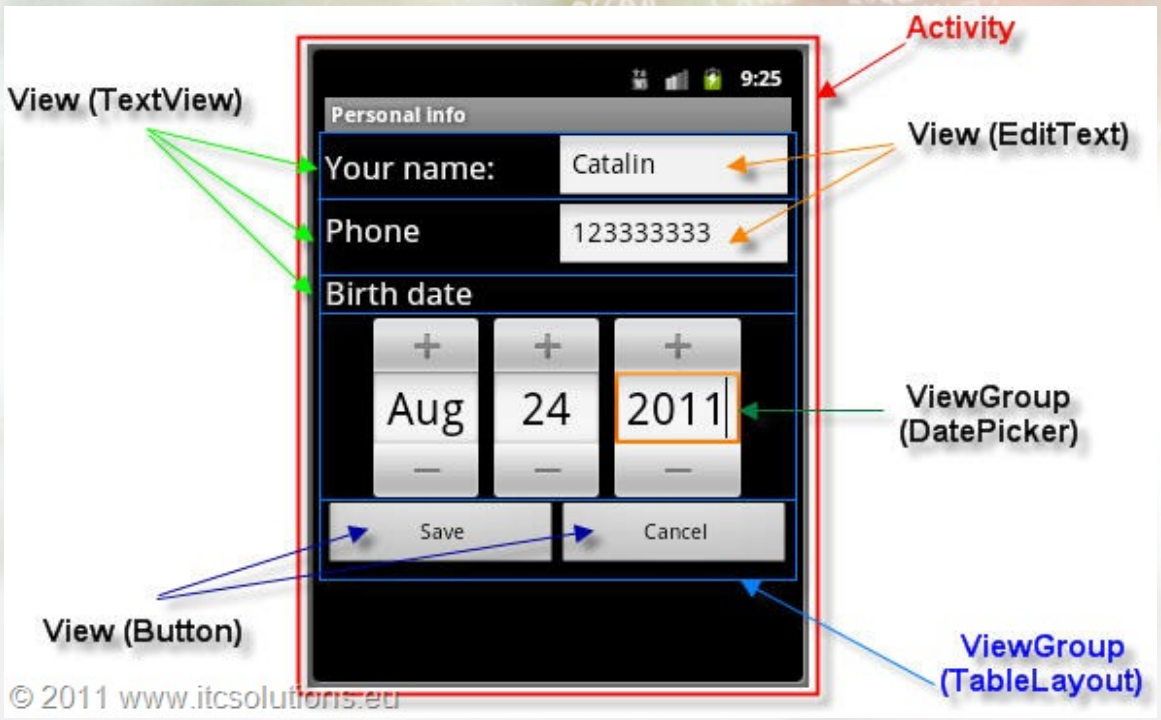
Android

- Activities
  - Control user in- and output
  - Several per screen possible
  - Contain at least on view



Android

- View / Viewgroups



Android

- JNI (Java Native Interface)
  - Low-level programming (C, C++) in Java
  - Often faster, but might not be
  - May have to be recompiled for different architectures
  - Complicated data- and thread-sharing

Android

- Advantages
  - More freedom
  - Works on different architectures (mostly)

Android

- Advantages
  - More freedom
  - Works on different architectures (mostly)
- Disadvantages
  - More complicated to use
  - Less easy-to-use tools
  - Java really good language for embedded systems? (Double as standard? No unsigned or fixed-point data-types?)

Object-oriented vs. data-driven

- Object-oriented
  - Data grouped by abstract objects

Object-oriented vs. data-driven

- Object-oriented
  - Data grouped by abstract objects
- Data-driven
  - Data grouped by access patterns

Object-oriented vs. data-driven

- Object-oriented
  - Data grouped by abstract objects
- Data-driven
  - Data grouped by access patterns
- Data-driven approach can be implemented by using objects

Concurrent Progamming

- Android
  - Up to developer
  - Threads, barriers, atomic commands etc.
  - Flexible, but can get complicated

Concurrent Progamming

- Android
  - Up to developer
  - Threads, barriers, atomic commands etc.
  - Flexible, but can get complicated
- iOS
  - Only asynchronous
  - Waiting queues, managed by the OS
  - Less freedom, but easy to use

Summary

- Android
  - More flexible, but more complicated
  - Odd language choice
  - Biggest, fastest growing market

Summary

- Android
  - More flexible, but more complicated
  - Odd language choice
  - Biggest, fastest growing market
- iOS
  - Easy to learn and use, but locked to the apple-way
  - Most important market?

Summary

- Android
  - More flexible, but more complicated
  - Odd language choice
  - Biggest, fastest growing market
- iOS
  - Easy to learn and use, but locked to the apple-way
  - Most important market?
- Other?

Summary

- Be aware of limitations
  - Optimize as much as possible
  - Avoid bus- and memory-usage
  - Be efficient! Try to make the most of it!

Summary