Introduction to Embedded Systems

EHB326E

Lectures

Prof. Dr. Müştak E. Yalçın

Istanbul Technical University

mustak.yalcin@itu.edu.tr

Embedded system

Billions of computing systems which are built every year for a very different purpose are embedded within larger electronic devices, repeatedly carrying out a particular function, often going completely unrecognized by the device's user!

An embedded system is nearly any computing system other than a desktop, laptop, or mainframe computer.

Embedded system

The Dozens of Computers That Make Modern Cars Go (and Stop) The New York Times, 2010. ▸ Link

... "It would be easy to say the modern car is a computer on wheels, but it's more like 30 or more computers on wheels," said Bruce Emaus, the chairman of SAE International's embedded software standards committee.

... IEEE Spectrum reported that electronics, as a percentage of vehicle costs, climbed to 15 percent in 2005 from 5 percent in the late 1970s — and would be higher today.

Embedded system

Some common characteristics of embedded systems;

- Single-functioned

  Executes a single program, repeatedly
- Tightly-constrained

  Low cost, low power, small, fast, etc.
- Reactive and real-time

  Continually reacts to changes in the system's environment

  Must compute certain results in real-time without delay

Design challenge

Obvious design goal

Construct an implementation with desired functionality

Key design challenge: Simultaneously optimize numerous design metrics.

What are relevant metrics ?

Design challenge

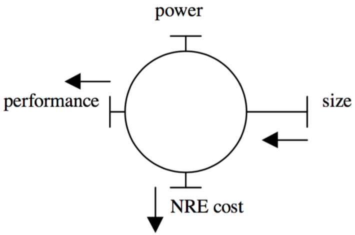
- Unit cost: the monetary cost of manufacturing each copy of the system, excluding NRE cost
- NRE cost (Non-Recurring Engineering cost): The one-time monetary cost of designing the system
- Size: the physical space required by the system
- Performance: the execution time or throughput of the system
- Power: the amount of power consumed by the system
- Flexibility: the ability to change the functionality of the system without incurring heavy NRE cost
- Time-to-prototype: the time needed to build a working version of the system
- Time-to-market: the time required to develop a system to the point that it can be released and sold to customers
- Maintainability: the ability to modify the system after its initial release
- Correctness, safety, many more

Apple's iPhone 6s Plus Costs an Estimated $236 to Make – $749 to Purchase ▸ Link

Costs for new product development: wages, supplies, proprietary capital equipment construction costs, etc.

Design challenge

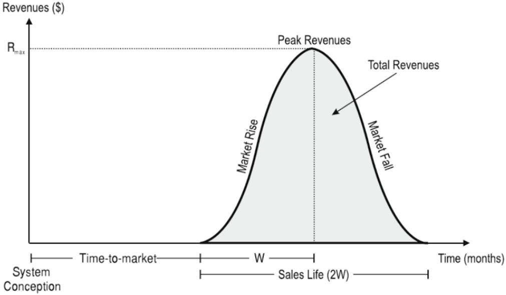
Design metric competition – improving one may worsen others



For example, if we reduce an implementation's size, its performance may suffer.

Design challenge

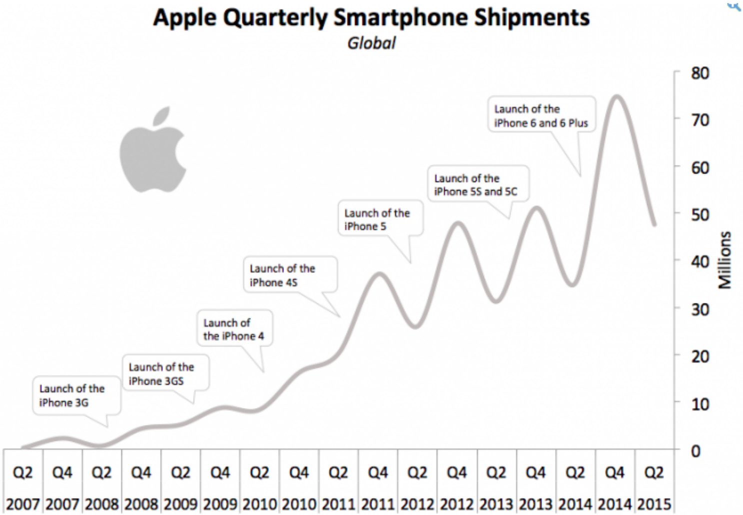
hre Most of these metrics are heavily constrained in an embedded system ! Expertise with both software and hardware is needed to optimize design metrics.



market time-windows for products are becoming quite short, often measured in months.

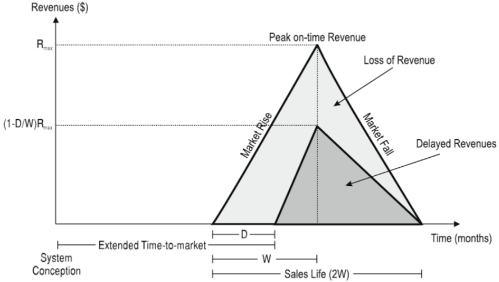
Design challenge

market time-windows



▸ Link

Design challenge



On-time $= \frac{1}{2} \times 2 \times W \times W$

Delayed $= \frac{1}{2} \times (2 \times W - D) \times (W - D)$

Percentage revenue loss(%) $= \frac{W \times (3 \times W - D)}{2 \times W^2}$

Lifetime 52 wks, delay D=10 wks: $(10 \times (3 \times 26\text{–}10)/2 \times 26^2) = 50\%$

NRE and unit cost metrics

- Unit cost
- NRE cost
- Total cost
- per-product cost

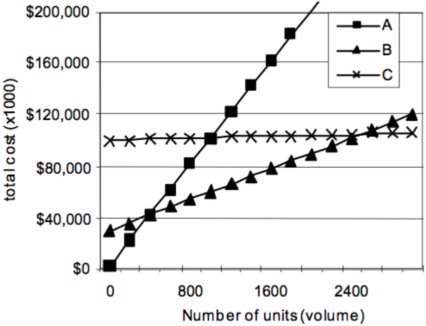
Definition

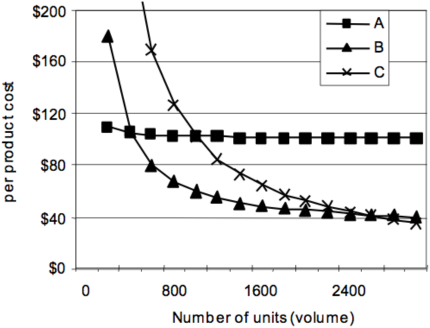
The monetary cost of manufacturing each copy of the system, excluding NRE cost. The one-time monetary cost of designing the system. =NRE cost + unit cost * # of units = total cost / # of units=(NRE cost / # of units) + unit cost

Example:
NRE=\$2000, unit=\$100
For 10 units
total cost = \$2000 + 10*\$100 = \$3000
per-product cost = \$2000/10 + \$100 = \$300

Compare technologies by costs – best depends on quantity

- Technology A: NRE=$2,000, unit=$100
- Technology B: NRE=$30,000, unit=$30
- Technology C: NRE=$100,000, unit=$2





But, must also consider time-to-market !

The performance design metric

- Widely-used measure of system, widely-abused
  Clock frequency, instructions per second – not good measures
  Digital camera example – a user cares about how fast it processes images, not clock speed or instructions per second
- Latency (response time)
  Time between task start and end
  e.g., Camera's A and B process images in 0.25 seconds
- Throughput
  Tasks per second, e.g. Camera A processes 4 images per second
  Throughput can be more than latency seems to imply due to concurrency, e.g. Camera B may process 8 images per second (by capturing a new image while previous image is being stored).
- Speedup of B over S = B's performance / A's performance
  Throughput speedup = 8/4 = 2