

Sisteme Încorporate (Embedded Systems)

Cursul 1
Intro

Ce au în comun o mașină de spălat și un Boeing 747?



Amândouă sunt sisteme încorporate.

- Definiție
 - Un sistem de calcul încorporat în dispozitivele electronice.
- Sistemul implementează o anumită funcționalitate
 - Este o combinație de hardware și software proiectată să îndeplinească o funcție anume.

Vedere de ansamblu

- Sisteme de calcul încorporate
 - Greu de definit. Aproape toate sistemele diferite de un calculator desktop/laptop obișnuit
 - Miliarde de unități produse anual vs. milioane de unități pentru sistemele desktop
 - În jur de 50 sisteme într-o locuință sau într-un automobil obișnuit
 - Preț semnificativ mai mic decât al unui sistem desktop
 - Putere de calcul redusă (nu în toate cazurile)

Exemple Obișnuite

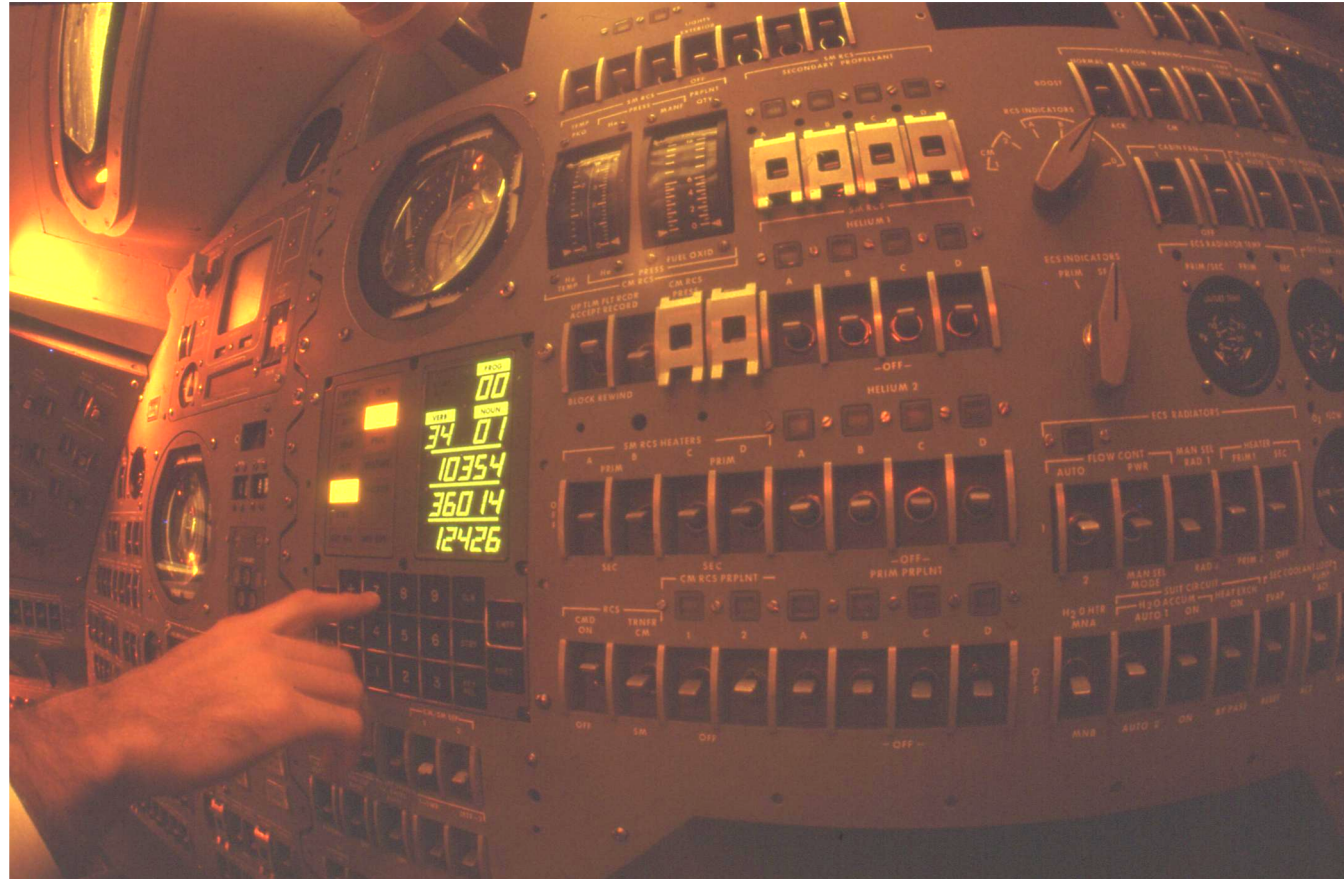


Mai multe exemple

- Sisteme de procesare semnal
 - radar, sonar, real-time video, playere MP3/CD/DVD, echipament medical
- Sisteme critice
 - avionică, echipamente de control al navelor spațiale, controlul reactoarelor nucleare
- Sisteme de control distribuit
 - Routere și switch-uri de rețea, sisteme de transport în masă
- Sisteme “mici”
 - Telefoane celulare, electrocasnice, jucării, carduri, playere MP3, PDA-uri, camere digitale, senzori, ceasuri digitale, receptoare GPS

Primul Sistem Embedded

AGC – construit de MIT Instrumentation Laboratory ('60)



Apollo Guidance Computer

20 Iulie 1969 – Primul Computer care a ajuns pe Lună

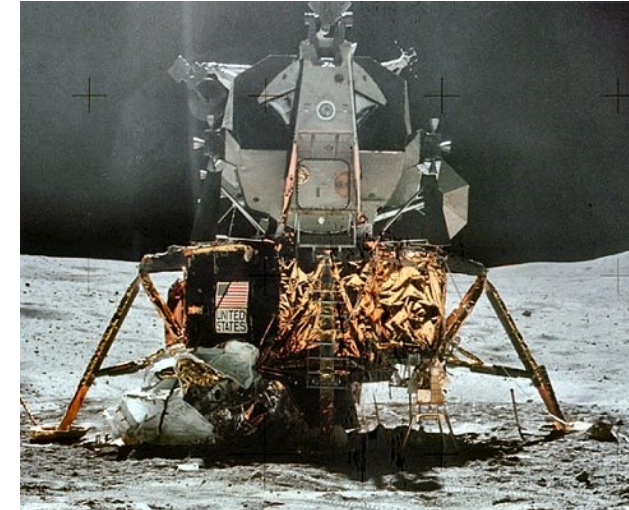
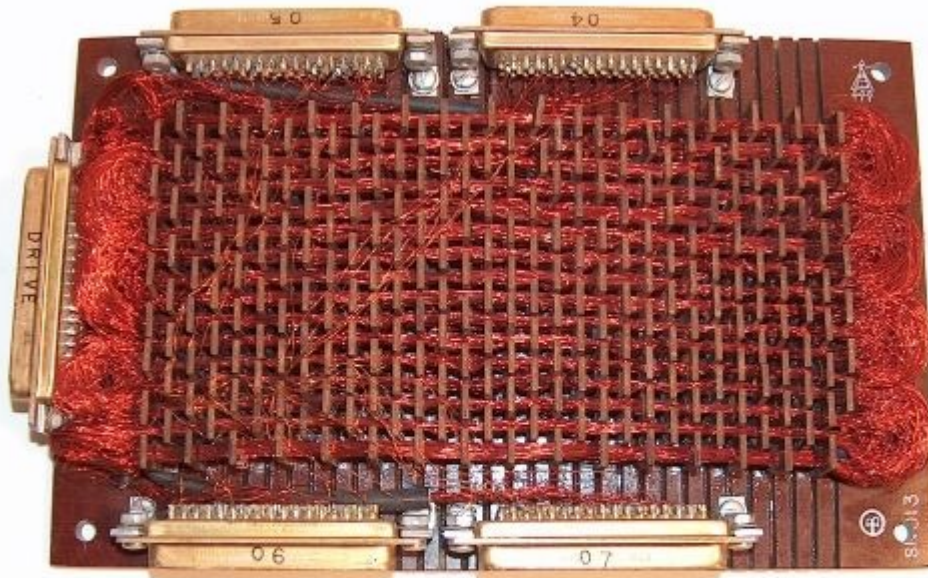
Specificații:

Frecvența de ceas: 1,024 MHz

Memorie RAM: 8KB memorie cu miez de ferită

Memorie program: 64KB ROM Core Rope

Primul sistem de calcul care a folosit circuite integrate



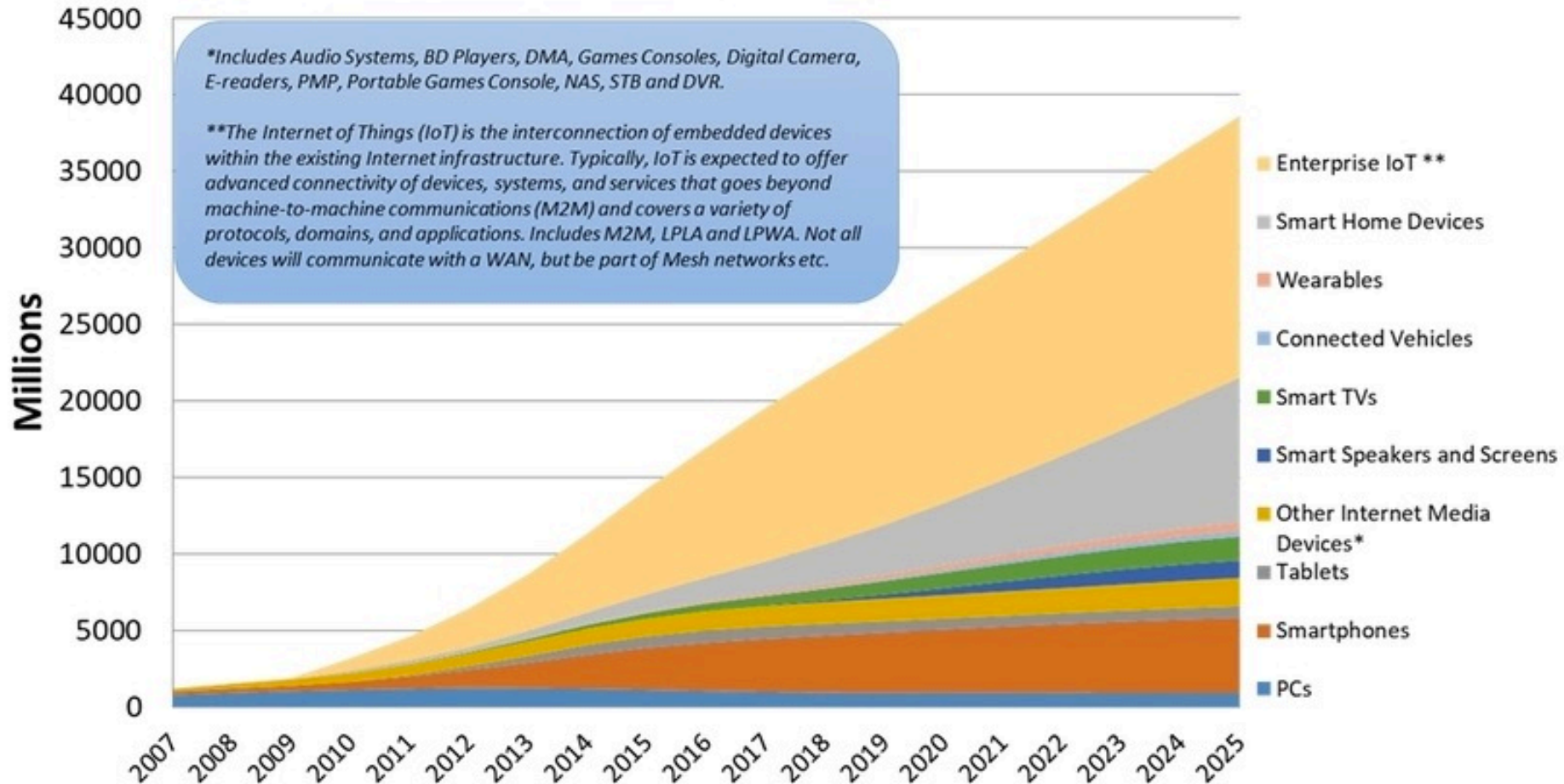
- Are o singură funcție
 - Execută un singur program, repetat
- Supus la constrângeri
 - Cost redus, consum mic de electricitate, dimensiuni, viteză de execuție, etc.
- Reacționează la evenimentele de timp real
 - Reacționează la schimbările din mediul înconjurător
 - Trebuie să obțină rezultate în timp real cu un minim de întârzieri

De ce ne pasă? Câteva cifre de piață

- Echipamentele embedded înlocuiesc PC-urile
 - Au diferite forme și funcții specializate față de multifuncționalitatea unui PC.
 - Anul 2002: mai mult de 50% din dispozitivele de acces la Internet NU sunt PC-uri.
 - În 1997, 96% din dispozitivele de acces la Internet erau PC-uri
 - 2020 – 24 miliarde de echipamente embedded legate la Internet
- Sistemele tradiționale depind din ce în ce mai mult de sisteme de calcul
 - Automobilele moderne conțin aproape 100 de procesoare care rulează software complex
 - Reglarea combustiei, controlul noxelor emise, sistem anti-derapare, monitorizarea consumului, transmisia automată, gestionarea afișajului de bord etc.
 - <http://www.howstuffworks.com/car-computer.htm>

Global install base

Global Connected and IoT Device Installed Base Forecast








Source – Strategy Analytics research services, May 2019: IoT Strategies, Connected Home Devices, Connected Computing Devices, Wireless Smart Home Strategies, Wearable Device Ecosystem, Smart Home Strategies

Connected devices

Connected devices (billions)

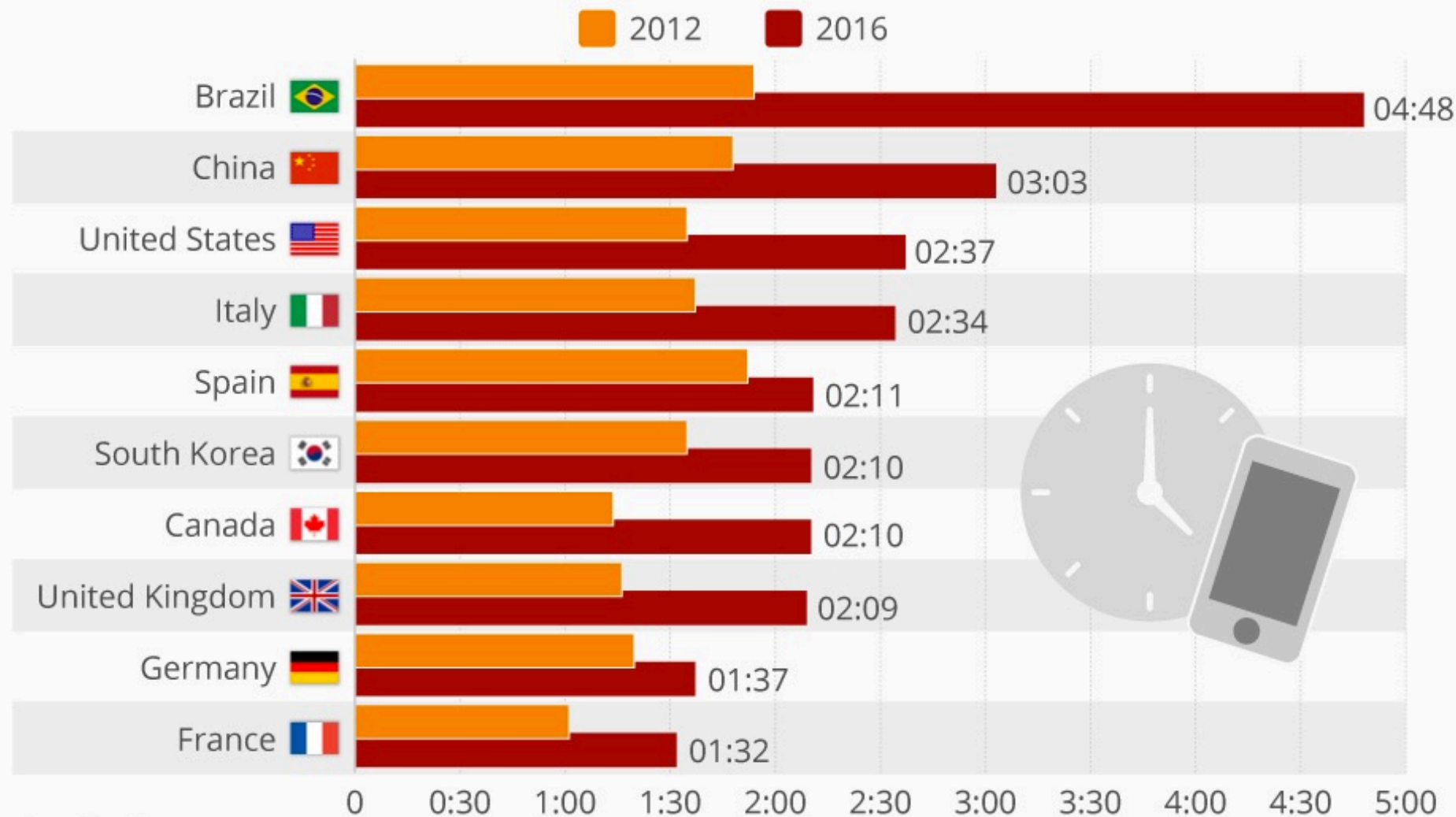


	2016	2022	CAGR
 Wide-area IoT	0.4	2.1	30%
 Short-range IoT	5.2	16	20%
 PC/laptop/tablet	1.6	1.7	0%
 Mobile phones	7.3	8.6	3%
 Fixed phones	1.4	1.3	0%
	16 billion	29 billion	10%

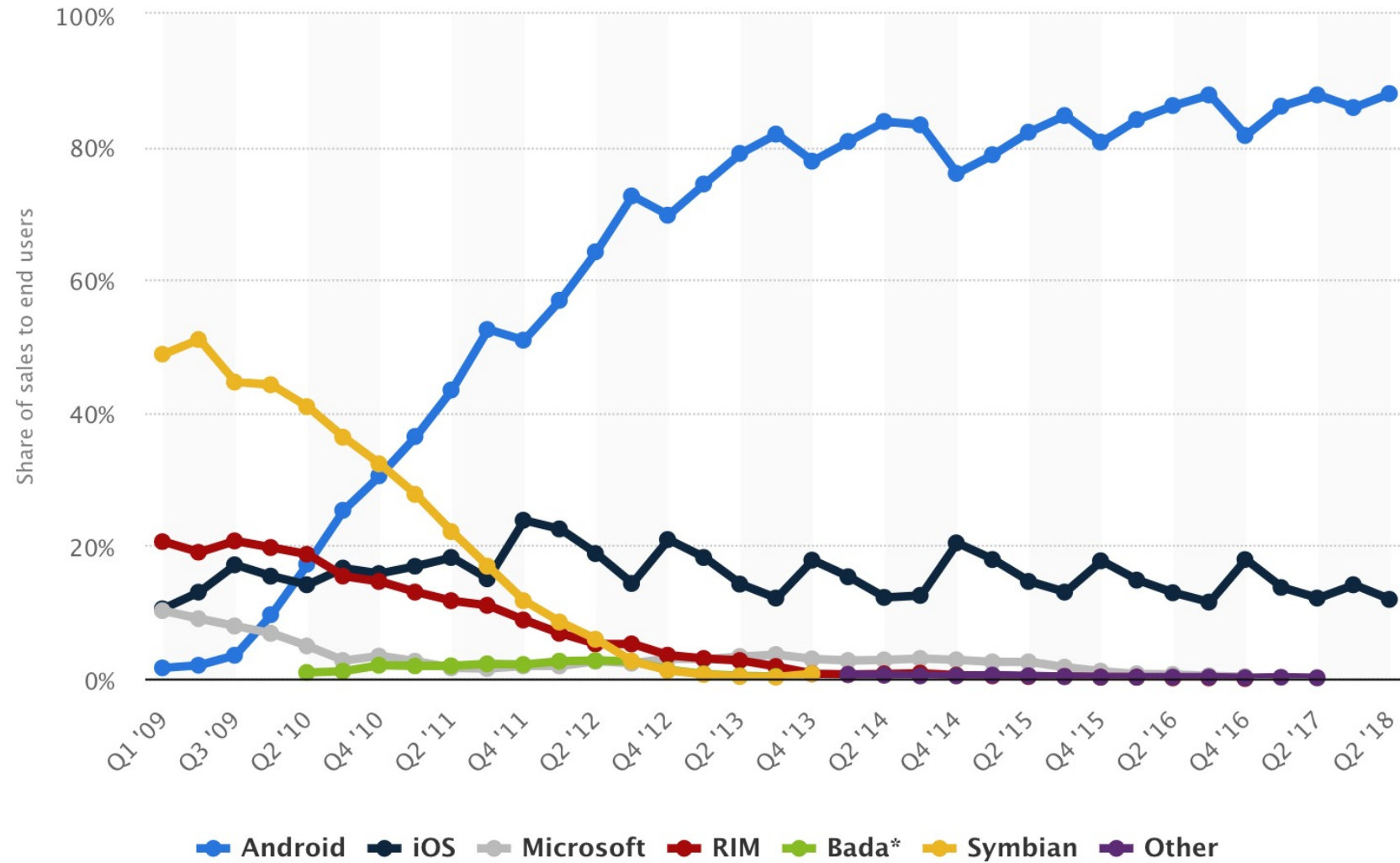
<https://www.ericsson.com/en/mobility-report/internet-of-things-forecast>

Smartphone Addiction Tightens Its Global Grip

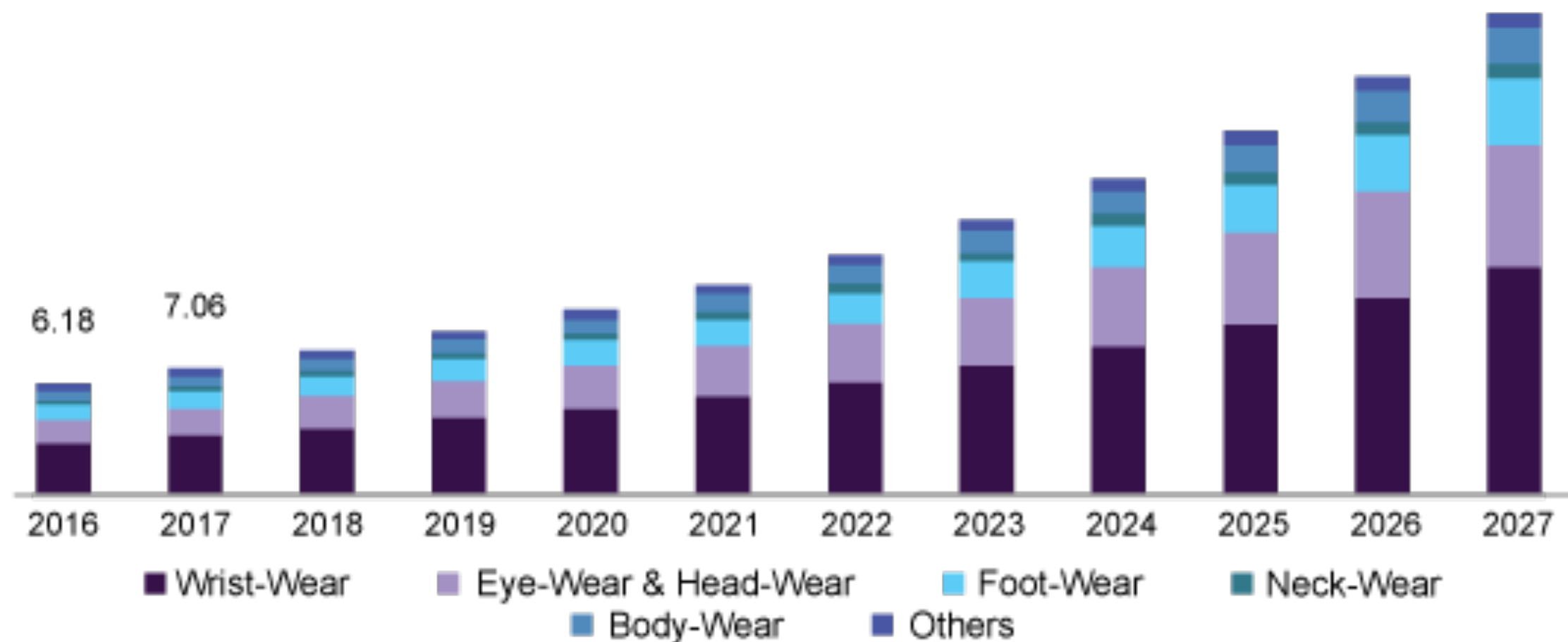
Hours per day spent online via a mobile device, per user



Smartphone sales by OS



U.S. wearable technology market size, by product, 2016 - 2027 (USD Billion)



Source: www.grandviewresearch.com

Lucrurile private din perspectiva CS

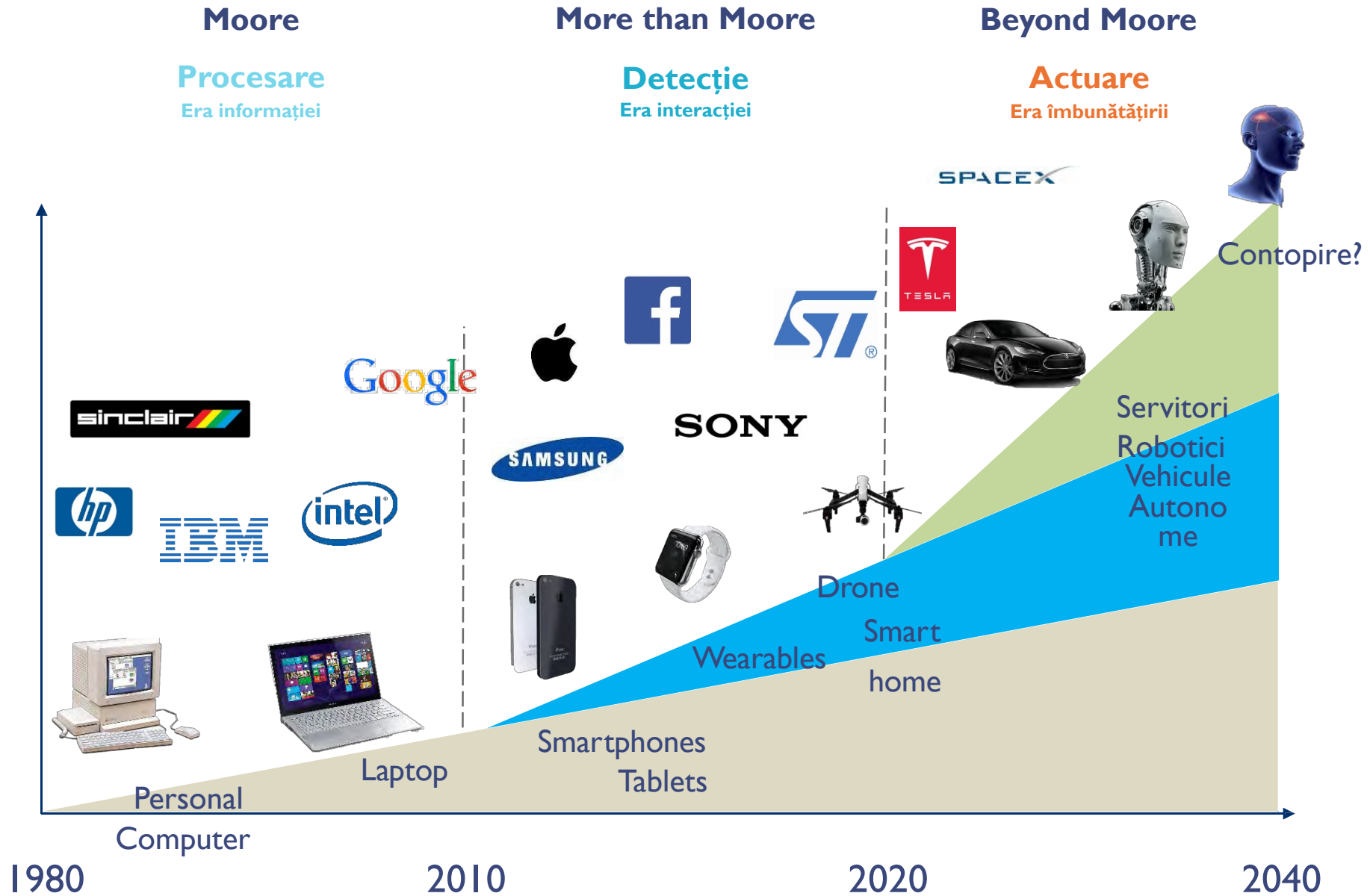
- În trecut știința calculatoarelor nu era interesată de domeniul embedded
 - Cantitate limitată de software
 - Programare la nivel de bază
- În prezent există un interes în creștere pentru cercetare și software development în CS
 - Creșterea complexității sistemelor embedded necesită sistematizarea proiectării

Legea lui Moore (1965)

- Numărul tranzistoarelor de pe 1 cm pătrat
 - De două ori mai mulți după ~1.5-2 ani
- Tendințe asociate
 - Performanța procesoarelor
De două ori mai rapide după ~18 luni
 - Capacitatea memoriei
Se dublează după < 2 ani

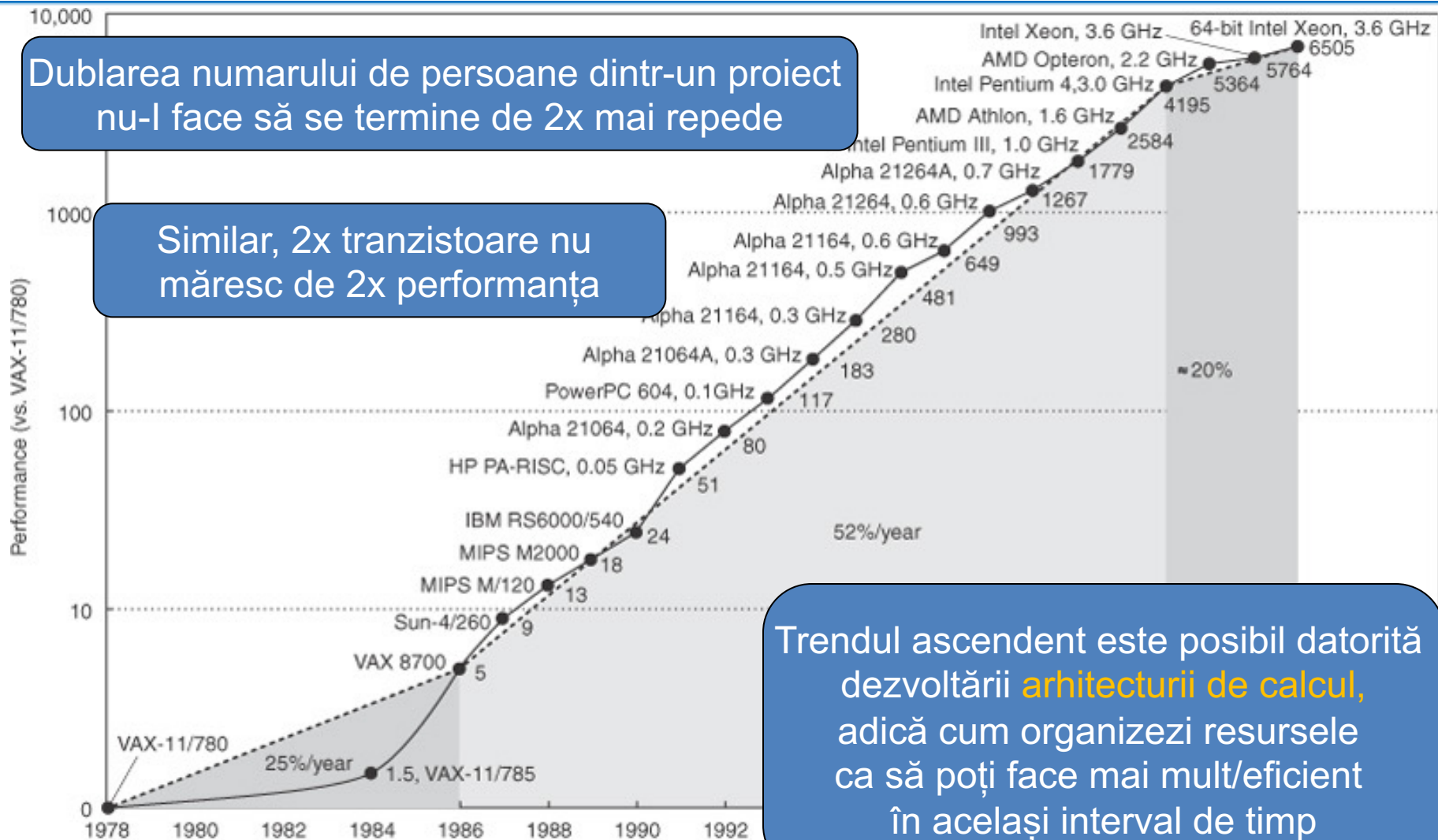
Moore's Not-Exactly-Law

- Nu este o lege a naturii
 - Dar aproximează destul de bine ce s-a întâmplat în industrie în ultimii 45 de ani
- No exponential is forever but we can delay “forever” (Gordon Moore in 2003)
- Mai multe despre legea lui Moore la <http://www.intel.com/research/silicon/mooreslaw.htm>

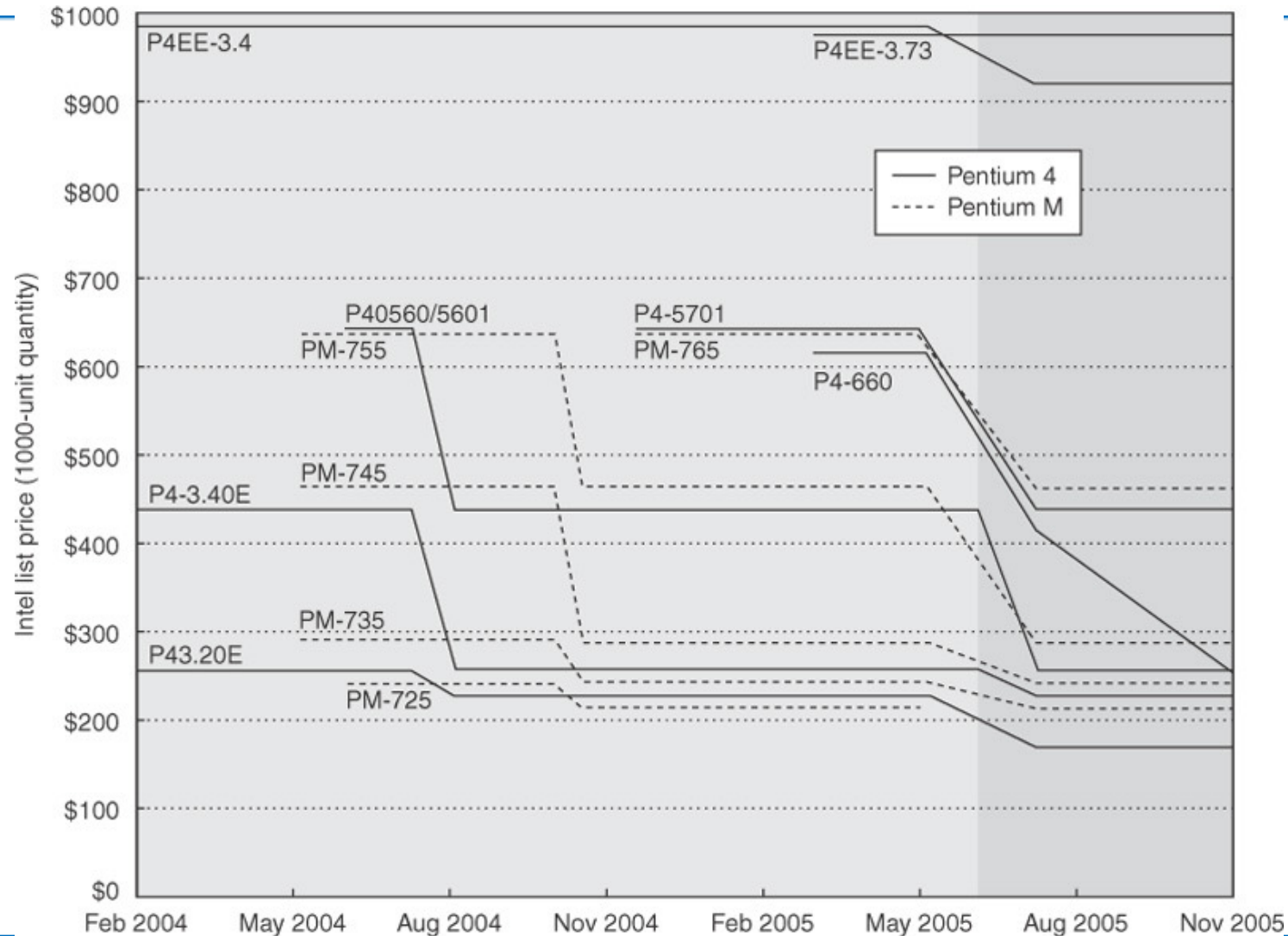


Creștere exponențială

Evoluția Performanței

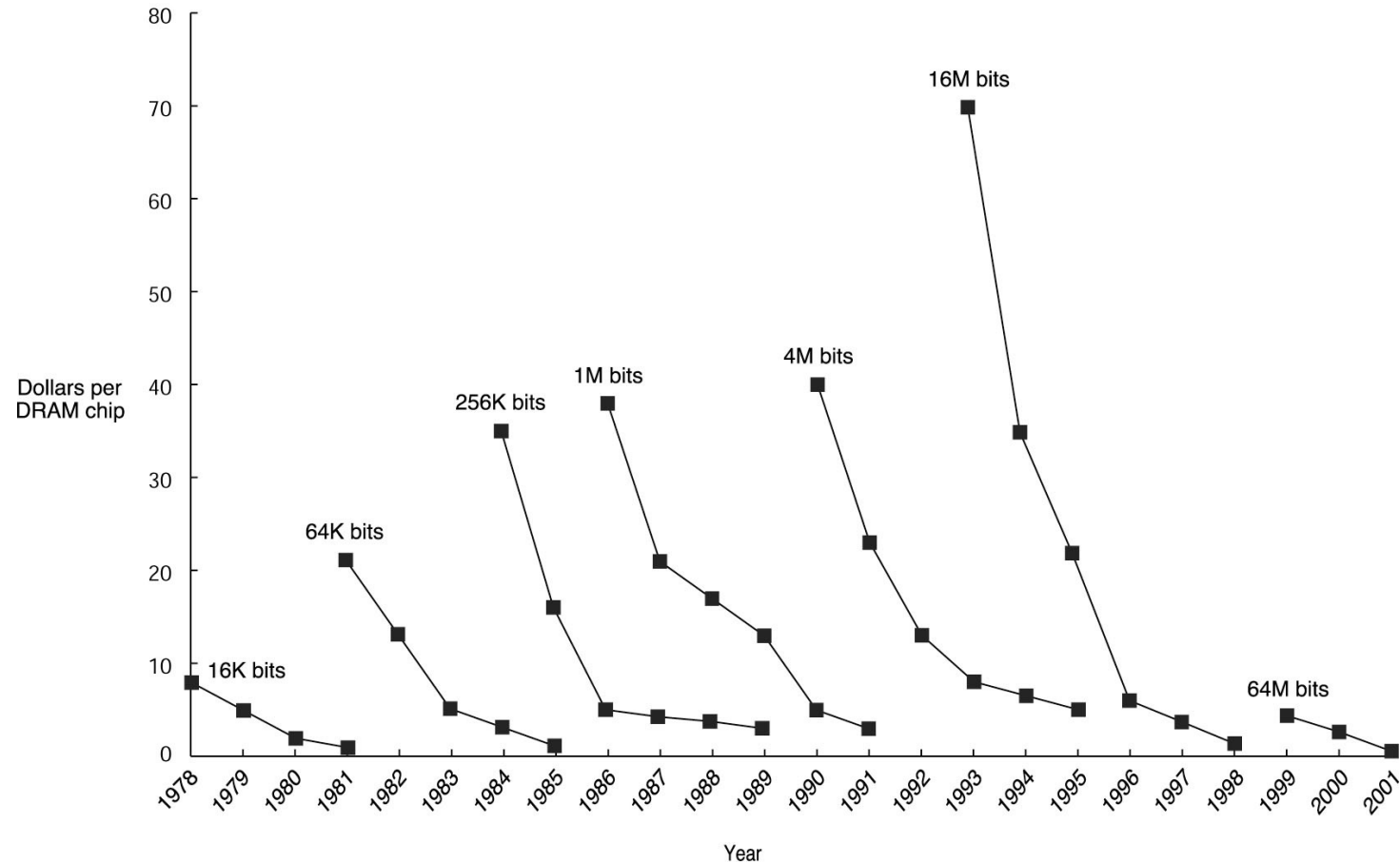


Evoluția Prețurilor (Pentium)



© 2007 Elsevier, Inc. All rights reserved.

Evoluția Prețurilor (memoria DRAM)



© 2003 Elsevier Science (USA). All rights reserved.

De ce ne pasă prețul?

- Publicul țintă și prețurile de pe piață influențează costul procesorului meu
 - preț = cu cât vând componenta
 - cost = cât mă costă s-o produc
- Deciziile luate în timpul proiectării afectează costul (și prețul)
 - Ex. Adaug mai mult cache -> îmbunătățește performanța dar crește costul
- Raportul *preț-performanță*

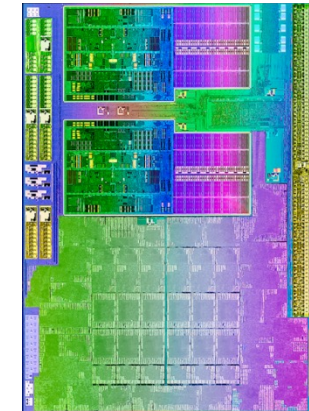
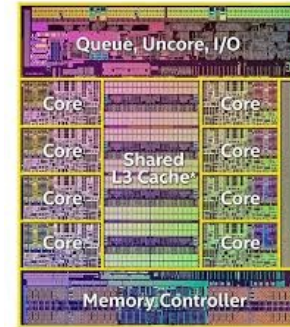
Deci,
ce influențează costurile?

Clasa din care face parte procesorul

Feature	Desktop	Server	Embedded
Preț sistem(USD)	\$500-\$5K	\$5K - \$5M	\$10 - \$100K (ex. high-end network routers)
Preț CPU (per processor)	\$50 - \$500	\$200 - \$10K	\$0.01 - \$100
Probleme de design	Price-performance, graphics performance	Throughput, availability, scalability	Price, power, application-specific performance

Sisteme Desktop/Laptop

- Exemple
 - Intel Core i7
 - AMD A4
- Aplicații: orice (general purpose)
 - Office, Internet, Multi-media, Video Games...
- Țeluri
 - Performanță, preț/performanță
 - Putere mărită → afectează costul, zgomotul, mărimea

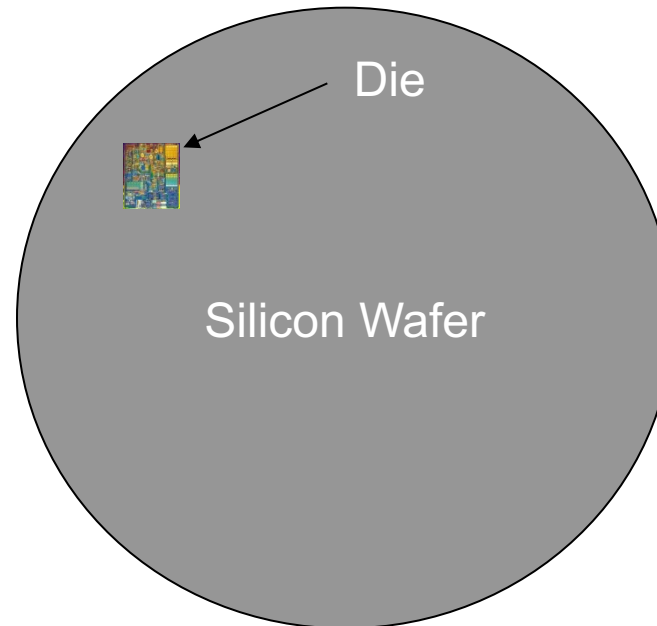


- Exemple
 - Intel Xeon
 - IBM Power
 - AMD Opteron
- Aplicații
 - infrastructură: file server, email server, ...
 - business: web, e-commerce, databases
- Țeluri
 - Productivitate (tranzacții/secundă)
 - Disponibilitate (fiabilitate, toleranță la defecte ...)
 - Costul nu este un aspect important

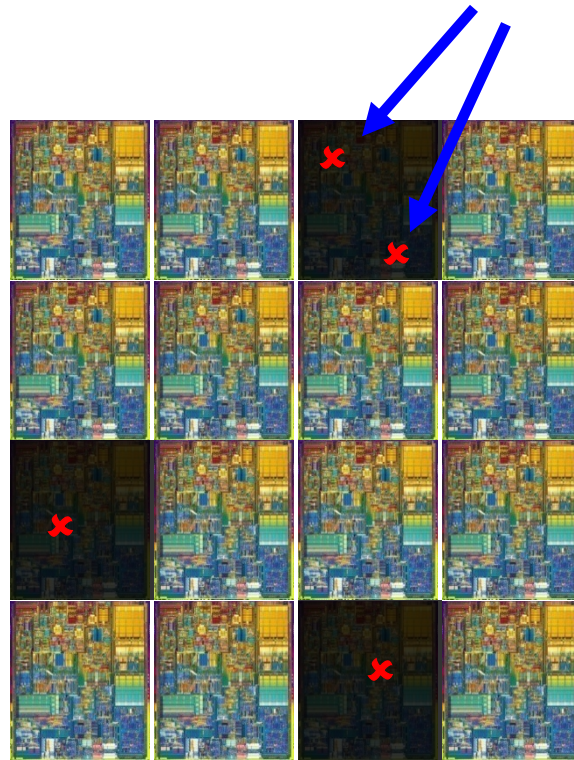
- Exemple
 - **ARM**, RISC-V, MIPS, PIC, AVR, x86, ... (multe “sortimente”)
- Aplicații
 - telefoane, tablete, console de jocuri, electrocasnice (frigider, cuptor microunde), automobile, ... (și mai multe sortimente)
- Țeluri
 - Cost, Putere
 - Performanțe *suficiente*, real-time
 - Dimensiune (CPU, memory footprint, cip count...)

Costuri de fabricație

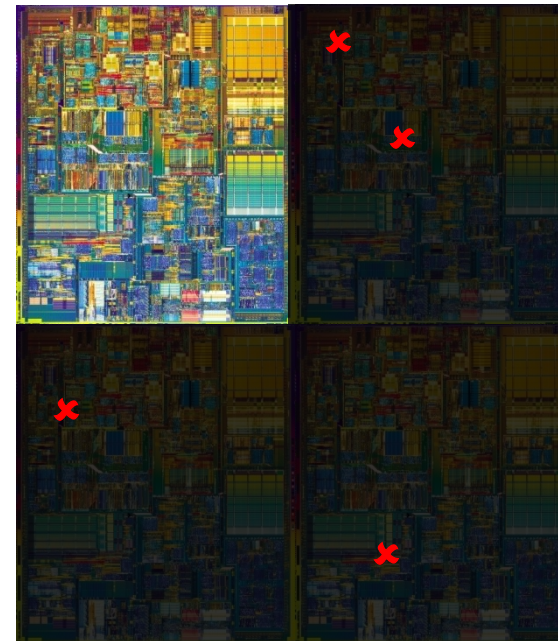
- Mărimea matriței CPU (die) influențează foarte mult costul tuturor sistemelor (desktop/server/embedded)
 - CPU actual 1-2 cm²
 - Embedded – mult mai mici
 - Costul și amprenta contează foarte mult într-un telefon mobil sau wearable



Yield

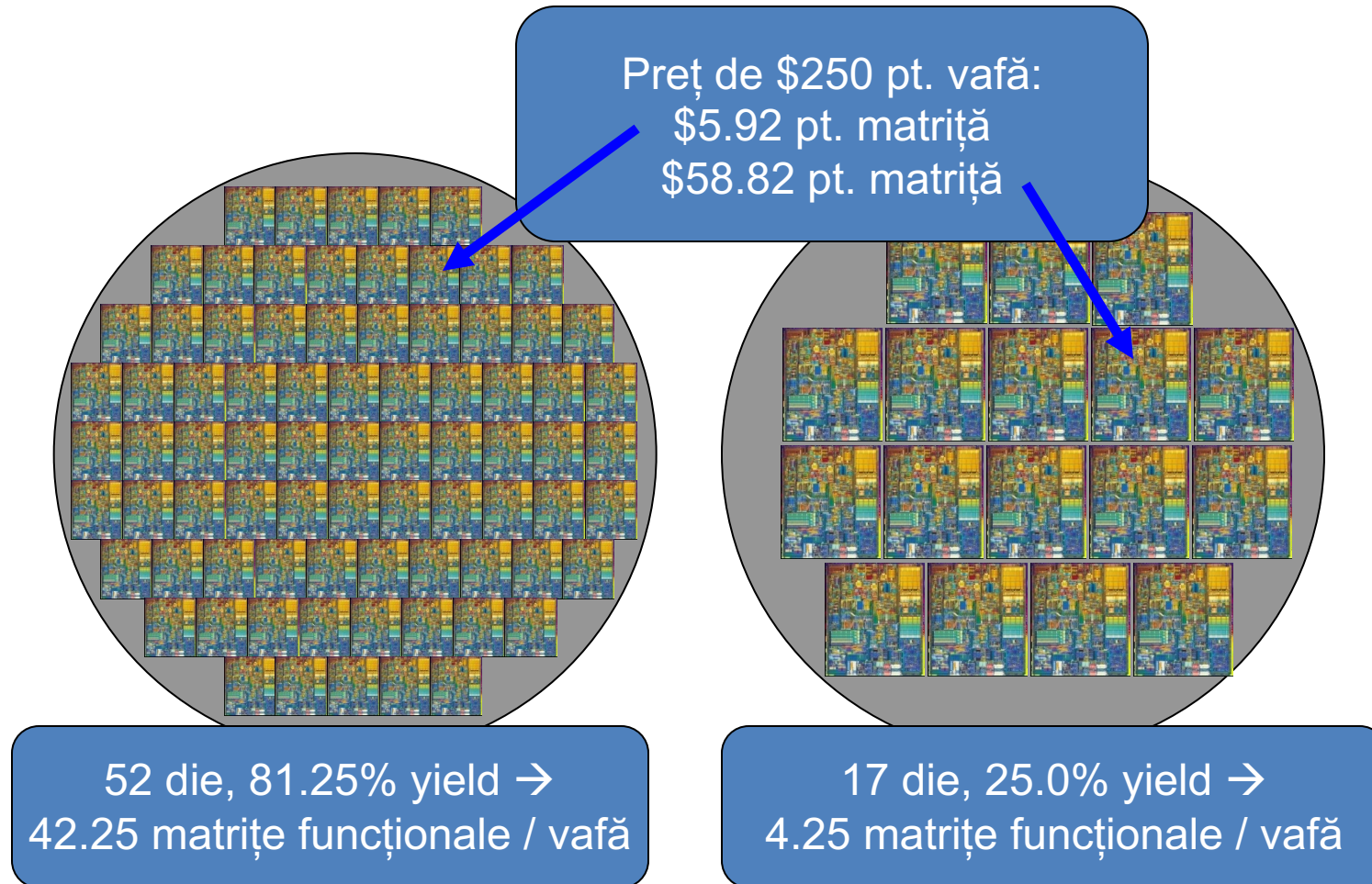


13/16 cipuri funcționale
81.25% yield



1/4 cipuri funcționale
25.0% yield

Yield (2)



Ecuatii Cost/Yield (aproximații)

$$\text{Cost of Die} = \frac{\text{Cost of wafer}}{\text{Dies per wafer} \times \text{Die yield}}$$

$$\text{Dies per wafer} = \frac{\pi \times (\text{Wafer diameter} / 2)^2}{\text{Die area}} \quad \text{---} \quad \frac{\pi \times \text{Wafer diameter}}{\text{sqrt}(2 \times \text{Die area})}$$

$$\text{Die yield} = \text{Wafer yield} \times \left(1 + \frac{\text{Defects per unit area} \times \text{Die area}}{\alpha} \right)^{-\alpha}$$

Numărul de
vafe defecte

Tipic: 0.4 defecte pe
cm² in 90nm, dar
scade cu timpul

Parametru legat de
complexitatea producerii, de
obicei $\alpha = 0.4$

Interacțiunea dintre preț și performanță

- Adaug un nou subsistem pe cip
 - Pentru performanță sporită, power management, etc.
- Suprafața cipului crește
 - Mai puține matrițe pe vafă
 - Mai multe vafe cu defecte
- Testarea matrițelor e mai costisitoare
 - Trebuie testat dacă noul sistem funcționează în orice condiție
- Capsula în care este pusă matrița e mai scumpă
 - Mai mare, poate mai mulți pini
 - Dacă consumă mai multă energie -> radiator mai bun etc.

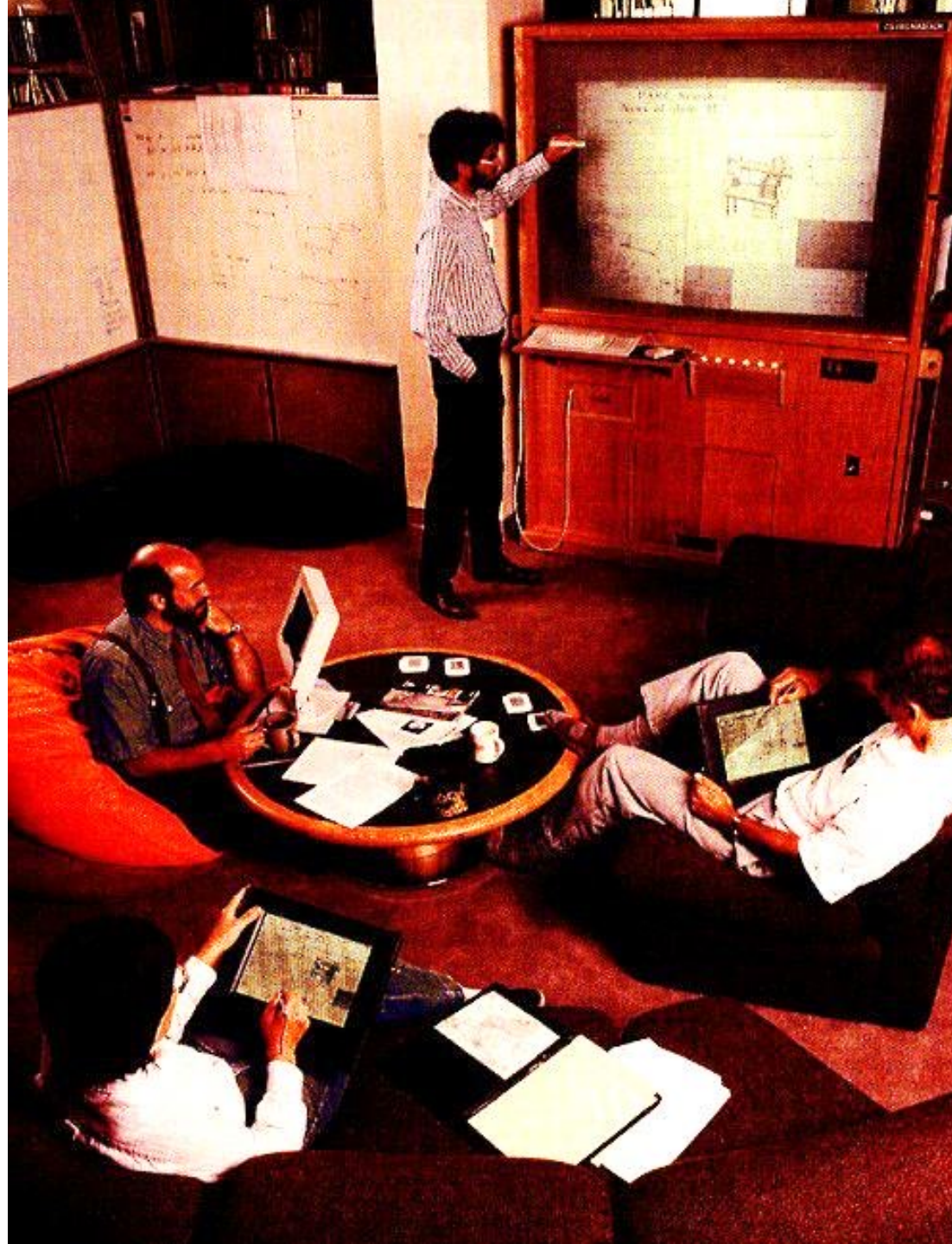
Țeluri în designul unui procesor

- Maximizarea performanței
- În limitele următoarelor constrângeri
 - Puterea maximă, medie, disiparea căldurii, fiabilitate, costuri de producție, complexitatea circuitelor, complexitatea verificării, time-to-market, cost to manufacturer (Intel), cost to OEM (Dell), cost to end-customer (voi)
 - Ce înseamnă defapt:
 - **Maximizează performanța per \$\$\$**
- Problemă greu de optimizat, cu mai multe variabile!
 - Nu toate variabilele sunt independente

Laptop/Embedded:
Constrângerile de putere,
greutate, dimensiune sunt
mai puternice decât la un
Desktop/Server

Tendințe Recente

- Cerințe crescute de calcul
 - Procesare multimedia, HDTV
- Conectare în rețea
 - Pentru monitorizare și depanare de la distanță.
 - Server Web embedded
 - e.g. Axis camera <http://neteye.nesl.ucla.edu>
 - e.g. Mercedes car with web server
 - Mașini Java embedded
 - e.g. Java ring, smart card, imprimante
 - Camere foto/video care se conectează direct la rețea
- Nevoie crescută de flexibilitate
 - Timpul necesar dezvoltării unui produs nou scade tot mai mult





Ubiquitous computing

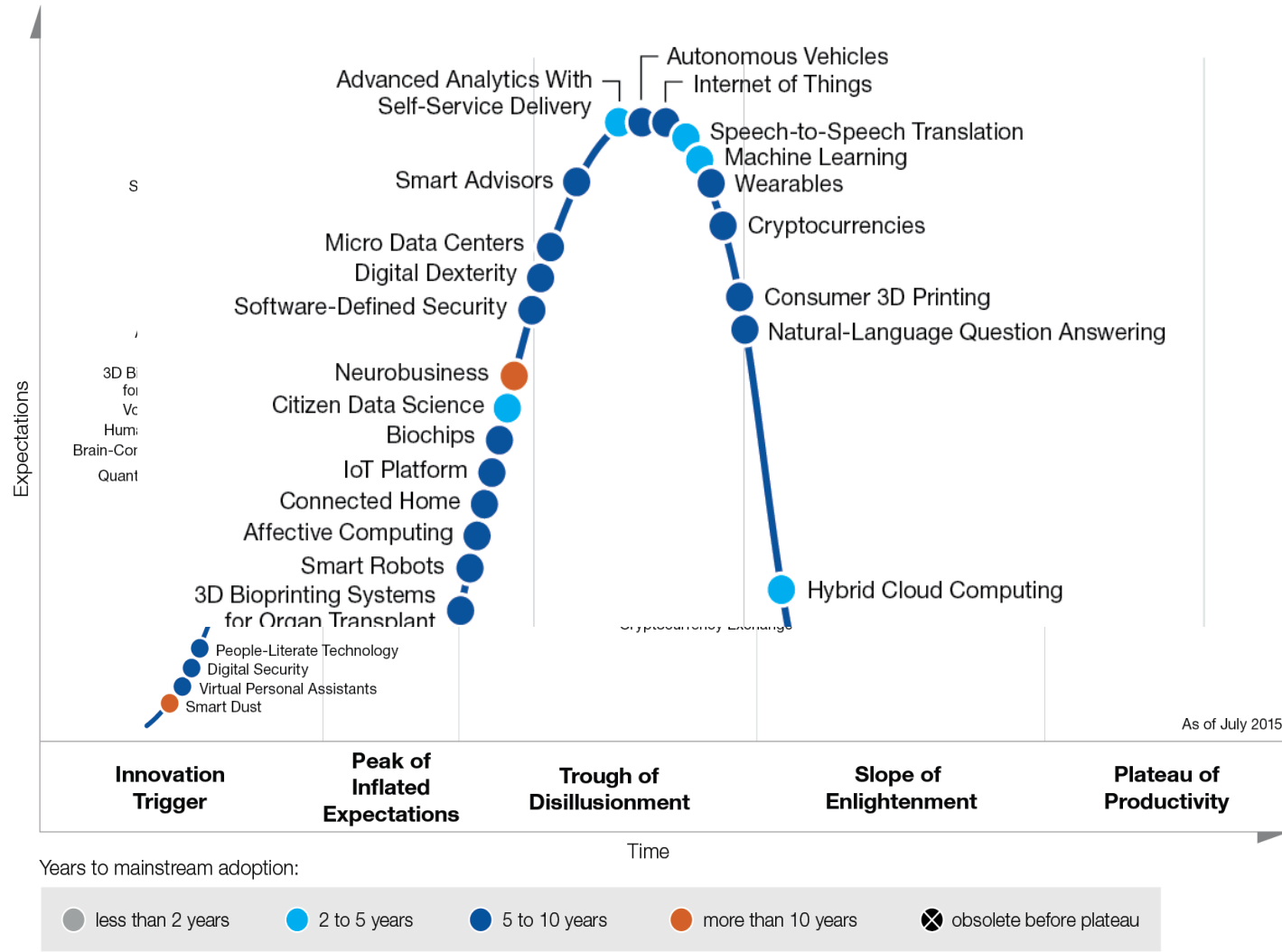
Pervasive computing, ambient intelligence sau everywhere

- Model de interacțiune om-masină în care procesarea informațiilor este adânc integrată în obiectele și activitățile zilnice.
- "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." (Mark Weiser – The Computer for the 21st Century)

Principii :

- The purpose of a computer is to help you do something else.
- The best computer is a quiet, invisible servant.
- The more you can do by intuition the smarter you are; the computer should extend your *unconscious*.
- Technology should create calm.

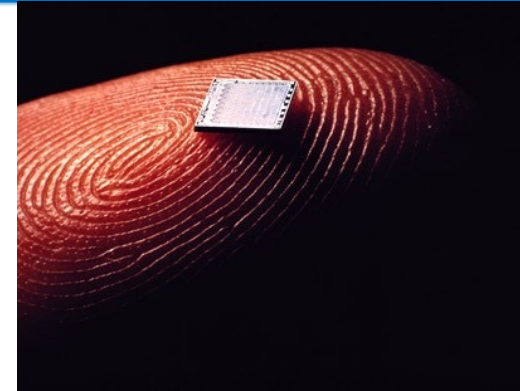
Emerging Technology Hype Cycle



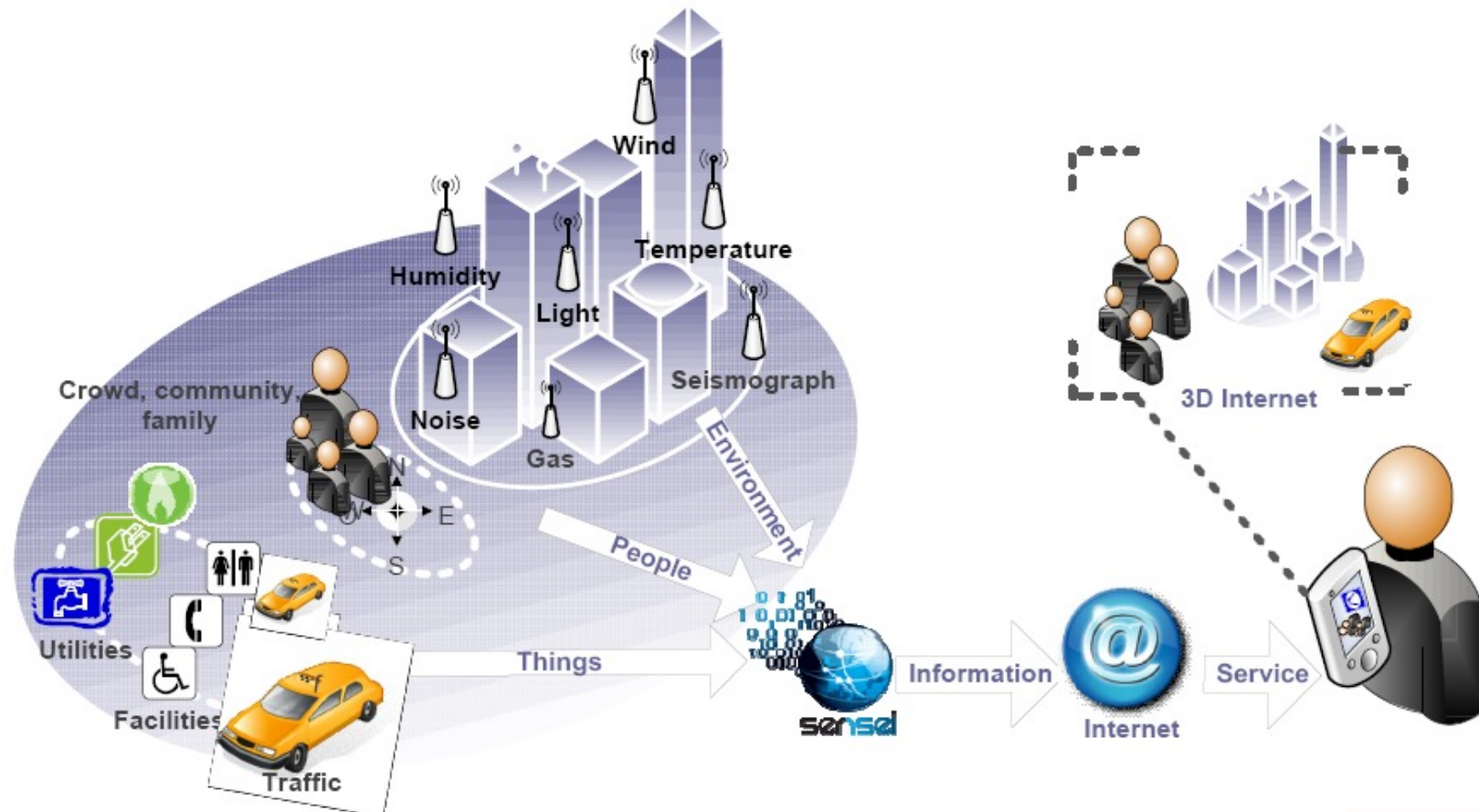
Ubiquitous computing

Clasificare device-uri:

- ▶ Tabs - wearable centimetre sized devices
 - ▶ Pads - hand-held decimetre-sized devices
 - ▶ Boards - metre sized interactive display devices
-
- ▶ Dust – dispozitive miniaturizate
 - Micro Electro-Mechanical Systems (MEMS)
 - Dimensiuni de la nanometri la milimetri
 - Smart dust
 - ▶ Skin
 - Wearable computing
 - "Painted" MEMS and circuitry
 - ▶ Clay – claytronics, programmable matter
 - Agregate MEMS
 - Formează structuri 3D arbitrare



Exemplu: Smart City



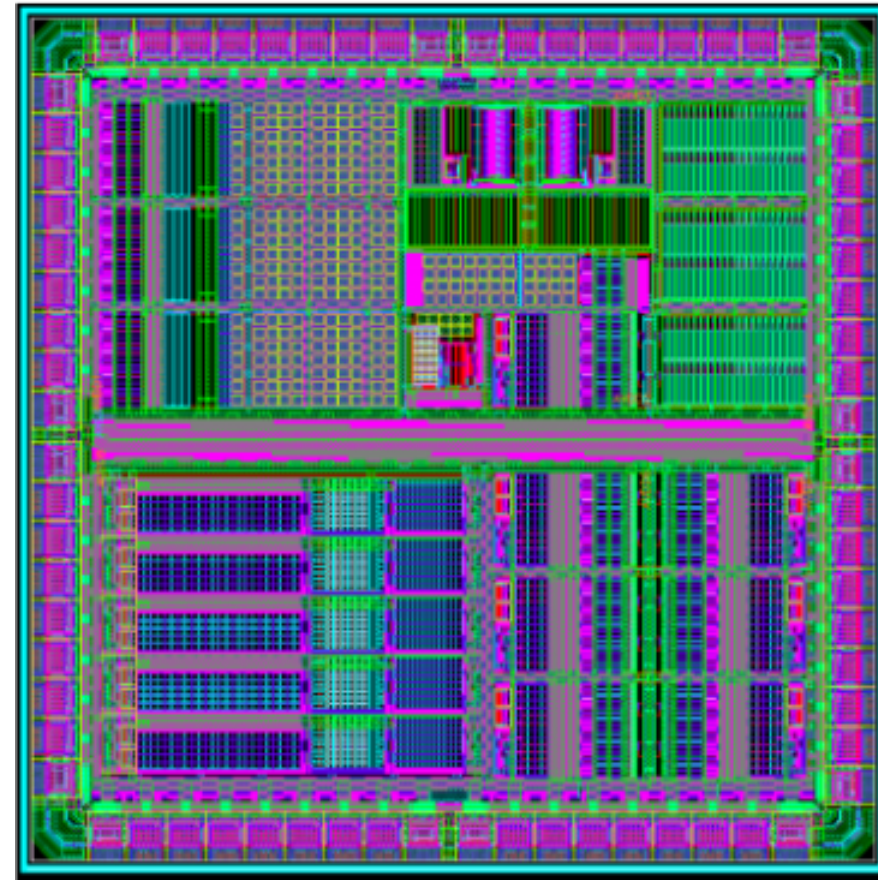
www.sensei-project.eu

Application-Specific Integrated Circuit

Circuit non-standard proiectat
pentru o anumită aplicație

Clasificate în trei categorii

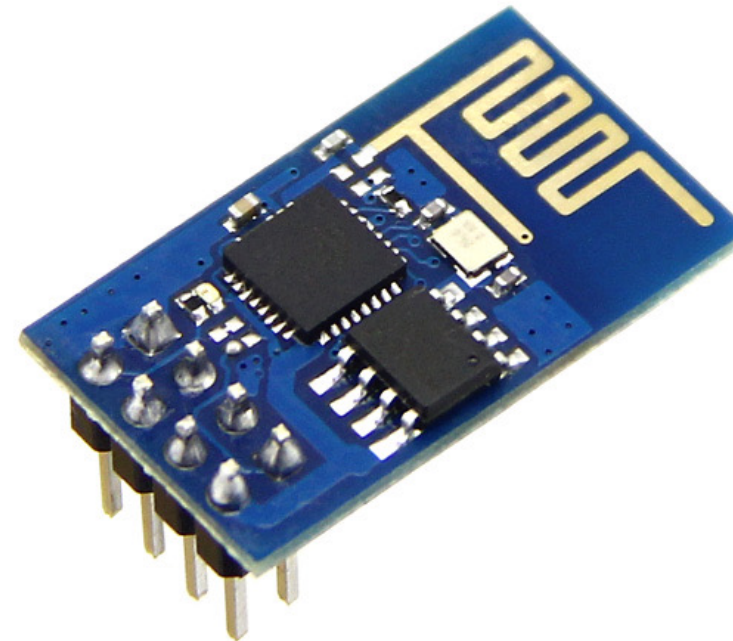
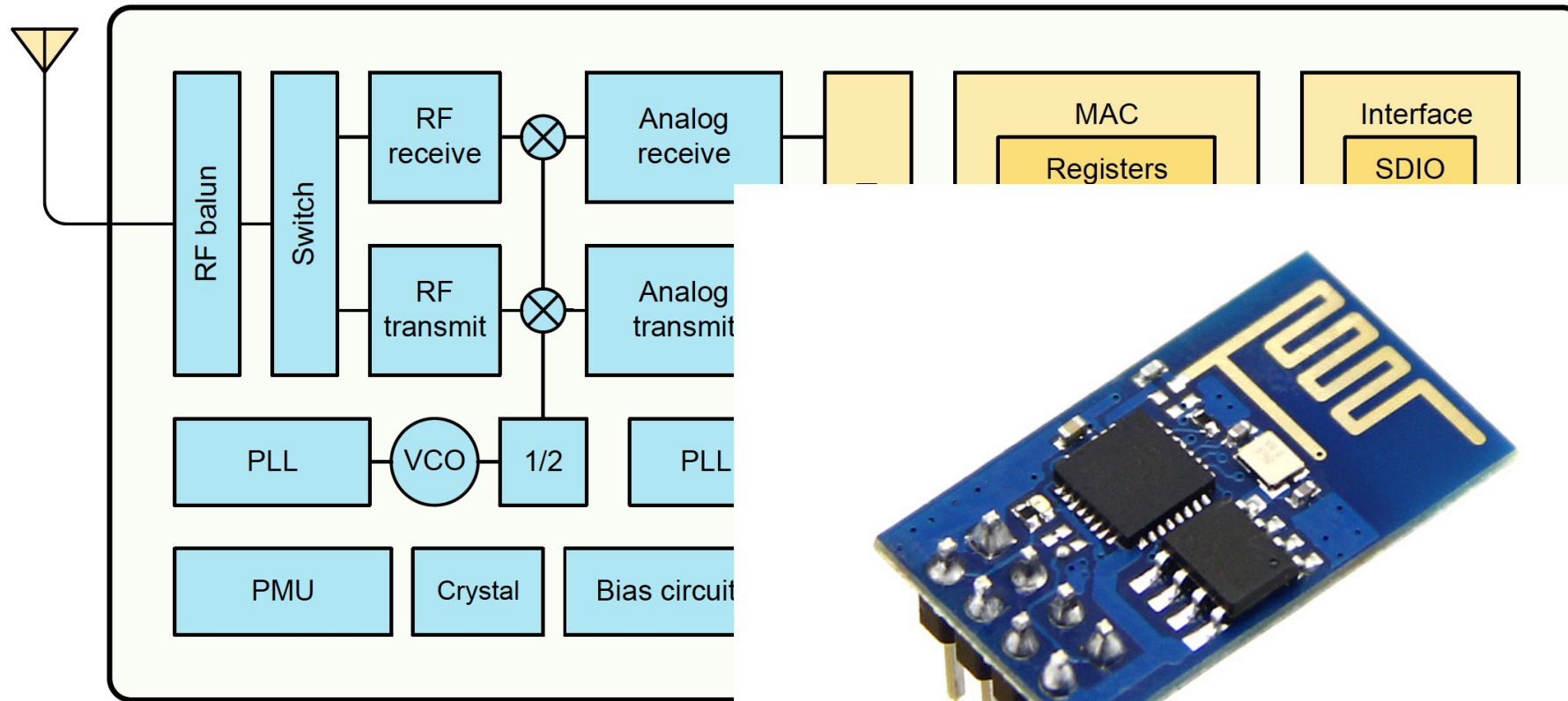
- full-custom
- semi-custom
- programmable



System-on-Chip (SoC)

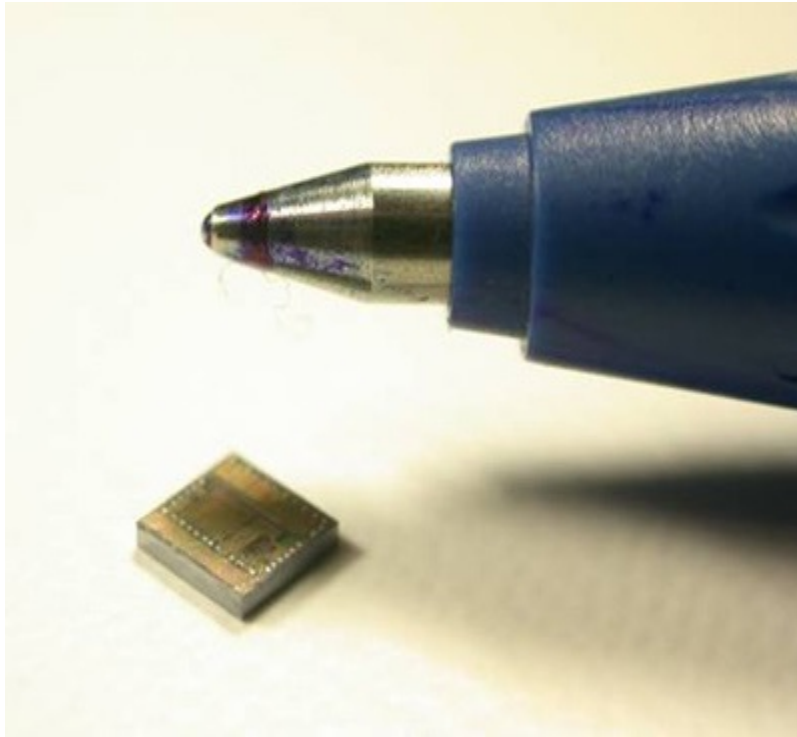
- Definiție: un cip care conține toate circuitele electronice necesare pentru un sistem complet.
 - SoC au în compoziție memorie(RAM și ROM), microprocesorul, interfețe periferice, interfețe I/O, convertoare și alte componente necesare sistemului în cauză.
- SoC este în concordanță cu legea lui Moore.

Exemplu de System-on-chip (SoC)



ESP8266 – cheapest WiFi SoC

Mai multe exemple

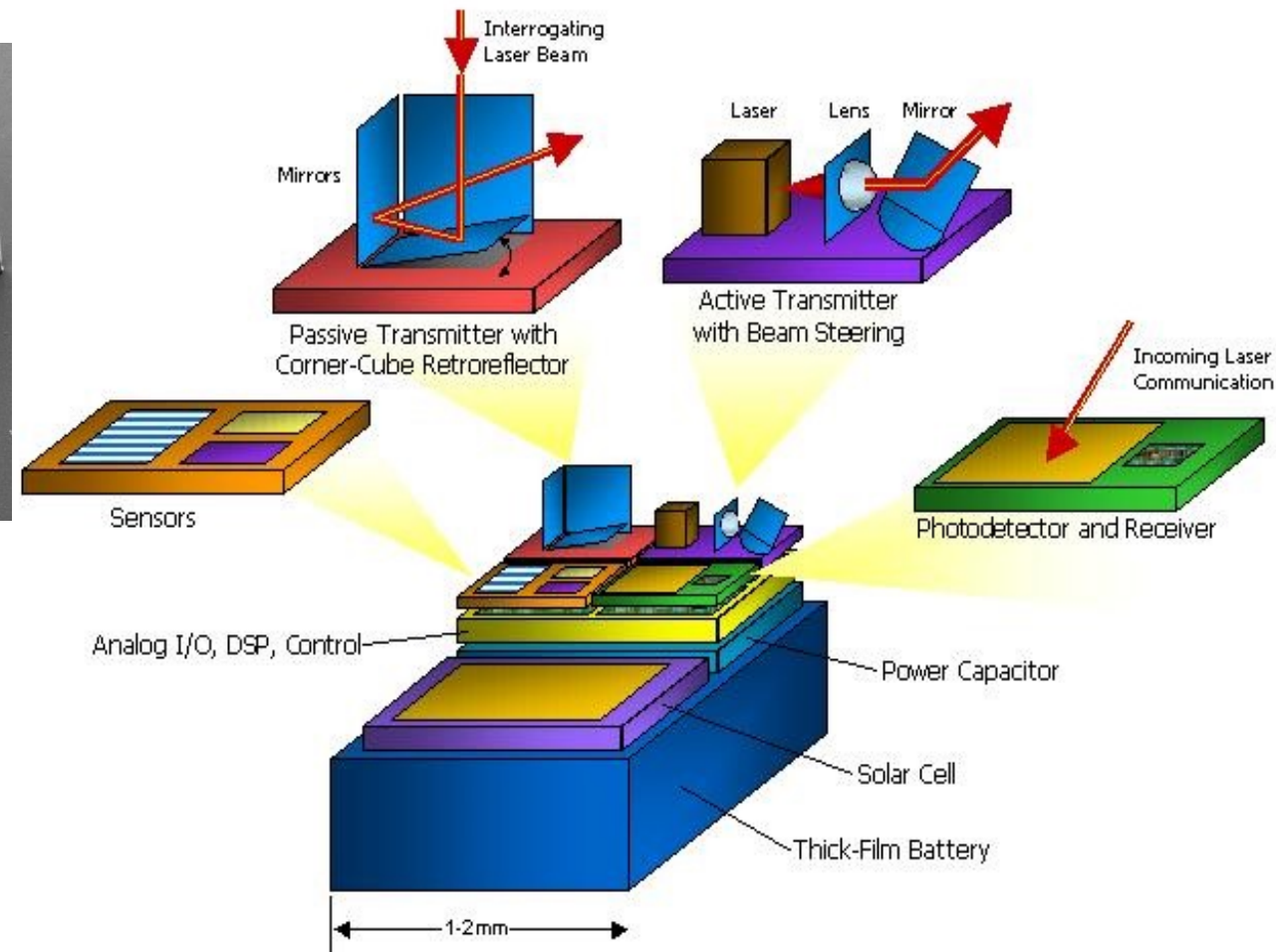
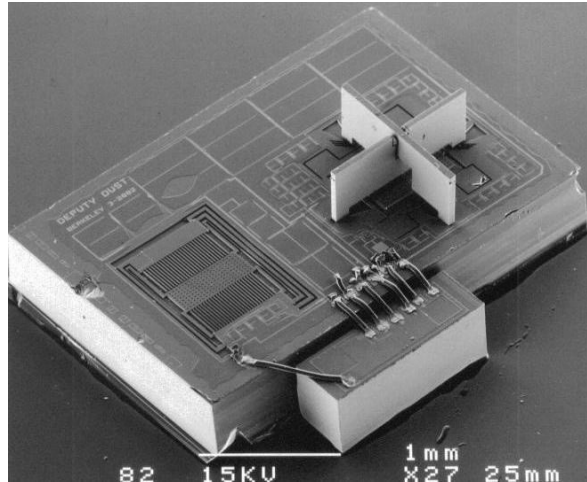


Spec Mote (Berkeley)



Solar-power Wireless Sensor (Berkeley)

SoC + mecanică: Berkeley Smart Dust



- Unele porțiuni din cip sunt predefinite pentru un anumit domeniu de aplicație
 - Procesor, blocuri periferice proprietate intelectuală (IP), memorie sau magistrală de un anumit tip, sistem de operare de timp real etc.
- Particularizarea designului prin adăugarea de module hardware IP sau software încorporat
- De ce există?
 - Creșterea diferenței de productivitate
 - Prețul ridicat de fabricație al ASIC

“+” Ciclu de design rapid din cauza folosirii unor subsisteme deja existente

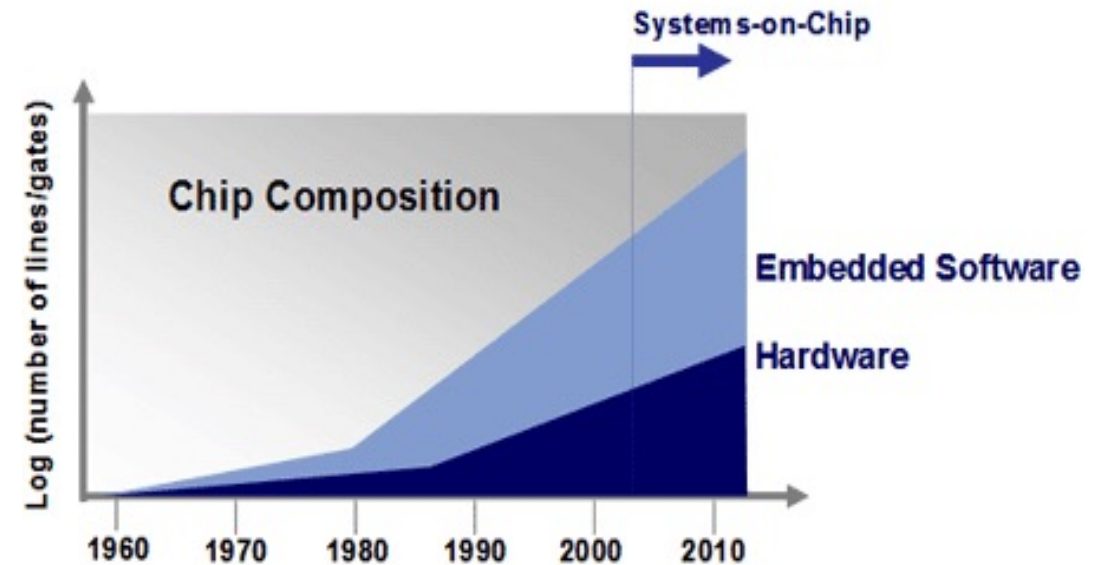
“- “ Flexibilitate redusă

Migrarea de la hardware la software embedded

Piața pentru software embedded crește cu 16% pe an.

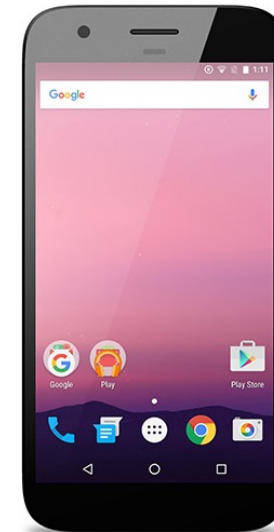
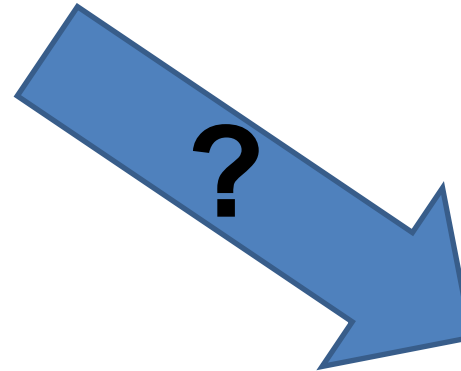
- 1.6 miliarde \$ în 2004
- 3.5 miliarde \$ în 2009
- >20 miliarde \$ în 2025

Introducerea tehnologiilor multimedia care cer o lățime foarte mare de bandă (ex. HDTV, 4k, 5G) necesită încorporarea unui volum din ce în ce mai mare de software în dispozitivele de larg consum.



Sursa: ChipDesign Magazine

Embedded Design – HowTo?



- ✓ 4p – examen scris
- ✓ 3p – laborator
 - 2p – activitate
 - 1p – test final
- ✓ 1p – activitate la curs
- ✓ 2p – teme de casă

Condiții de promovare:

1. Minim 2p în examen
ȘI
2. Minim 1.5p în laborator
ȘI
3. Minim 1.5p activitate + teme
ȘI
4. Minim 6 prezențe la laborator

Copiatul Temelor / Examenelor

- Majoritatea studentilor nu îmbrățișează aceste metode
- Pentru toți ceilalți se aplică următoarele reguli:
 - Copiatul unei teme de casă anulează punctajul temei
 - Copiatul lucrării de laborator = 0p la laboratorul respectiv
 - “Infracțiuni” repetate -> anularea punctajului pentru toate temele / laboratoarele
 - Copiat la examen -> se repetă materia anul următor (în cel mai bun caz)

- Paper-uri de citit la fiecare curs
- Internet (Google & Wikipedia are your best friends)
- Michael Barr - **Programming Embedded Systems - With C and GNU Development Tools** 2e (O'Reilly, 2006)
- Jorg Henkel - **Designing Embedded Processors - A Low-Power Perspective** (Springer, 2007)
- Stuart Ball - **Embedded Microprocessor Systems - Real World Design** 3e (2002)
- **Embedded Linux Primer - A Practical Real-World Approach** (2006)
- **O'Reilly - Building Embedded Linux Systems**

- Stanford
 - Embedded Systems Design
 - <http://www.stanford.edu/class/ee281/course.html>
- U. California
 - Embedded System Design
 - <http://esd.cs.ucr.edu/index.html>
- Berkeley
 - EE 249: Design of Embedded Systems: Models, Validation, and Synthesis
 - <http://www-cad.eecs.berkeley.edu/~polis/class/index.html>
- U.T. Austin
 - EE382C-9 Embedded Software Systems
 - <http://www.ece.utexas.edu/~bevans/courses/ee382c/index.html>
- Berkeley
 - EE290N: Specification and Modeling of Reactive Real-Time Systems
 - <http://ptolemy.eecs.berkeley.edu/~eal/ee290n/index.html>
- UCI
 - ICS 212: Introduction to Embedded Computer Systems
 - <http://www.ics.uci.edu/~rgupta/ics212.html>
 - ICS 213: Software for Embedded Systems
 - <http://www.ics.uci.edu/~rgupta/ics213.html>