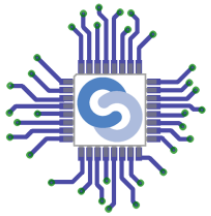


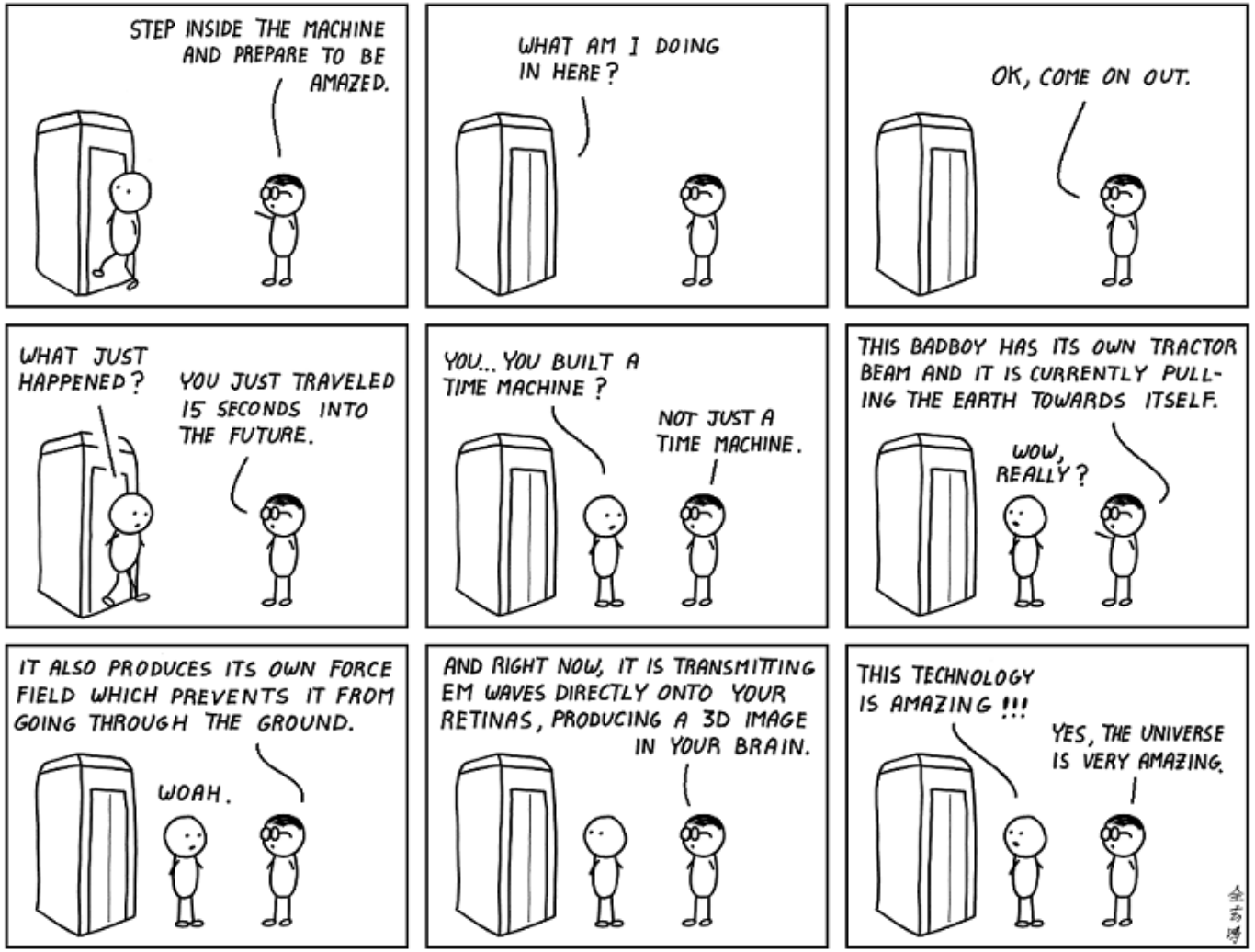
Calculatoare Numerice (2)

– Cursul 8 –

Magistrale si arhitecturi de calcul

Facultatea de Automatică și Calculatoare
Universitatea Politehnica București

Comic of the day

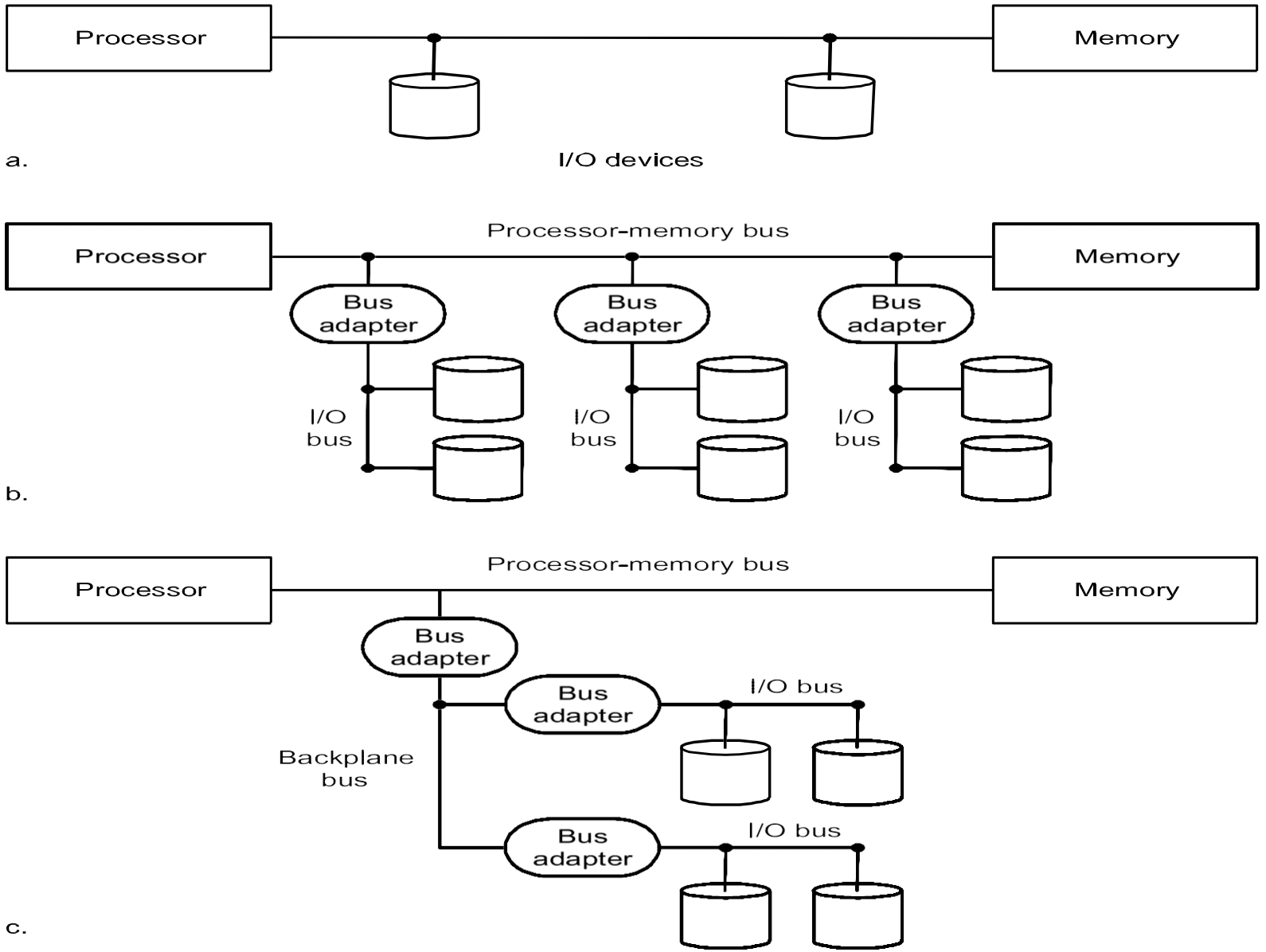


Din episodul anterior

1. **Processor-memory (local) bus:**
   - De obicei scurt și de viteză mare pentru a maximiza lățimea de bandă
2. **I/O Bus**
   - Lung, trebuie să suporte viteze de date diferite de la dispozitive diferite
   - Trebuie să facă față unei game largi de latențe, lățimi de bandă și specificații
3. **Backplane Bus**
   - Este numit așa pentru că era la început pe spatele șasiului
   - Permite interconectarea memoriei, procesorului și a dispozitivelor I/O
   - Necesită logică suplimentară pentru a se lega la alte magistrale

- Magistralele locale sunt de obicei design-specific în timp ce cele de I/O și backplane sunt portabile și de obicei respectă un standard industrial
- Un sistem poate să folosească un backplane sau o combinație de toate trei

Configurații de bus



Exemplu de calcul performanță

Un bus sincron are o perioadă de ceas de 50ns și fiecare transmisie durează un ciclu de ceas. Alt bus asincron are nevoie de 40ns pentru fiecare handshake. Fiecare bus are 32 de biți de date. Care este lățimea de bandă a fiecărui bus pentru citiri de un cuvânt dintr-o memorie cu 200ns per read.

Răspuns:

Pașii pentru magistrala sincronă:

1. Trimite adresa la memorie: 50 ns
2. Citește memoria: 200 ns
3. Trimite datele la dispozitiv: 50 ns

} 300 ns

Lățimea maximă de bandă este 4 octeți la 300 ns $\Rightarrow \frac{4 \text{ bytes}}{300 \text{ ns}} = \frac{4 \text{ M B}}{0.3 \text{ sec}} = 13.3 \text{ MB/sec}$

Exemplu de calcul performanță

Secvența de pași pentru magistrala asincronă:

1. Memory read address atunci când apare ReadReq : 40 ns
2,3,4. Data ready & handshake: max(3x40 ns, 200 ns)= 200 ns
5,6,7. Read & Ack. : 3 × 40 ns = 120 ns

} 360 ns

Lățimea maximă de bandă este 4 octeți la 360 ns $\Rightarrow \frac{4 \text{ bytes}}{360 \text{ ns}} = \frac{4 \text{ MB}}{0.36 \text{ sec}} = 11.1 \text{ MB/sec}$

Creșterea lățimii de bandă

- Lățimea de bandă este de obicei determinată de protocol și de caracteristicile de temporizare a semnalelor
- Alți factori:
  - Lățimea magistralei de date:
    - Transferă cuvinte multiple, necesită mai puțini cicli de bus
    - Creșterea numărului de linii de date (scump!)
  - Multiplexarea liniilor de adrese și date:
    - Folosirea liniilor separate de date și adrese accelerează tranzacțiile
      - Simplifică logica de control pe bus
      - Mărește numărul de linii de date
  - Transfer pe blocuri:
    - Transferă cuvinte multiple de la adrese consecutive în rafală
    - Mărește timpul de răspuns deoarece tranzacțiile vor fi mai lungi
    - Mărește complexitatea logicii de control a magistralei

Bus Master

- Single master
  - Bus master (ex. procesorul) controlează toate accesele pe bus
  - Bus slave (ex. device sau memorie) doar răspunde la cereri
- Multiple master
  - Mai multe dispozitive pot iniția o tranzacție
  - Bus control logic și protocolul adoptat rezolvă conflictele

Arbitrare

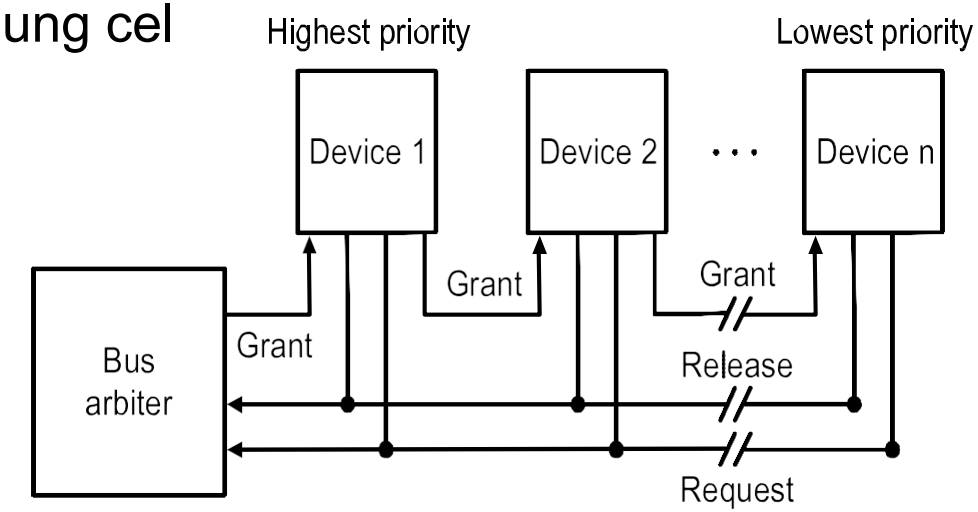
- Arbitrarea pe bus coordonează utilizarea magistralei folosind mecanisme de request, grant, release
- Arbitrarea încearcă de obicei să echilibreze doi factori atunci când alege dispozitivul care preia magistrala:
  - Dispozitivele cu prioritate mai mare trebuie servite primele
  - Menține corectitudinea; nici un dispozitiv nu va fi reprimat total
- Timpul de arbitrare este un overhead
  - Vrem să îl minimizăm

Tehnici de arbitrare

- *Distributed arbitration by self-selection:* (ex. NuBus folosit la Apple Macintosh)
  - Folosește linii multiple de request pentru dispozitive
  - Dispozitivele care fac request determină cui i se va da acces
  - Dispozitivele asociază un cod cu cererea, indicând prioritatea
  - Dispozitivele cu prioritate mică cedează busul dacă observă o cerere de prioritate mai mare
- *Distributed arbitration by collision detection:* (ex. Ethernet)
  - Dispozitivele cer independent acces la bus și preiau accesul
  - Cereri simultate produc o coliziune
  - Un algoritm de selecție între entitățile care au generat coliziunea
    - Ethernet: back off and try again later

Tehnici de arbitrare (Cont.)

- Daisy chain arbitration: (e.g. VME bus)
  - Linie specială de Grant Access, partajată de toate dispozitivele în ordinea priorității
  - Dispozitivele de prioritate mai mare arbitrează cererile dispozitivelor cu prioritate mai mică
  - Simplu de implementat
  - Duce la starvation pentru dispoztitivele de prioritate mică
  - Limitează viteza transmisiilor (semnalele de grant ajung cel mai târziu la ultimul device din lanț)
  - Permite propagarea defectelor
    - (o defecțiune a unui dispozitiv duce la defectarea întregului lanț)



Rolul sistemului de operare

- Sistemul de operare are rol de interfață între I/O hardware și programe
- Caracteristici importante ale sistemelor cu I/O:
  - Sistemul I/O este partajat între mai multe programe
  - Sistemele I/O folosesc întreruperi pentru a comunica starea lor procesorului
    - Întreruperile trebuie să fie tratate de SO pentru că transferă execuția în modul supervizor
  - Controlul low-level al unui dispozitiv I/O este complex:
    - Evenimente concurente
    - Cerințele pentru controlul corect al dispozitivului pot să fie foarte complexe

Responsablitățile SO

- Protecție pentru resursele I/O partajate
  - Garantează că un proces nu poate să acceseze I/O-ul altor procese
- Abstractizare pentru accesarea dispozitivelor
  - Rutine care supervizează operarea low-level a dispozitivului (drivere).
  - Interrupt handling pentru I/O
  - Acces echitabil la resursele I/O
    - Toate programele trebuie să aibă acces egal la resursele I/O
  - Planifică accesele pentru a mări productivitatea sistemului

Comunicația cu dispozitivele I/O

- SO trebuie să știe când:
  - Dispozitivul I/O a terminat o operație
  - Dispozitivul I/O a întâlnit o eroare
- Poate fi realizat în două moduri:
  - Polling:
    - Dispozitivul I/O pune informația de stare într-un registru
    - SO verifică periodic registrul de stare
  - I/O Interrupt:
    - Eveniment extern, asincron execuției principale, dar care NU interferează cu rularea codului principal
    - Ori de câte ori un dispozitiv I/O device cere atenție de la procesor, îl întrerupe
    - Unele procesoare tratează întreruperile ca pe niște excepții speciale

Slide: in part David Patterson

Acknowledgements

- These slides contain material developed and copyright by:
  - Arvind (MIT)
  - Krste Asanovic (MIT/UCB)
  - Joel Emer (Intel/MIT)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)
  - David Patterson (UCB)

- MIT material derived from course 6.823
- UCB material derived from course CS252