CPEG 422/622 EMBEDDED SYSTEMS DESIGN

MW 3:35 – 4:50, SHARP 116

Chengmo Yang

chengmo@udel.edu

Evans 201C

LECTURE 13
EMBEDDED SYSTEM DESIGN

OUTLINE

- Design overview
- Design metrics
  - Cost
  - Performance
  - Power
- Design methodology
- Various design flows

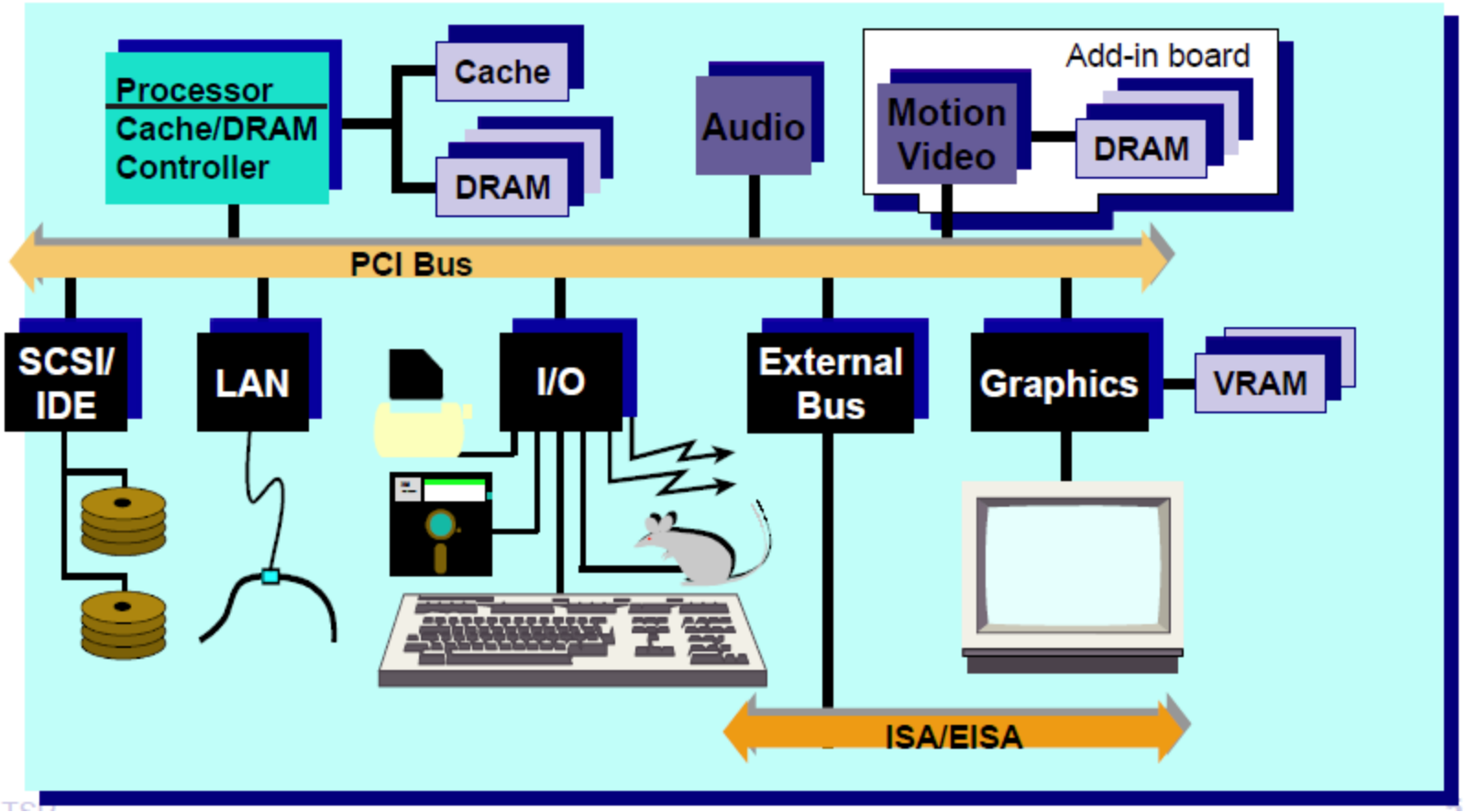
ES APPLICATION CLASSES

Class	Application	Processor	Requirements
Data flow	laser printers, X-terminals, routers, bridges, image processing	R4600, I960,29k, Coldfire, PPC (403, 605)	Processes data and passes it on. High memory bw, high throughput
Interactive video &portable	set-top boxes, videogames, PDAs, portable info appliances	R3900,R4100/4300/4600, ARM6xx/7xx, V851,SH1/2/3	Interactive, low cost, low power, high throughput
Classic embedded	controllers, disk controllers, automotive, industrial control	Piranha, ARM,MIPS, Cores	mix of CPU power, low cost, low power, peripherals

Design is purely application-driven

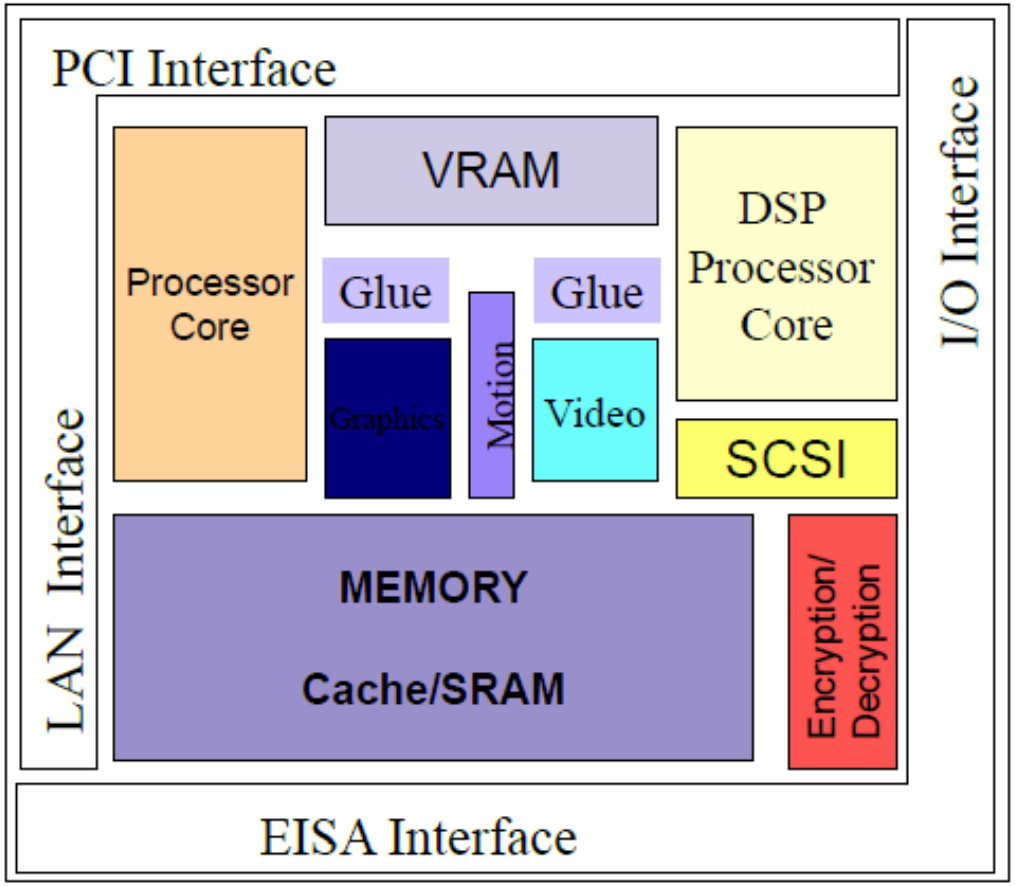
SYSTEM ARCHITECTURE: YESTERDAY

PCB design



A SYSTEM ARCHITECTURE: TODAY

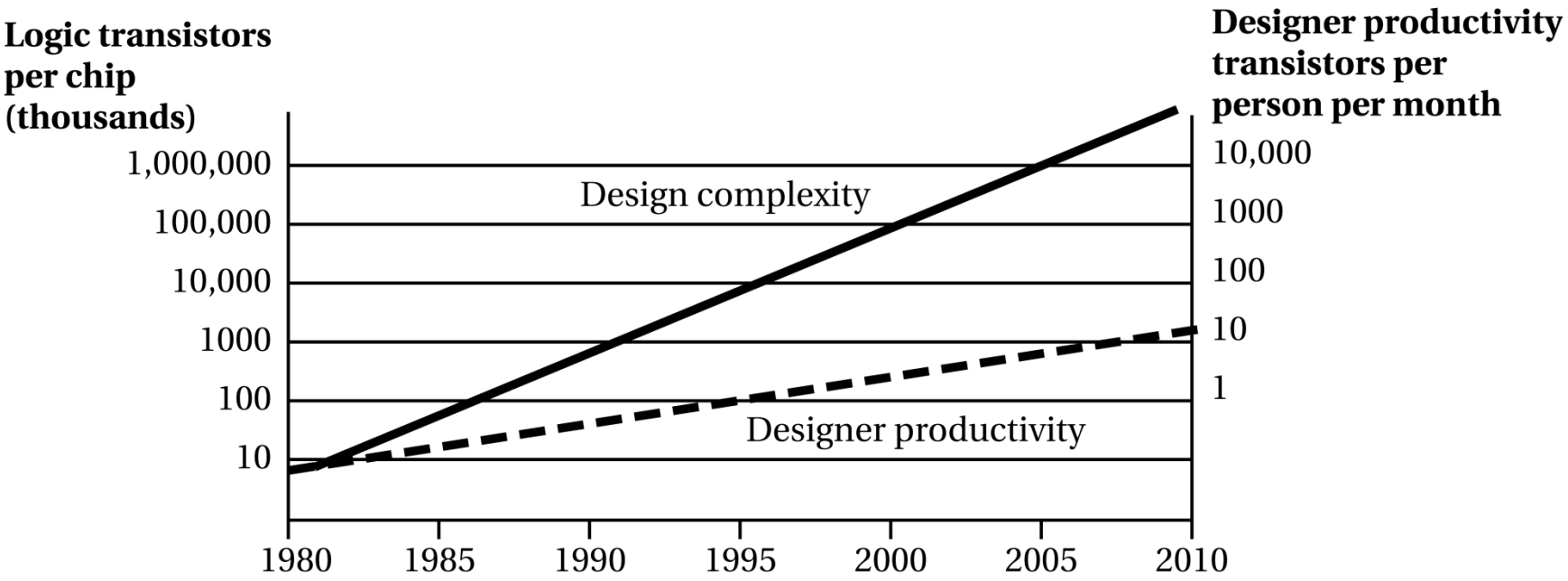
HW/SW Codesign of an SoC (System-on-Chip)



EMBEDDED SYSTEM DESIGN CHALLENGES

- Design space is large and irregular.
- We don't have synthesis tools for many steps.
- Can't simulate everything.
- Often need to start software development before hardware is finished.

DESIGN COMPLEXITY VS. DESIGNER PRODUCTIVITY



TRENDS IN EMBEDDED SYSTEMS

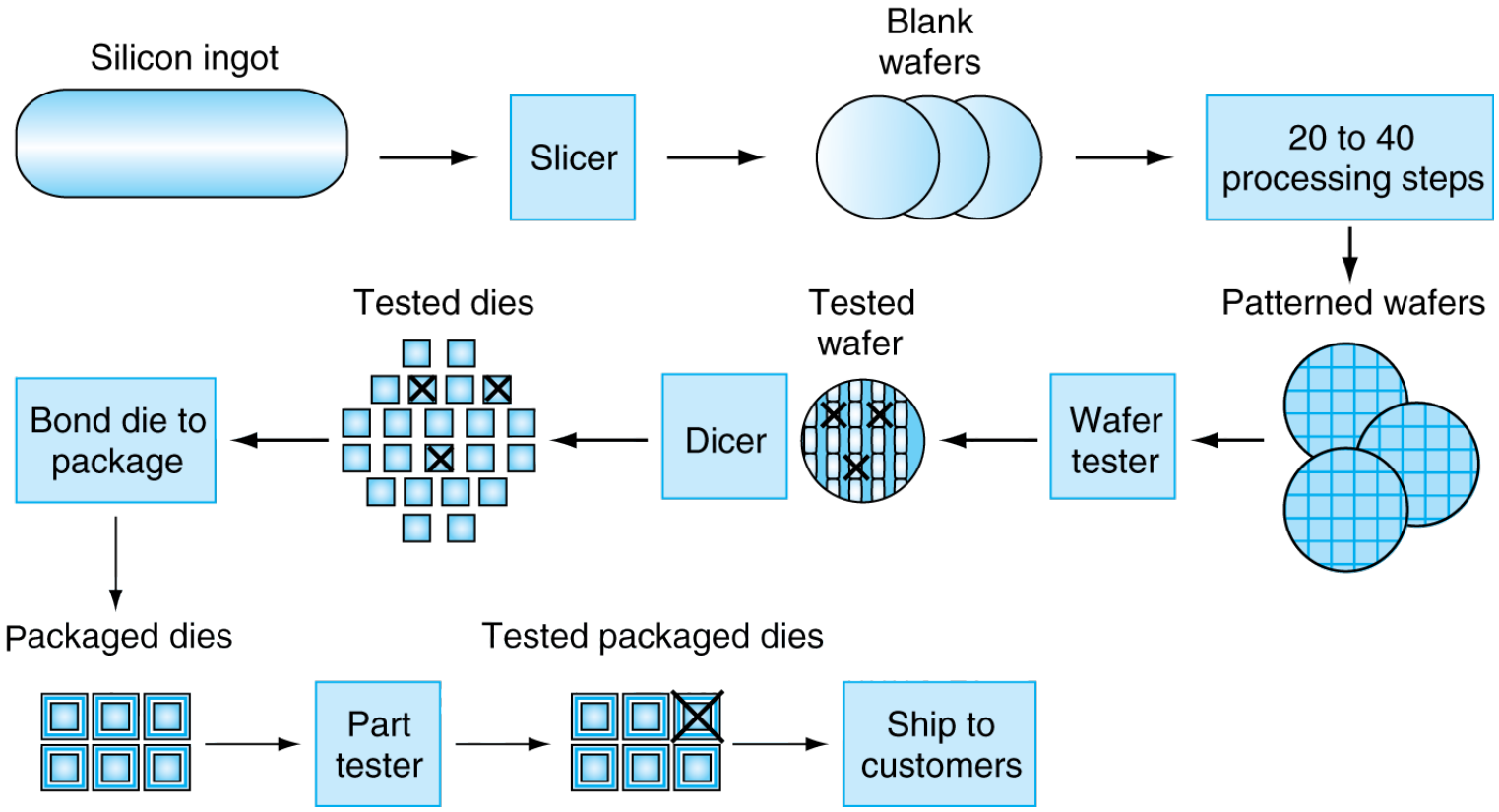
- Increasing code size
  - migration from hand (assembly) coding to high-level languages
- Reuse of hardware and software components
  - processors (micro-controllers, DSPs)
  - software components (drivers)
- Increasing integration and system complexity
  - integration of RF, DSP, network interfaces
  - 32-bit processors, IO processors (I2O)

Structured design and composition methods are essential.

DESIGN GOALS

- Functional requirements:
  - input/output relations.
- Non-functional requirements:
  - cost, performance, power, etc.

MANUFACTURING ICS



- **Yield:** **proportion of working dies per wafer**

IC MANUFACTURING COST

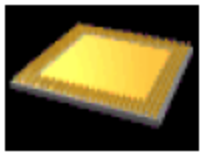
$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \times \text{Yield}}$$

$$\text{Dies per wafer} \approx \text{Wafer area}/\text{Die area}$$

$$\text{Yield} = \frac{1}{(1+(\text{Defects per area} \times \text{Die area}/2))^2}$$

- Nonlinear relation to area and defect rate
  - Wafer cost and area are fixed
  - Defect rate determined by manufacturing process
  - Die area determined by architecture and circuit design

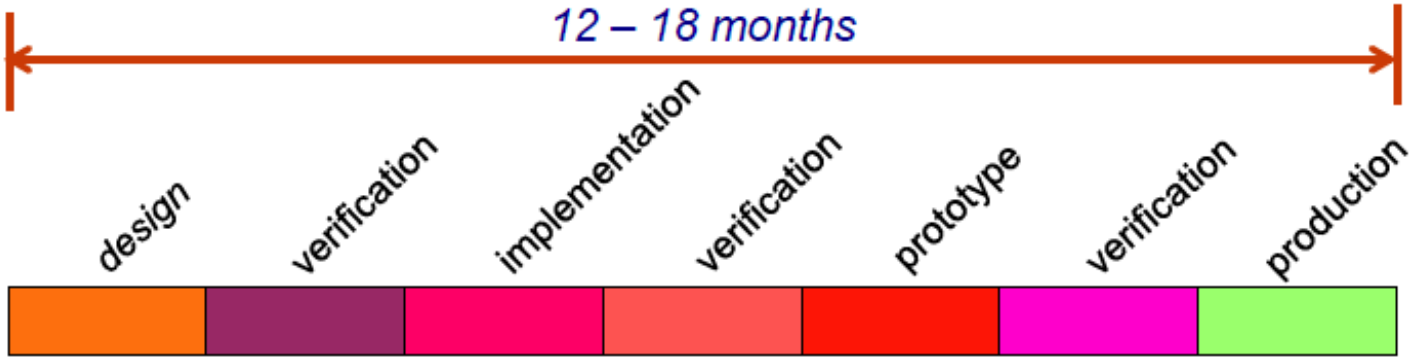
INCREASING CUSTOMIZATION COST



Estimated Cost - $85 M -$90 M

Example: Design with 80 M transistors in 100 nm technology

- Top cost drivers
  - Verification (40%)
  - Architecture Design (26%)
  - Embedded Design
    - 1400 man months (SW)
    - 1150 man months (HW)
  - HW/SW integration



IC COST ANALYSIS

- Cost breakdown:
  - Design/verification cost
  - Manufacturing cost
  - Testing cost
  - Maintenance/update cost

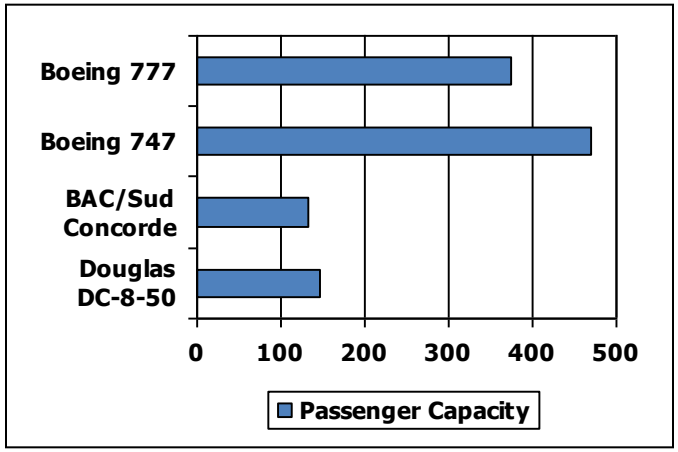
ASIC	FPGA
High	Low
Low	High
Same	Same
Same	Same

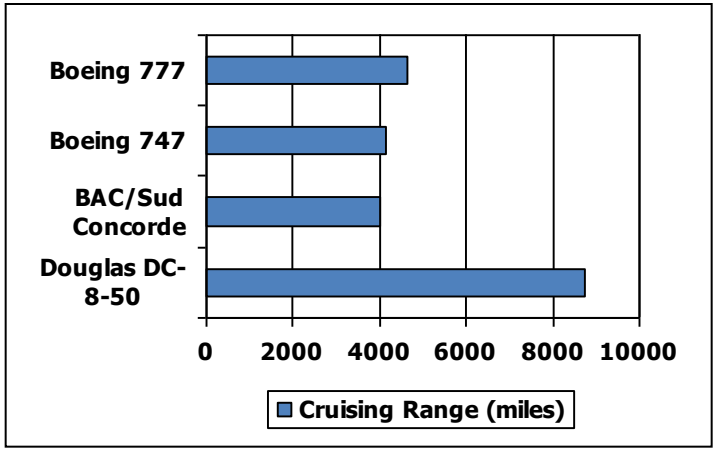
- Design and Manufacturing costs must be paid off across all the systems.
  - Small volume uses FPGA
  - Large volume uses ASIC

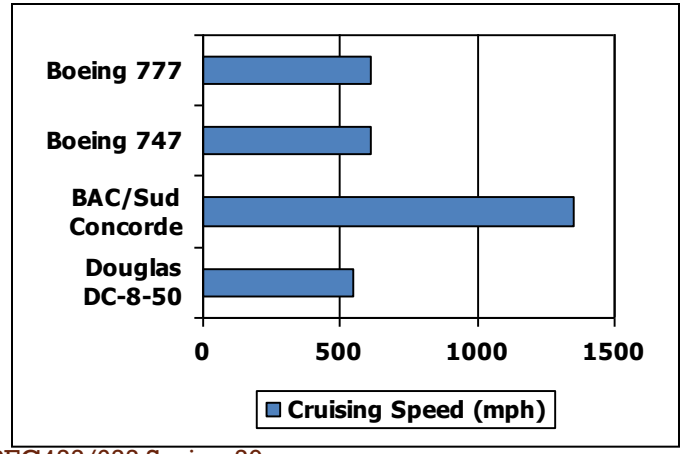
- Lifetime costs include software and hardware maintenance and upgrades.

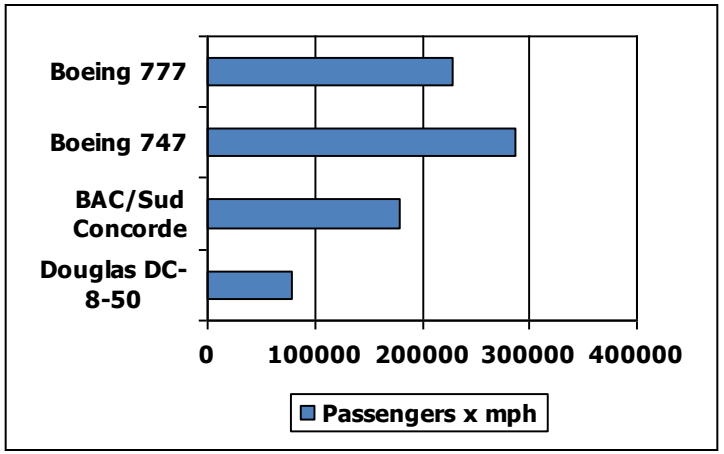
DEFINING PERFORMANCE

- Which airplane has the best performance?









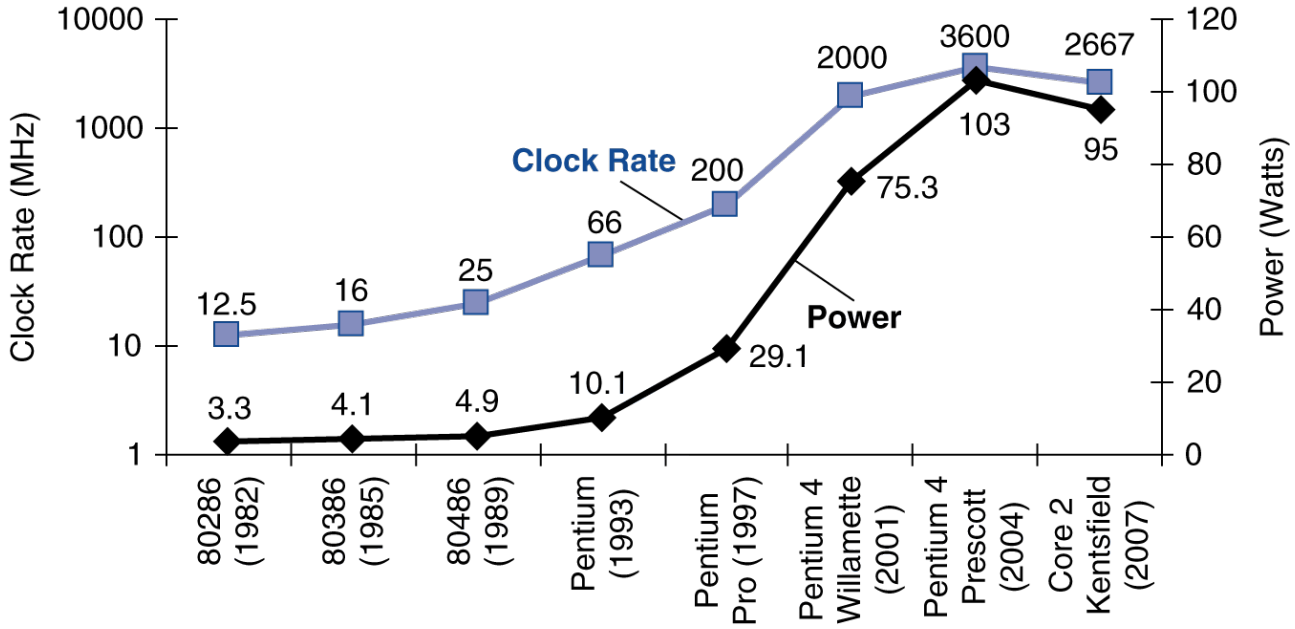
ASPECTS OF PERFORMANCE

- Embedded system performance can be measured in many ways:
  - Peak vs. sustained.
  - Average vs. worst/best-case.
  - Throughput vs. latency.

LATENCY VS THROUGHPUT

- Latency
  - How long it takes to do a task
- Throughput
  - Total work done per unit time
    - e.g., tasks/transactions/… per hour
- Notes:
  - One example, use pipeline will help throughput but not latency!
  - Similarly, use more processor will help throughput but not latency!

(DYNAMIC) POWER TRENDS



- In CMOS IC technology

$$\text{Power} = \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}$$

$$\text{Energy} = \text{Capacitive load} \times \text{Voltage}^2$$

ENERGY/POWER

- Energy consumption is important for battery life.
- Power consumption is important for heat generation or for generator-powered systems (vehicles).
- Notes:
  - Reduce the speed (frequency) can only save power but not energy!
  - Energy and power are the fundamental motivations for the adoption of non-volatile memories!

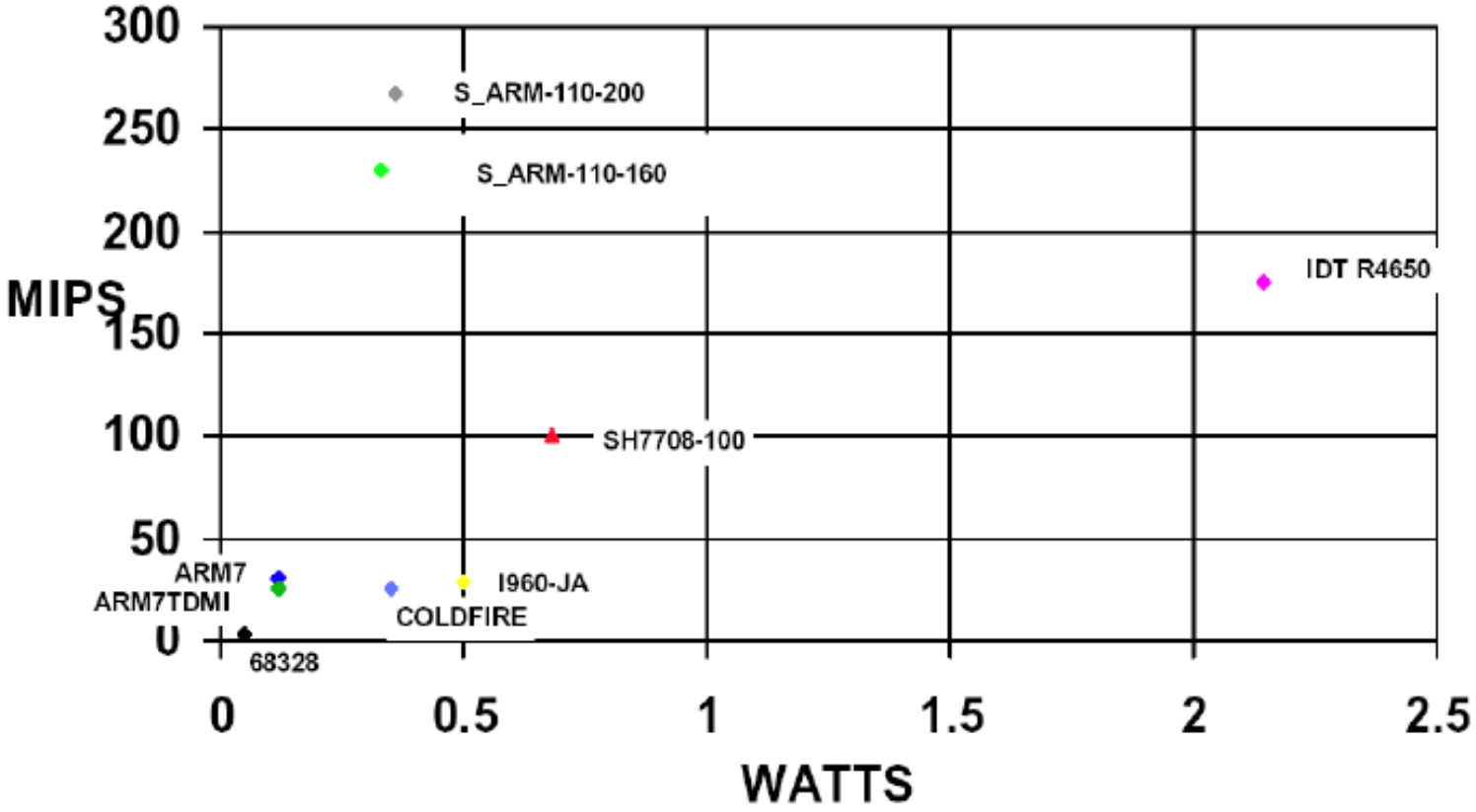
EMBEDDED SYSTEM METRICS

- Some metrics:
  - *performance*: MIPS, reads/sec etc.
  - *power*: Watts
  - *cost*: Dollars
    - Nonrecurring engineering cost, manufacturing cost
  - *size*: bytes, # components, physical space occupied

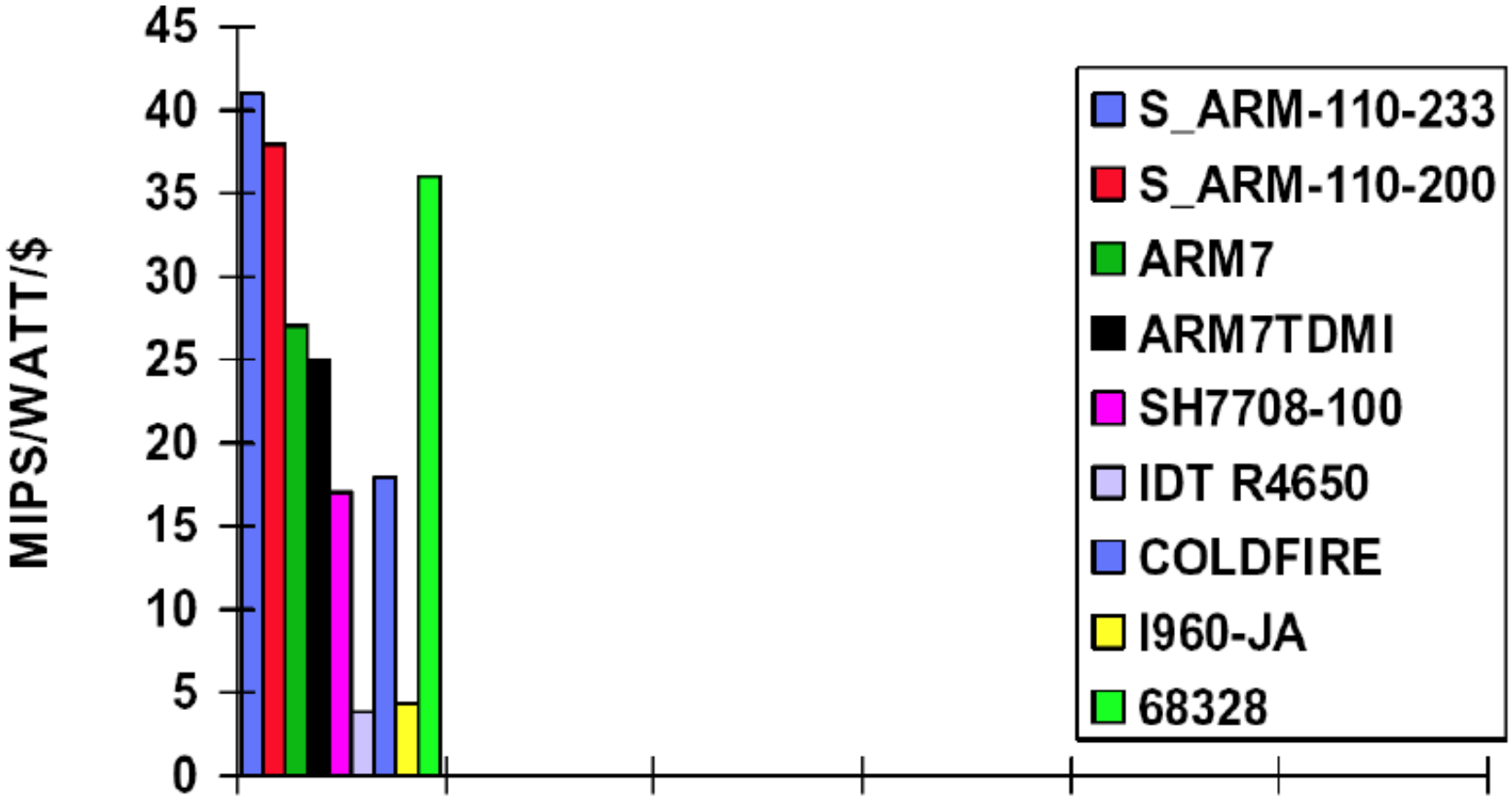
million instructions per second

- **MIPS, Watts and cost are related!**
  - technology driven
  - to get more MIPS for fewer Watts
    - look at the sources of power consumption
    - use power management and voltage scaling

MIPS VS. WATTS



MIPS/W/$


MIPS/WATT/$
45
40
35
30
25
20
15
10
5
0
S_ARM-110-233
S_ARM-110-200
ARM7
ARM7TDMI
SH7708-100
IDT R4650
COLDFIRE
I960-JA
68328

OTHER DESIGN ATTRIBUTES

- Design time must be reasonable. May need to finish by a certain date.
- System must be reliable; reliability requirements differ widely.
- Quality includes reliability and other aspects: usability, durability, etc.
- Security
- …

OUTLINE

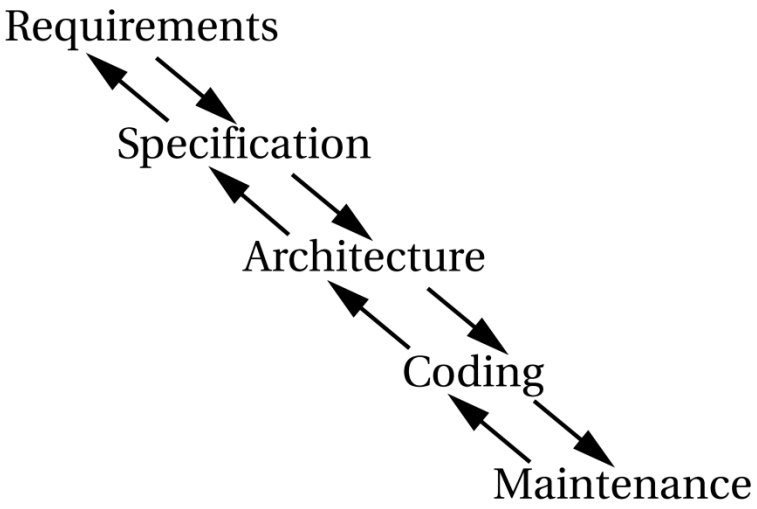
- Design overview
- Design metrics
  - Cost
  - Performance
  - Power
- **Design methodology**
- Various design flows

BASIC DESIGN METHODOLOGIES

- Design methodology:
  - A procedure for creating an implementation from a set of requirements.

- Basically, need to figure out flow of decision-making.
  - Determine when bottom-up information is generated.
  - Determine when top-down decisions are made.

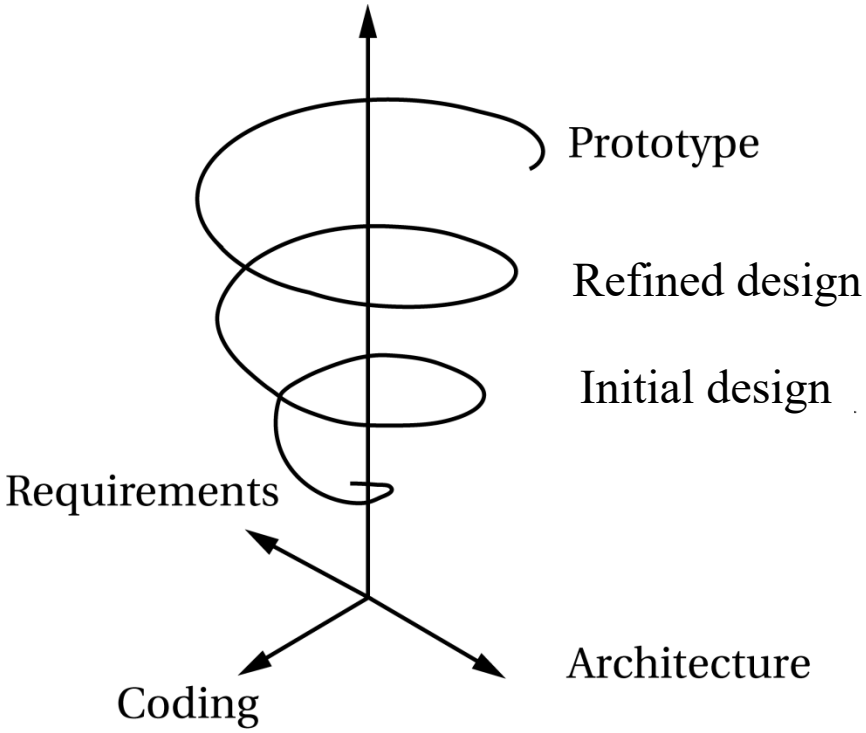
WATERFALL MODEL



Waterfall

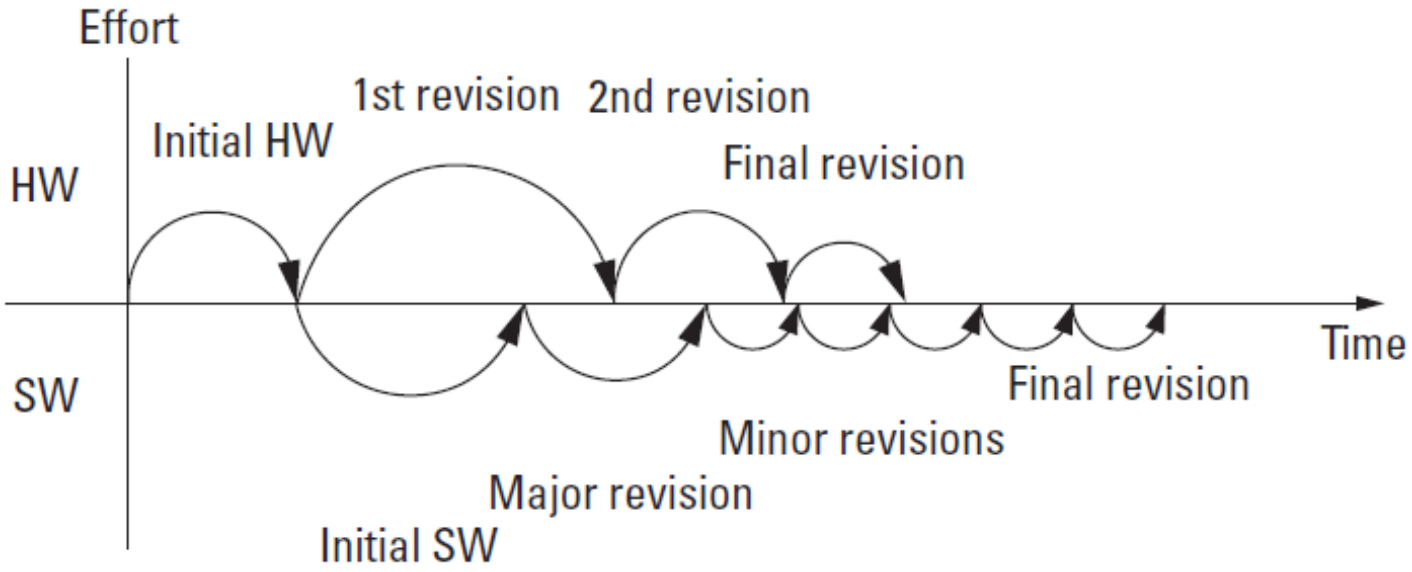
- Basic ideas:
  - complete each stage before advancing
  - the cost of going backwards (up the waterfall) to fix a mistake is very steep
- Mostly, this model helps a designer consciously think about organizing the project's activities

SPIRAL MODEL



Spiral

HARDWARE & SOFTWARE REVISION RATES



- **Hardware designers** usually have very expensive non-recurring costs; they tend to test extensively and revise cautiously
- **Software developers** have the advantage of fast turn around (i.e. compilation time is fast); they tend revise quickly