

Lista subiectelor pentru anul 2 Informatică – L2

A. Teorie

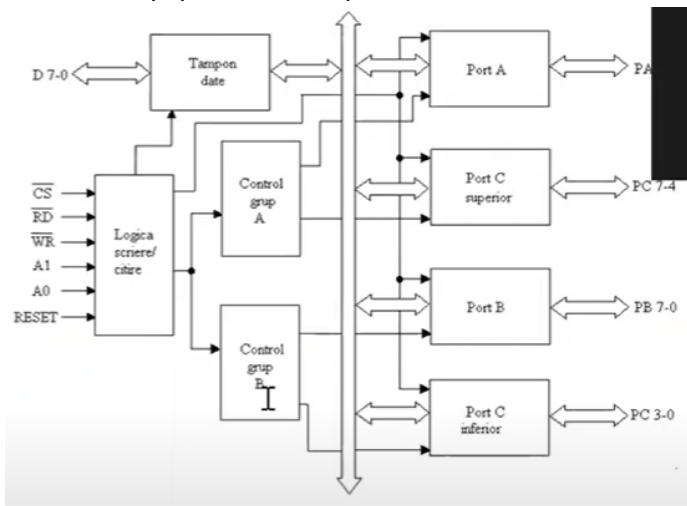
1. Câte linii sunt necesare pentru un transfer paralel? Justificați.

Numărul de linii care este necesar pentru un transfer paralel depinde în principal de dimensiunea datei care se transferă (un octet, un cuvânt, un dublu-cuvânt). Într-o situație în care se face transferul paralel al unui octet, sunt necesare 11 linii (8 linii octet+2 linii dialog+1 linie masă). Dacă ar fi să generalizăm la tipurile de date, formula ar fi: număr biti/linii tip date+2 biti/linii dialog+1 linie masă.

2. Ce este circuitul 8255? Prezentați caracteristicile sale.

În familia x86 există circuitul 8255 specializat pentru comunicări paralele. Are 24 de linii de intrare/ieșire pentru legătura la aplicații, iar ele pot fi configurate sau grupate în mai multe feluri în mai multe feluri (2 grupe de câte 12 linii de intrare sau ieșire, fără semnale de dialog). Pentru comunicarea cu procesorul, există magistrala de date, semnal de selecție R/W. În plus, mai are nevoie de încă două linii pentru a selecta porturile din interiorul circuitului (în interior sunt 4 porturi: A, B, C și cel de comandă).

3. Prezentați și comentați structura internă a circuitului 8255.



Avem o magistrală internă la care se leagă toate blocurile.

Tamponul de date separă magistrala de date externă de comunicarea internă

Logica de scriere/citire care primește toate semnalele

În partea dreaptă avem resursele pentru comunicarea cu

aplicațiile (porturile A, B, C). Se specifică faptul că portul C este "rupt" în două, existând portul C superior și inferior.

Mai exista Control grup A/B in structura, care nu este vizibil pentru noi ca utilizatori, care gestioneaza resursele aferente porturilor.

4. Cum se programează circuitul 8255 (nu este necesară structura cuvintelor ci doar ce anume i se comunică circuitului)?

Circuitul 8255 se programeaza prin transmiterea unui cuvânt de comanda la adresa portului cuvântului de comanda. Porturile pot lucra in trei moduri:

- modul 0 (numit mod intrare/iesire pentru porturile A, B si C)
- modul 1 (numit mod intrare/iesire pe dialog pentru porturile A si

B)

- modul 2 (numit mod bidirectional cu dialog, doar pt portul A)

In plus mai avem posibilitatea ca, printr-o singura instructiune de iesire, microprocesorul sa comande iesiri individuale ale portului C - e nevoie de un cuvânt de comanda special. Aceasta caracteristica este utila atunci cand folosim portul C sa comande mai multe dispozitive si se doreste folosirea unui singur dispozitiv, fara a afecta utilizarea celorlalte. Se mai poate genera un semnal care poate fi folosit ca cerere de intrerupere catre procesor.

Daca bitul 7 este 1: Circuitului i se comunica starea porturilor A, B, C superior si C inferior si modul de selectie.

Daca bitul 7 este 0: se intra in modul bit set/reset.

5. Descrieți unul din modurile de lucru cu dialog ale circuitului 8255.

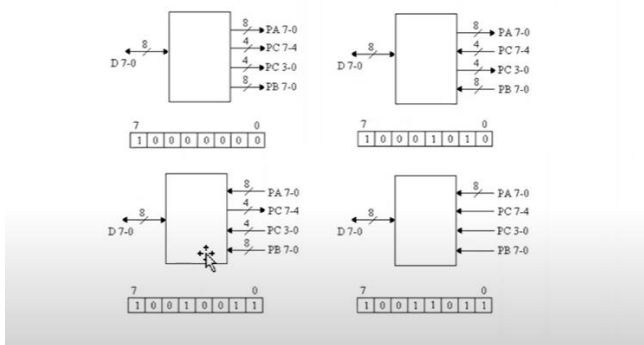
Modul 2 se mai numeste si mod bidirectional cu dialog.

Semnalele de dialog au aceleasi semnificatii ca in modul 1, iar pozitiile lor relative sunt aceleasi.

Caracteristici :

- doar grupul A poate lucra in acest mod
- grupul A este alcatuit din portul A si 5 semnale de dialog care folosesc ranguri ale portului C
- grupul B poate fi programat in modul 1 sau 0
- exista facilitatea de memorare si la intrare, si la iesire
- portul C are si rol de registru de stare

- 4 configurații posibile cu porturile A, B și C în modul 0 și cuvintele de comandă corespunzătoare:



Aici avem 4 configurații din cele 16 (cred că e OK una sau două, și încercat alte combinații). În primul exemplu toate porturile sunt declarate ieșiri (se pot vedea numerele transmise ca semnal), în următoarele 2 avem 2 intrări, două ieșiri, iar ultimul exemplu ne prezintă toate nodurile ca intrare.

Mai există și modul 1 și 2, dar sper că ne da la alegere.

6. Care sunt posibilitățile de conectare a porturilor la o UC cu microprocesor?

Există trei posibilități de conectare a porturilor la o UC cu microprocesor:

- intrarea/ieșirea programabilă în care transferul se desfășoară prin program. Este o soluție simplă, cu resurse puține, dar are ca dezavantaj timpul care este folosit neeficient.

- întreruperi în care procesorul este întrerupt la transferul fiecărui octet, după care acesta revine înapoi la programul pe care îl desfășoară. Un dezavantaj la acest tip de conectare este că nu toate întreruperile pot fi acceptate.

- Accesul direct la memorie (DMA) unde transferul are loc direct între memorie și periferic cel mai frecvent (mai poate apărea transfer de tip periferic-periferic și memorie-memorie, neutilizat pentru că este mai rapid prin UC). UC nu este implicată în acest transfer, dar este nevoie de un hardware în plus.

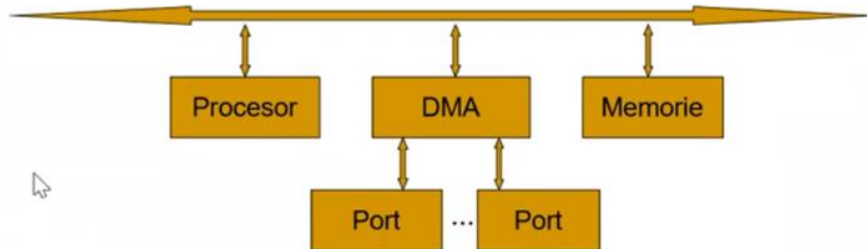
7. Prezentăm 3 configurații DMA. Caracteristici.

Magistrală comună, controler DMA separat. Pentru această configurație avem pe aceeași magistrală un procesor, memorie, DMA și porturi.

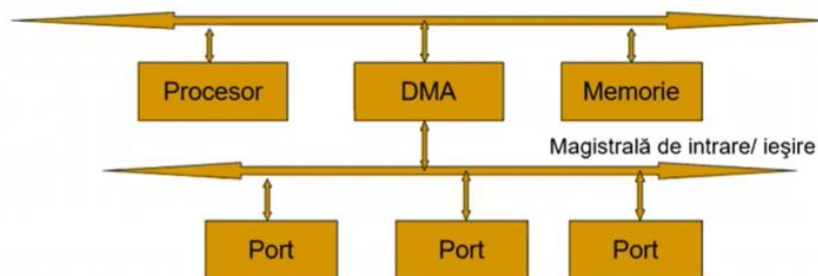
Fiecare transfer DMA folosește magistrala de două ori (Port → DMA și DMA → memorie). UC se oprește de două ori pe transfer pt că se folosește aceleași magistrale.



Magistrala comuna, controler DMA integrat. La aceasta configuratie, partea de periferie (porturi) este mutata la nivelul DMA-ului, astfel castigandu-se timp. Fiecare transfer foloseste magistrala o data (DMA->memorie). UC este oprit o singura data pe transfer.



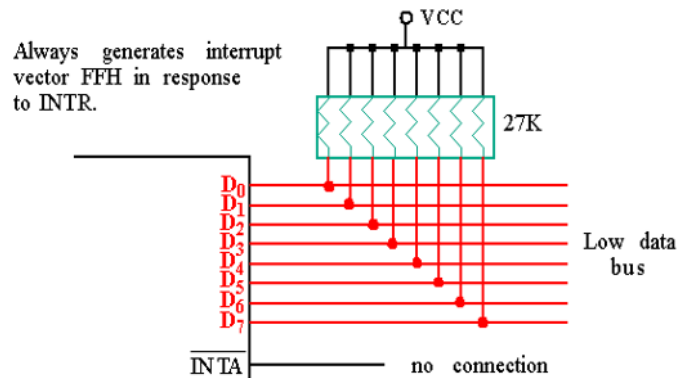
Magistrala de intrare/iesire distincta. In configuratia de acest tip, controlerul DMA este legat la porturi altfel, folosind o magistrala de intrare/iesire, nu prin canale distincte. UC este oprit o singura data pe transfer.



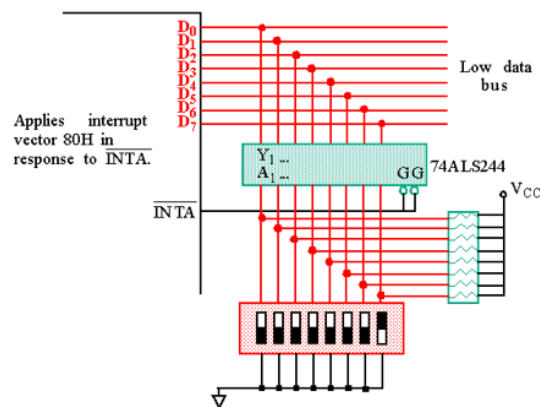
8. Descrieți întreruperile externe.

Întreruperile externe sunt provocate de evenimente externe microprocesorului care necesita atentie imediata din partea acestuia. Cel care a provocat intreruperea trebuie sa tina cererea activa pana cand microprocesorul devine disponibil si rapsunde si sa se identifice la cererea microprocesorului. Apar doua tipuri de intreruperi externe, cele mascabile (care vin din partea unui microprocesor, iar mascarea se face dezactivand sistemul de intreruperi) si nemascabile (pe care microprocesorul se sesizeaza mereu, nu pot fi blocate decat prin hardware extern si sunt folosite pentru situatii catastrofale cum ar fi o eroare la memorie).

- Exemplu de logică simplă pentru generarea unui vector de întrerupere fix:



- Exemplu de logică pentru generarea unui vector de întrerupere care poate fi modificat:



Cu schemele de mai sus banuiesc ca se poate face comparatii in ce difera de la una la alta.

9. Descrieți întreruperile interne.

Intreruperile interne apar la microprocesoarele peste 8 biti si la microcontrolere, purtand numele de exceptii sau traps la unele microprocesoare. Aceste intreruperi pot aparea din cauze interne ale procesorului, iar la microprocesoare nu pot fi mascate, spre deosebire de microcontrolere unde se poate realiza mascarea lor. La microprocesoare pot aparea din doua cauze: evenimente interne speciale si intreruperi software.

10. Ce este circuitul 8259? Prezentați caracteristicile sale.

Circuitul 8259 este un circuit specializat pentru intreruperi, pentru procesoarele Intel.

o necesitate pentru prioritatea minimă.

Ce este circuitul 8259? Prezentați caracteristicile sale.

Controlerul pentru întreruperi (PIC) 8259A

- Poate gestiona și prioritiza 8 cereri de întrerupere;
- Poate fi legat în cascadă cu alte 8 circuite similare asigurând astfel gestionarea și prioritizarea a 64 cereri de întrerupere;
- Posibilitate de mascare individuală a cererilor;
- Mai multe moduri de gestionare a întreruperilor:
 - Fully nested,
 - Rotating priority
 - Special mask și
 - Poll
- Necesită 2 - 4 cuvinte de inițializare (ICW1 - 4) și 4 cuvinte de operare (OCW1 - 4);
- Generează vectorul de întrerupere asociat cererii luată în considerare de procesor.

11. Justificați existența unui circuit de memorare (bistabil, registru) în schema de comandă a unui led prin program.

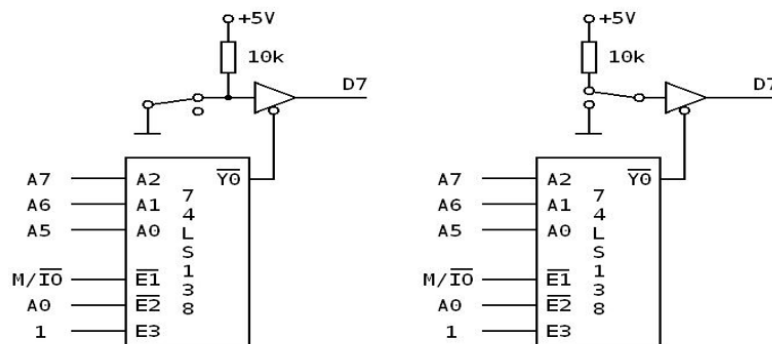
Comanda vine pe magistrala de date, dar pe magistrala de date informația se schimbă continuu și nu se poate comanda ca un LED să fie aprins un timp extrem de scurt, deoarece nu avem posibilitatea de a sesiza acest lucru. Astfel, avem nevoie de un circuit de memorare al semnalului primit până când comanda se schimbă. Cel mai simplu circuit de memorare este un bistabil pentru cazul prezentat.

12. Descrieți o soluție pentru citirea stării unui comutator prin program.

(în curs, avem 2 soluții, alegem una)

Descrieți o soluție pentru citirea stării unui comutator prin program

Soluția A:



Secvența pentru soluția a:

```
IN    AL,00H
AND   AL,80H
JZ    SUS ; ramura corespunzătoare poziției din figură
        ; ramura corespunzătoare poziției contrare
```

13. Care este diferența între arhitectura și microarhitectura unui microprocesor?

(s-a prezentat la începutul cursului cu microprocesoare moderne)

Arhitectura este setul de instrucțiuni, registrele sau structura datelor în memorie, utilizatorul având acces la ele, în timp ce microarhitectura reprezintă implementarea arhitecturii pe suportul fizic.

14. Dați exemple de arhitecturi ale seturilor de instrucțiuni ale microprocesoarelor. Caracteristici.

CISC, RISC, Very long instruction word+explicații

Arhitecturi ale seturilor de instrucțiuni ale microprocesoarelor sunt cele CISC, RISC sau VLIW, fiecare având câteva lucruri caracteristice:

La CISC, instrucțiunile sunt lungi și complexe cu mai multe moduri de adresare, iar durata de execuție a acestuia este mare. Un avantaj ar fi că programarea este simplă.

La RISC, instrucțiunile sunt simple, cu un timp de execuție scurt, dar efectul este redus. Spre deosebire de CISC, nu mai există variația modurilor de adresare. Compilatoarele au o viteză mai mare și sunt mai simple.

VLIW, instrucțiunile sunt lungi și se întind pe mai multe cuvinte. Acestea se pot grupa mai multe instrucțiuni (având câte una pe fiecare nucleu).

15. Descrieți evoluția soluțiilor implementate în microprocesoare pentru execuția instrucțiunilor (fără hyperthreading).

16. Care este cauza limitării eficienței benzii de asamblare? Prezentați soluții pentru creșterea eficienței benzii de asamblare.

Cauza limitării eficienței benzii de asamblare o reprezintă instrucțiunile care rup secvențialitatea în execuția unui cod, instrucțiuni de salt, de apelare a unor funcții, de reveniri din funcții.

Soluții pentru rezolvarea acestor limitări pot fi:

-prin ghicit: ghicirea ramurii a unui salt sau apel condiționat, iar dacă s-a ales bine, procesul nu pierde timp. Dacă s-a ales greșit, atunci banda trebuie golită de instrucțiunile intrate.

-întârzierea execuției: se utilizează instrucțiunea NOP și are rolul de a introduce o întârziere pentru a câștiga timp. În acest timp, instrucțiunea de salt

poate alege ramura pe care continua

-predictia ramurii: se iau in calcul anumite caracteristici (de exemplu istoria salturilor) dupa care se "ghiceste" ramura. Acest tip de solutie este util in bucle.

Cauzele sunt instructiunile care rup secventialitatea in executia unui cod: instructiuni de salt, de apel de functii, de reveniri din functii, intreruperi.

Solutiile sunt: executia speculativa: in care se ghiceste ramura unui salt sau apel conditionat, daca s-a ghicit bine, lucrurile se desfasoara in continuare fara pierdere de timp, daca s-a ghicit gresit banda trebuie golita de instructiunile care au intrat si trebuie aduse altele; intarzierea executiei: in care dupa instructiunea de salt sau apel conditionat se foloseste instructiunea NOP, ce se executa indiferent de ramura care continu si are rolul sa introduca intarziere ca sa castge timp, pentru ca instructiunea de salt sau apel conditionat sa stabileasca ramura pe care se continua; predictia ramurii: in care se ghiceste ramura pe baza anumitor considerente, de obicei istoria salturilor, solutia este bine venita cand se executa o bucla

17. Care sunt diferențele principale între microprocesoarele unice, multimicroprocesoare, microprocesoarele multitasking și cele multicore?

poze + mica analiza

Poze+analiza

Unice: executie secventiala, eventual se foloseste pipeline si exec out of order

Multimicroprocesoare: avem mai multe procesoare care concure pentru aceleasi resurse, fiecare multimicroprocesor poate sa isi execute codul lui => se castiga performanta. Pot aparea probleme cand acestea doresc acces la resursa comuna, mai ales ca pot avea loc in acelasi timp

Multitasking: sunt procesoare care au un singur nucleu, dar care se multiplexeaza intre mai multe task-uri/mai multe linii de executia.

Nucleul ia fiecare task pe rand (trece de la unul la altul) pana ce le executa pe toate. Problema cu token

Multicore: cea mai performanta, fiecare core isi ia un thread. Acest tip prezinta cele mai mici erori.

18. Prezentați schema bloc a unui microprocesor multicore cu hyperthreading.

Procesoarele unice au un singur nucleu și execuția instrucțiunilor este secvențială, au ca facilități pipeline și out-of-order execution, dar performanța este limitată de interdependențele între instrucțiuni sau de instrucțiuni care rup secvențialitatea.

Multimicroprocesoarele sunt sisteme cu mai multe microprocesoare și circuite, cu acces comun la resurse, și problemele apar la accesul resurselor comune.

Microprocesoarele multitasking au tot un singur nucleu, dar care se multiplexează pe mai multe taskuri, problema apare la comutarea taskurilor.

Microprocesoarele multicore în care există mai multe nuclee fizice incluse în același circuit, problema apare la cache-ului comun sau necomun între mai multe nuclee este încă nerezolvată.

19. Prezentați structura unui microprocesor multicore simetric și a unui asimetric.

B. Probleme

Aplicațiile prezentate la cursurile 8, 10 și 11. Acestea trebuie considerate ca exemple. Enunțurile și, posibil, și structurile vor fi modificate dar tipurile de probleme vor fi păstrate.