Introduction

Embedded system defⁿ: a computing system designed for a specific purpose
- made of purpose-built hardware & software
- examples: microwave oven, smart watch, engine control unit, nuclear reactor

- cars contain 70-100 electronic control units (ECUs) and ~100 million lines of code (Boeing 787 has 14 million)
source: Marius Mihailovic, Porsche Engineering Romania

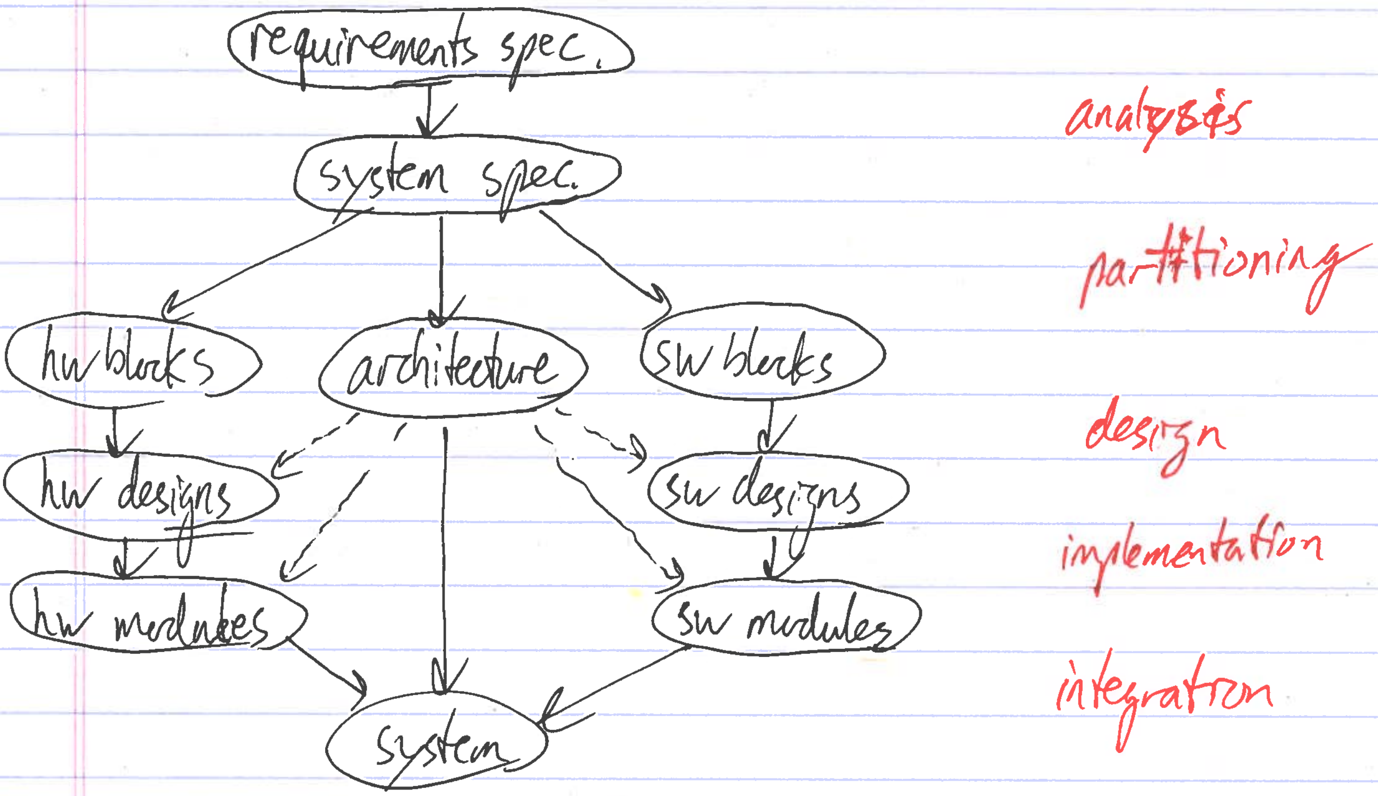
	safety critical	non-safety critical
cost of failure	high (e.g. loss of life)	low
real-time constraints	hard	soft or none
analysis	worst-case performance	average-case performance
verification	extensive testing / formal methods	testing (partial)
design goals	reliability certification	quality of service, cost, time-to-market
design cycle	long (e.g. 10-20 years for aerospace)	short (e.g. ~~[illegible]~~ 6 mo. - 1 yr for smartphones)

- hardware/software co-design defⁿ: the concurrent design of hardware and software so as to better meet design constraints such as cost, power, and performance

- hw/sw co-design flow

artifacts

action



1) requirements specification
   - functional requirements (what it should do)
     e.g. play back mpeg-1 video files
   - non-functional requirements (how it should do it)
     e.g. max cost, max power draw, min frame rate, min video file storage, max time to market

2) analysis
   - determine how the project can be realized
     e.g. algorithms, opportunities for parallelism

3) system specification (a.k.a. high-level spec.)
- implementation independent
- a set of functional blocks
- various models used (statecharts, system C) – possibly executable or simulatable to facilitate analysis/estimation

e.g. file selection, file parsing, image decoding, image output

4) hw/sw partitioning
- determine architecture (platform)
- partition functional blocks into hardware blocks and software blocks

5) architecture
- set of processing elements (PEs) that execute the functional blocks

e.g. FPGA for hw blocks (Zynq 7000 SoC programmable logic)

e.g. ISP for software blocks (Zynq 7000 SoC 650 Hz dual-core ARM A9 processor)

instruction-set processor

- interconnect between PEs

e.g. Advanced Microcontroller Bus Architecture – includes AXI, APB for connecting to hardware

- other hw elements such as memories, I/O

e.g. 512 MiB DDR3 memory, DMA controller, HDMI port

6) hw/sw design
- detailed specification for each functional block
e.g. statemachines and dataflow diagrams for hw
e.g. set of threads and UML diagrams for sw

7) implementation
- translate detailed specifications into executable modules
- high-level synthesis (HLS) is a way to automate this step

8) integration
- combine modules on the architecture

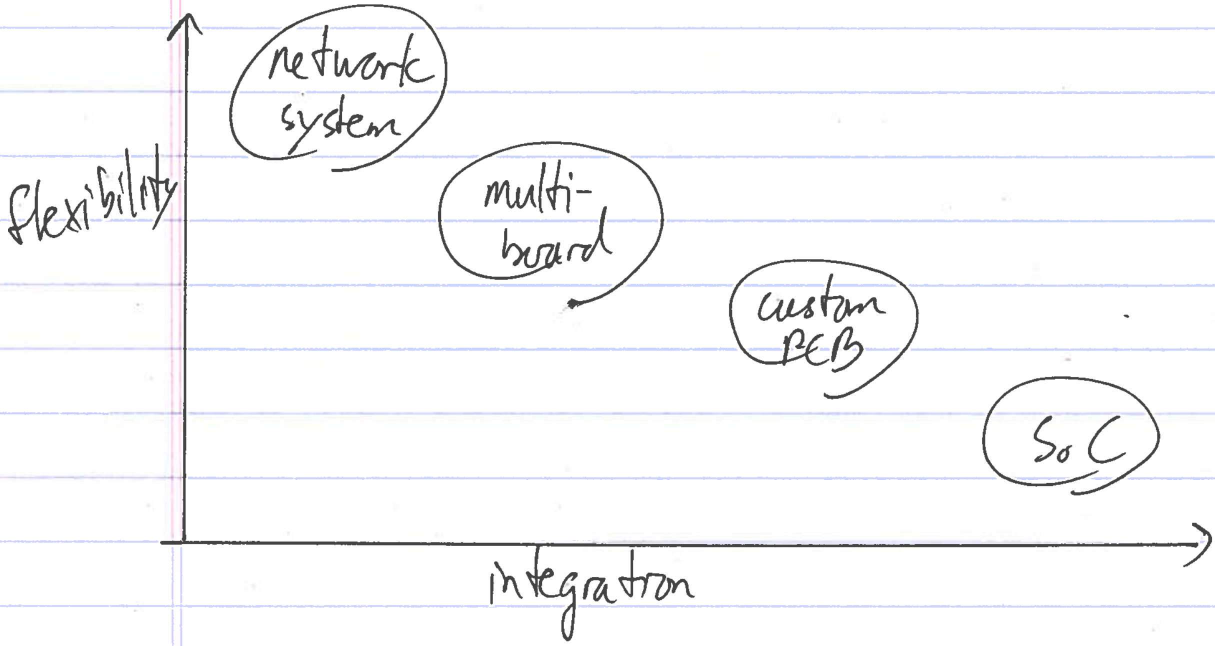
Design Technologies

1) Networked System
- processing nodes connected by off-chip network
  e.g. automotives: ECUs connected by CAN bus (controller area network)

2) Multi-board System
e.g. VMEbus chassis with compute, memory and I/O cards (superceded by VPX)
- widely used in aerospace and military applications
eg PC with PCIe expansion cards
e.g. Arduino expansion boards (shields)

3) Custom PCB
- discrete chips integrated on custom printed circuit board
~~e.g.~~ common in commercial applications
  e.g. dishwasher controller, rowing machine monitor

4) System-on-Chip (SoC)
- all components on same chip (processors, flash memory, I/O) - DRAM usually separate (different fabrication process)



- advantages of integration:
  - increased communication bandwidth
  - reduced cost, weight, energy consumption

SoC circuit technologies

① Field Programmable Gate Array (FPGA)
- reprogrammable (SRAM) or one-time-programmable (fused)
- lower design cost but higher per-unit cost
- lower performance (100's of MHz)
- good for prototyping and low-volume production

② Application Specific Integrated Circuit (ASIC)
- custom chip design
- very high design cost but lower per-unit cost
- higher performance (GHz) and lower power
- good for high-volume production