



Arhitectura Sistemelor de Calcul – Curs 3



Computer Science
& Engineering
Department

Universitatea Politehnica Bucuresti
Facultatea de Automatica si Calculatoare

cs.pub.ro

curs.cs.pub.ro



Cuprins

2

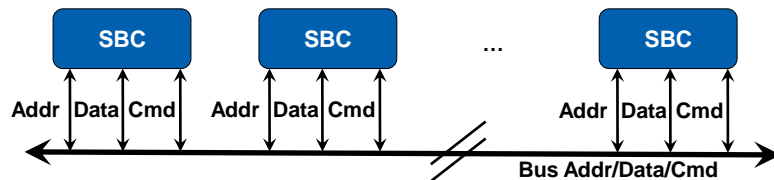
- Reprezentarea PMS a unei structuri multiprocesor organizata pe o magistrala comuna (SBC)
- K_{BUS}
- K_{ML}



Single Board Computer (SBC)

3

- Are procesor, memorie + interfete I/O
- Conecteaza pe o magistrala comenzi, date si adrese bidirectional
- Resurse SBC:
 - Private: procesor, EPROM, Interfete (S/P), RT Clock, Sistem de Intreruperi
 - Locale: Memoria locala
 - Globale: RAM (in spatiul de adresare)



SBC – Mecanisme Acces Magistrala

4

- Pentru a gestiona accesul la magistrala (exclusiv) sunt necesare:
 - K_{BUS} : unitate de comanda pt accesul la MAG
 - K_{ML} : unitate de control al accesului la memoria locala
 - K_{LD} : unitate de comanda a memoriei locale (refresh/etc)
 - K_s : unitate de comanda a intregului sistem – asigura controlul procesoarelor la resursele sistemului
- Fiecare procesor al SBC-ului vede memoria locala incapand de la 0 si memoriile celorlalte SBC-uri in spatii adiacente superioare



Elementele Structurii PMS a SBC

7

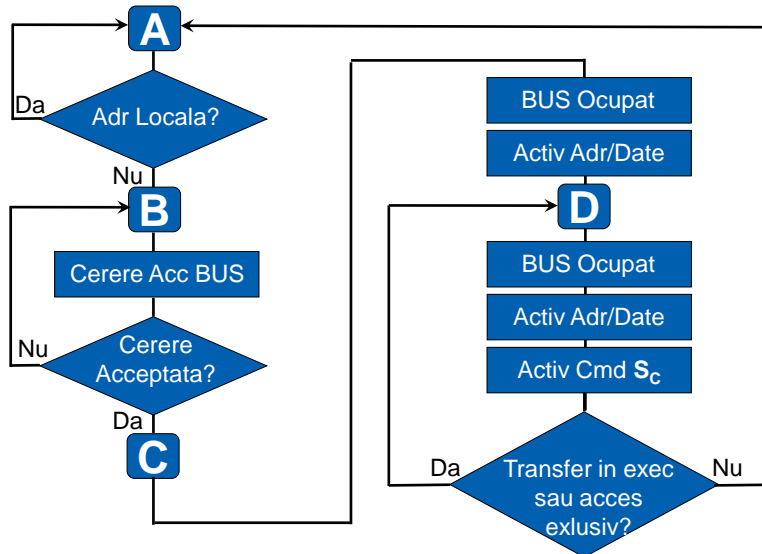
- P_L : procesor local – furnizeaza date si adrese
- Interfete:
 - T_T – de ceas de timp real
 - T_I – sistemului de intreruperi
 - T_S – seriala & T_P – paralela
 - Fiecare interfata are adresele portului sau de I/O
 - Daca isi recunoaste adresa isi pune datele pe magistrala
- M_{LD} – DRAM: memoria poate fi accesata de P_L si de procesoare din exterior
 - Exista niste S-uri care acorda accesul fie procesorului local fie celor externe
- S_{D1} si S_{D2} S-uri locale de date – separa procesoarele intre ele
- S_{A0} si S_{A1} – S-uri interne de adrese
- S_{A2} si S_{A3} – S-uri externe de adrese
- S_C – S prin care se transmit comenzi catre exterior
- Obs: structura PMS poate fi detaliata pe componente



Cuprins

8

- Reprezentarea PMS a unei structuri multiprocesor organizata pe o magistrala comuna (SBC)
- K_{BUS}
- K_{ML}

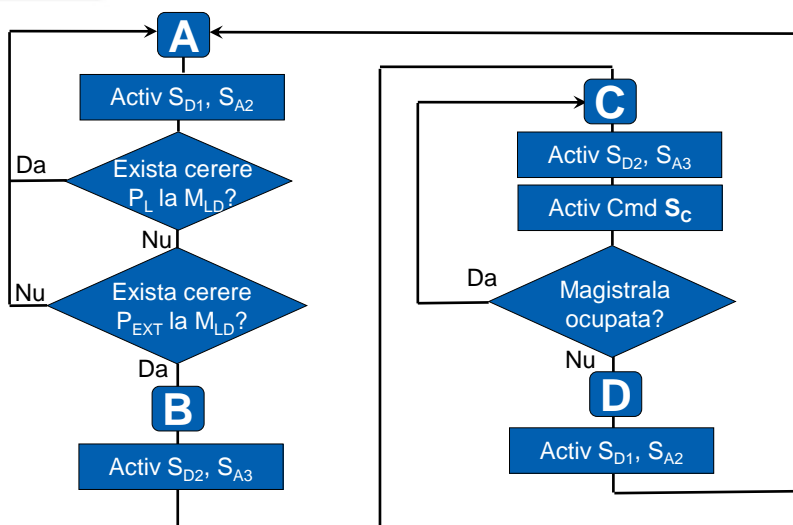


Descriere K_{BUS}

- K_{BUS} – reprezinta procesorul local in relatie cu magistrala externa
- Starea A: adresa e locala?
 - Da = se adreseaza resurse locale, nu trebuie iesit pe magistrala
 - Nu = adr nu e locala → se trece in starea B
- Starea B: cerere de acces la magistrala; daca cererea e acceptata se trece in starea C
- Starea C: se emite Bus Busy (mag ocupata) – doar in C & D; accesul pe BUS e exclusiv & cu rafala (overwrite) raman in D pt un nou ciclu daca mai e cazul; automatul preia functia procesorului; Dupa preluarea comenzii, procesorul face functia de secventiere a Adr, Date, Cmd
- Intrebare: de ce si starea C si starea D? Din cauza ciclului instructiune curent. Trebuie sincronizate comenzile cu decodificarea adreselor & stabilirea datelor pe magistrala (e un Δt acolo = T_{wait} Din cauza interpretarii adreselor) → se introduce starea D pentru a compensa acest timp de latenta: starea de overwrite
- Obs: K_{BUS} trebuie sa dispuna de unitati logice combinacionale ce identifica:
 - Daca adr mem/port I/O implicate in transfer sunt locale
 - Semnalul de cerere acceptata e combinat din mai multe conditii: mag libera & modul SBC activ



- Reprezentarea PMS a unei structuri multiprocesor organizata pe o magistrala comuna (SBC)
- K_{BUS}
- K_{ML}





Descriere K_{ML}

13

- K_{ML} – asigura accesul la Mem locala a P_L prin intermediul S_{D1} si S_{A2} & a proc externe prin S_{D2} si S_{A3}
- P_L are prioritate fata de proc externe (P_{EXT}) == daca vrea P_L la M_{LD} NU ma intereseaza ce vor celelalte P_{EXT}
- Starea A: P_L poate avea comenzi locale
- Starea B: activez accesul ext la M_{LD} – semnalele asociate starii
- Obs: Mem locala e vazuta de fiec proc intr-un ac spatiu de mem, iar de proc ext in spatiul complementar → trebuie sa existe o unitate de translatore a adreselor
- Mag ocupata: nu stii ciclul masina ai unui proc extern. Numai cand magistrala e libera un P_{EXT} a terminat ciclul masina curent → abordare top-down