

# **TEORIE CARTE**

## **Eight Great Ideas in Computer Architecture**

### **1. Design for Moore's Law**

**Resursele circuitelor integrate se dublează la fiecare 18-24 de luni. Deoarece proiectarea calculatoarelor poate dura ani de zile, resursele disponibile pentru fiecare chip se pot dubla sau cvadrupla cu ușurință între începutul și sfârșitul anului. De aceea este esențial ca arhitecturile calculatoarelor să anticipe această avansare tehnologică rapidă.**

### **2. Use Abstraction to Simplify Design**

**O tehnică majoră de productivitate pentru hardware și software este abstractizarea pentru a reprezenta proiectarea la diferite niveluri. Complexitatea din hardware este ascunsă pentru a obține un model mai simplu la nivelurile superioare.**

### **3. Make the Common Case Fast**

**Accelerarea cazului comun va tinde să îmbunătățească performanța mai bine decât optimizarea cazurilor rare.**

### **4. Performance via Parallelism**

**Efectuarea operațiilor în paralel produce mai multă putere computațională.**

### **5. Performance via Pipelining**

**Este un tip de paralelism la nivel de instrucțiune**

## **6. Performance via Prediction**

**În unele cazuri, poate fi mai rapid, în medie, să anticipi ceea ce urmează și să începi să execuți decât să aștepti până când știi sigur ce urmează să fie executat.**

**Presupunând că mecanismul de recuperare a unei previziuni greșite nu este prea costisitor, iar predicția este relativ precisă, predicția este o îmbunătățire masivă adusă asupra sistemului de calcul.**

## **7. Hierarchy of Memories**

**Pentru orice sistem de calcul ideal este ca memoria să fie rapidă, ieftină și cu o capacitate mare.**

**Aceste cerințe nu pot fi toate îndeplinite simultan. Viteza memoriei limitează performanța și capacitatea limitează dimensiunea problemelor care pot fi rezolvate.**

**Aceste cerințe contradictorii pot fi reprezentate cu ajutorul unei ierarhii de memorie.**

**Forma indică viteza, costul și dimensiunea:**

- **cu cât este mai aproape de vârf, memoria va fi mai rapidă, mai scumpă și cu o capacitate mai mică**
- **cu cât este mai aproape de baza, cu atât memoria este mai lentă, mai ieftină și cu o capacitate mai mare**

## **8. Dependability via Redundancy**

**Calculatoarele nu trebuie să fie doar rapide, ci și fiabile.**

**Deoarece orice dispozitiv fizic se poate defecta, noi facem sistemele fiabile prin includerea de componente redundante care pot prelua controlul atunci când apare o defecțiune și care pot ajuta la detectarea defecțiunilor.**