Electrical and Computer Engineering Design Procedure

Michael McGuire

Department of Electrical and Computer Engineering
University of Victoria

CENG/ELEC 399
Example ECE Design Procedure

System Engineering
Sub-system Engineering
Component Engineering
Integration Engineering
System Test

- High risk feedback
- Low risk feedback
- Feedback paths not shown are critical risk
System and Sub-system Engineering

- Decompose and define problem to be solved.
  - Research and investigate possible solutions.
  - Decide on sub-systems in design.
  - Educated guesses on technology potential.
  - Required component specifications calculated.

- Usually performed by the most experienced engineers.

- Example System Engineering: Personal Computer
  - Architecture of computer and bus specifications.

- Example Sub-system Engineering: Video system
  - Interface selection, required memory rate.
Component Engineering

- Design and implement each component.
  - Interface specifications from System engineering.
  - Design problem decomposed into sub-problems.
    - Select or design component to solve sub-problems.
  - Test that component meets specifications.

- Products of Design:
  - Design documents
  - Implementation
  - Test plans and results
  - Experimental test results of design.
Integration and System Test

- **Integration**
  - Get components to work together.
  - Check interactions of components with their neighbours in the system.

- **System Test**
  - Overall test of system.
  - Test to system’s user specifications *not* design specifications.
    - Many companies isolate system testers from the design team.
    - System test performed in environment as close to user environment as possible.
Technicians versus Engineers

- Technical work $\neq$ engineering design
  - Both are important.
  - Good technicians can also be good engineers.
  - However, the two skill sets are not identical.

- Construction worker versus Structural Engineer
  - Skyscrapers require expert construction/iron workers.
  - Construction workers do not design the skyscrapers.
  - Structural engineer is not an iron worker.

- These distinction is a less clear for software engineers.
  - Being able to program JAVA/C/C++ is not the only qualification to be a software engineer.
System Engineering Example: Smart Phone

System Engineering Selections/Decisions:

- Type of processor
- Bandwidth of system bus
- User interface of phone
- Size/capacity of battery
- Size of memory
- Software selections.
- Interfaces between components
Consequences of interface selections can last a long time:

- Microsoft’s limit of 640 kB conventional memory space for PCs
  - The limit of 640 kB simplified memory management interfaces.
  - Good decision for short term ‘one off’ project.
  - Bad decision for OS specification over long term.
- This mistake was made jointly by IBM and Microsoft.
System Engineering Example: Smart Phone

[Diagram showing various components of a smartphone system, including SD Card, SD Card Interface, Microprocessor, Power Management System, System Bus, Touch Panel, User Interface, Wireless LAN Radio, Video Processor, Buttons, Antenna, System Memory.]
Smart Phone System Engineering Considerations

- Different sub-systems affect each other
- Faster processor will change requirements on memory and power.
- Video system requirements linked to processor capability
- Fast interfaces are expensive in terms of power, space, and money
  - Full PCI interface on a mobile device?
  - Wide memory/system bus?
- Slow interfaces are cheaper
  - I2C is cheap but slow.
  - Short term gain of selection may affect future evolution.
“Slow Bus” uses I2C or SPI interface standard for Kbps or Mbps maximum data rates.
“Fast Bus” uses memory bus for > 10 Mbps or Gbps maximum data rates.
Two buses stop slow devices from blocking high speed device communications.
Integration Engineering

Interfaces between systems are the source of many if not most design failures.

- Misunderstandings of specifications/requirements for components.
- Components used in places where they are not well suited.
  - Software component for sorting may not scale as intended.
  - Lowpass filters for audio sampling should not be used for control applications.
    - Control requires linear phase while audio does not.
    - Shows in analog-to-digital conversion system design.
- Integration test verifies links between ‘neighbours’ in system.
System Test

- Verify that design system satisfies user requirements.
- ‘Engineers can be like cats and hide their own messes’
  - This is behaviour is not intentionally malicious
  - **Beware of this statement:** ‘No intelligent person would do this’
  - Read forums on Open Source software tools for examples of pitfalls from this kind of thinking.
Conclusions

- Design is an iterative process.
- System design should not be rushed.
- Testing is an important part of design.
  - You will have testing failures.