

Cuprins

- 1.Enumerati si caracterizati nivelurile ierarhice de proiectare ale unui sistem cu microprocesor.
- 2.Sa se prezinte o schema standard de sistem cu microprocesor.
- 3.Sa se clasifice standardele de magistrale de interconectare a unui sistem cu microprocesor.
- 4.Sa se realizeze o comparatie CISC-RISC.
- 5.Sa se prezinte o modalitate de evaluare (pentru frecventa de executie a instructiunilor sau prin program) sau de imbunatatire (prin utilizarea benzilor de asamblare sau a memoriilor intermediare „cache” incorporate) a performantei unui sistem cu microprocesor.
- 6.Sa se prezinte capabilitățile unui sistem cu microprocesor oferite pentru a lucra cu memoria virtuala.
- 7.Sa se prezinte modurile si tehnicile de translatore a adresei într-un sistem cu microprocesor.
- 8.Sa se prezinte tehnicile de protecție si securitate oferite de un sistem cu microprocesor.

Tehnici

- 9.Sa se prezinte modalitatea de diagnosticare si depanare a unui subsistem dintr-un sistem cu microprocesor.
- 10.Sa se realizeze o comparație între magistralele de interconectare dintr-un sistem cu microprocesor de la prima standardizare si pana in stadiul actual sau pentru un sistem actual cu microprocesor.
- 11.Sa se prezinte sintetic o magistrala standard dintr-un sistem actual cu microprocesor: caracteristici, categorii de semnale, protocol de funcționare, posibilități de conectare.
- 12.Sa se prezinte si sa se caracterizeze funcționalitățile oferite de o magistrala standard de interconectare dintr-un sistem cu microprocesor
- 13.Sa se prezinte evolutia microprocesoarelor utilizate intr-un sistem cu microprocesor.
- 14.Sa se clasifice sistemele cu microprocesor inainte si dupa aparitia procesorului pe 16 biti.
- 15.Sa se prezinte tipurile de placi de baza si de socket-uri dintr-un sistem cu microprocesor.
- 16.Sa se caracterizeze din punct de vedere functional resursele unui sistem cu microprocesor.
- 17.Sa se enumere si sa se caracterizeze tehnologiile introduse de Intel(sau AMD) intr-un sistem cu microprocesor CISC.
- 18.Sa se prezinte sintetic una dintre tehnologiile actuale Intel CISC (sau AMD CISC).

- 19.Sa se prezinte sintetic un microprocesor CISC(Intel Pentium, AMD K5 sau Motorola 68020).
- 20.Sa se clasifice smnalele unui procesor CISC (Intel Pentium, AMD K5 sau Motorola 68020) pe categorii functionale.
- 21.Sa se prezinte organigrama de functionare a unui procesor CISC (Intel Pentium, AMD K5 sau Motorola 68020) cu specificarea semnalelor de tranzitie intre stari.
- 22.Sa se prezinte un ciclu de magistrala reprezentativ pentru un procesor CISC(Intel Pentium, AMD K5 sau Motorola 68020).
- 23.Sa se prezinte sintetic un sistem cu microprocesor CISC(Intel Pentium, AMD K5 sau Motorola 68020).
- 24.Sa se prezinte si sa se caracterizeze modurile de lucru ale unui processor CISC(Intel Pentium, AMD K5 sau Motorola 68020).
- 25.Sa se prezinte resursele unui procesor CISC (Intel Pentium, AMD K5 sau Motorola 68020) din punct de vedere al programatorului.
- 26.Sa se prezinte modelul de adresare segmentata pentru procesorul Intel Pentium. cu precizarea modelelor de segmentare.
- 27.Sa se prezinte modelul de adresare paginata pentru procesorul Intel Pentium.
- 28.Sa se prezinte modul de lucru multitasking pentru procesorul Intel Pentium.
- 29.Sa se prezinte protectiile oferite de procesorul Intel Pentium.
- 30.Sa se realizeze o comparatie CISC-RISC.
- 31.Sa se prezinte sintetic o caracteristica RISC (arhitectura setului de instructiuni; arhitectura benzii de asamblare regulate; unitatile de executie).
- 32.Sa se prezinte sintetic un processor RISC(VLIW).
- 33.Sa se prezinte sintetic un sistem cu microprocesor RISC(VLIW).
- 34.Sa se clasifice semnalele unui microprocesor RISC(VLIW)pe categorii functionale.
- 35.Sa se prezinte organigrama de functionare a unui procesor RISC (VLIW) cu specificarea semnalelor de tranzitie intre stari.
- 36.Sa se prezinte un ciclu de magistrala reprezentativ pentru un procesor RISC (VLIW).
- 37.Sa se prezinte si sa se caracterizeze modurile de lucru ale unui processor RISC.
- 38.Sa se prezinte resursele unui procesor RISC (VLIW) din punct de vedere al programatorului.
- 39.Sa se prezinte un mecanism de translatare a adreselor pentru un procesor RISC (VLIW)
- 40.Sa se prezinte din punct de vedere functional arhitectura VLIW.

- 41.Sa se realizeze o comparatie intre modelul masinii de stare al arhitecturii VLIW si modelele masinii de stare pentru alte arhitecturi de procesare (SISD; SIMD; MIMD).
- 42.Sa se enumere si sa se caracterizeze succint tehnicile de crestere a paralelismului pentru arhitecturile VLIW.
- 43.Sa se prezinte succint modalitatile de pastrare a compatibilitatii arhitecturale VLIW.
- 44.Sa se prezinte sintetic modelul arhitectural VLIW pentru un procesor de semnal.
- 45.Sa se prezinte, pe baza schemei echivalente, functiile si aplicatiile unui procesor de semnal.
- 46.Sa se prezinte, pe baza schemei de structura,caracteristicile procesoarelor de semnal Motorola DSP56K (DSP96002).
- 47.Sa se prezinte sintetic un procesor de semnal (DSP56000; DSP96002; TMS320C50x(60x); ...) sau un procesor embedded..
- 48.Sa se prezinte sintetic un sistem cu procesor de semnal sau cu procesor embedded.
- 49.Sa se clasifice semnalele unui procesor de semnal sau un procesor embedded pe categorii functionale.
- 50.Sa se prezinte porturile de interfata ale unui procesor de semnal sau ale unui procesor embedded.
- 51.Sa se prezinte resursele unui procesor de semnal sau embedded din punct de vedere al programatorului.
- 52.Ciclul de viata al unui sistem incorporat.
- 53.Enumerati si caracterizati fazele de proiectare ale unui sistem incorporat.
- 54.Prezentati succint faza de creare a unui sistem incorporat.
- 55.Prezentati succint faza de implementare a unui sistem incorporat.
- 56.Prezentati succint faza de testare a unui sistem incorporat.
- 57.Prezentati succint faza de intretinere a unui sistem incorporat.

1.Enumerati si caracterizati nivelurile ierarhice de proiectare ale unui sistem cu microprocesor.

Un sistem este tipic vazut ca o structura ierarhica. Fiecare nivel de ierarhie reflecta un aspect major al sistemului si contine numai un mic subset din totalul parametrilor.

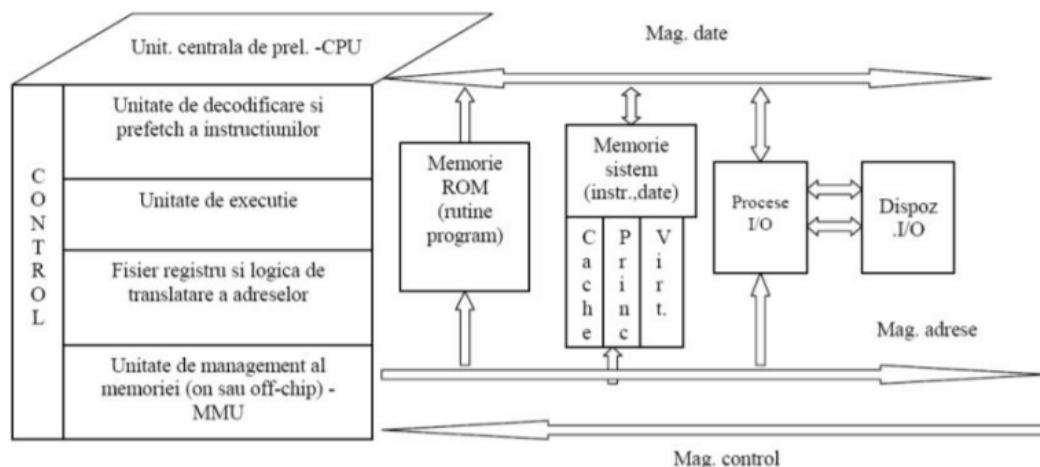
La **nivelul cel mai inalt al ierarhiei este programul de aplicatii**, adesea scris intr-un limbaj de nivel inalt. Astfel, limbajul poate fi proiectat fara a lua in considerare ceea ce ramane din ierarhie.

Mai jos de acest nivel se afla compilatorul si sistemul de operare. Sistemul de operare dirijeaza resusele sistemului si furnizeaza servicii de baza ale sistemului.

Sub acesta sta legatura arhitecturala, limita dintre hardware si software. Aceasta cheie de legatura consta din toate facilitatile si operatiile care sunt facute disponibile pentru programator (eg: registre, setul de instructiuni, capacitatile de adresare).

In final este nivelul hardware, care poate fi in plus stratificat intr-o organizare de sistem (structura pipe-line, microcod si flux de date, familii de circuite si tehnologia insasi).

2.Sa se prezinte o schema standard de sistem cu microprocesor.



Memoria cache poate fi on-chip sau off-chip.

Intr-un sistem cu μP exista de regula 3 magistrale:

1) magistrala bidirectionala de date (implementata cu logica permite lucrul direct cu memoria)

2) magistrala unidirectionala de adrese, conectata intern, in adrese (un numarator de program – PC), care si ea necesita a permite lucrul direct cu memoria

3) magistrala de control (comenzi), care pastreaza variatiile sincronizare la si de la μP (nu este necesara logica tri-state)

3.Sa se clasifice standardele de magistrale de interconectare a unui sistem cu microprocesor.

Pentru conectarea mai multor module este necesara o cale de comunicatie intre acestea, o magistrala. Magistralele au fost impartite in 2 mari clase: **magistrale paralele si magistrale seriale**.

a) paralele:

- IEEE 488, 625: Hewlett Packard Interface Bus (HPIB)
- IEEE 696 : S100 Bus
- ISA: IBM PC/AT • EISA: Compaq, HP
- NuBus: Macintosh Family • VME: Sun 3, Sun 4
- Q22 Bus: DEC PDP, DEC MICROVAX
- PCI: IBM PC

b) seriale:

- EIA RS 232C – comunicatie asincrona
- EIA RS 422 si 423 – comunicatie asincrona si sincrona

4.Sa se realizeze o comparatie CISC-RISC.

In prezent exista 2 mari familii de μP :

- procesoare cu set redus de instructiuni (RISC- Reduced Instruction Set Computer)
- procesoare cu set complex de instructiuni (CISC – Complex Instruction Set Computer).

Pentru a face o comparatie intre procesoare RISC si CISC se vor prezenta caracteristicile principale ale celor 2 arhitecturi, in momentul aparitiei procesoarelor RISC. Caracteristicile arhitecturilor

RISC:

- 1) instructiunile trebuie sa fie in numar redus (<100) si sa se execute intr-un singur ceas
- 2) accesul la memorie sunt limitate la 2 instructiuni, numite load si store
- 3) decodificarea instructiunilor este cablata
- 4) jocul de instructiuni trebuie sa se bazeze pe un format fix (de regula 32 biti), cu moduri de adresare simple (2 moduri: load si store)
- 5) fiecare operatie complexa trebuie sa fie rejectata la nivelul compilatorului
- 6) ansamblul arhitecturii trebuie sa profite la maximum de pe urma modului pipeline
- 7) procesorul trebuie sa posede un numar mare de registre interne (>10), cu sau fara tehnica de ferestre
- 8) procesorul trebuie sa fie adaptat adaugarii sistemelor de memorie cache si coprocesoarelor
- 10) ~ 100 MIPS Caracteristicile arhitecturilor

CISC:

- 1) set cat mai larg de instructiuni (>100)
- 2) numar cat mai mare posibil de moduri de adresare (>4)
- 3) mai multe formate de instructiuni (>4), deci lungime variabila a instructiunilor
- 4) cele mai multe instructiuni se executa in mai mult de un ciclu masina
- 5) accesul la memorie se fac prin tipuri variate de instructiuni
- 6) existenta unui numar relativ mic de registre interne (<10)
- 7) control μ programat

- 8) instructiunile masina sunt de un nivel relativ inalt
- 9) ~ 10 MIPS

Paralela RISC vs. CISC:

- viteza de executie ridicata la RISC fata de CISC datorita numarului mai mic de instructiuni de calcul.
- RISC reprezintă un set de calculatoare cu instructiuni reduse. CISC reprezintă setul complex de instructiuni pentru calculatoare.
- arhitectura RISC utilizează un set mic de instructiuni optimizate, în timp ce arhitectura CISC foloseşte un set de instructiuni complexe
- sistemele RISC sunt mai orientate pe aplicatii si nu sunt utile in anumite medii de programare; ele au insa eficienta ridicata pe aplicatii
- RISC este simplu şi necesită un ciclu de ceas pentru a executa instructiuni. CISC este complexă şi necesită mai multe cicluri de ceas pentru a executa o instructiune.
- RISC are mai multe registre, în timp ce CISC are mai puţine registre
- Deoarece RISC are instructiuni simple, lungimea programului este lungă. Dar, deoarece CISC are instructiuni complexe, lungimea programului este scurtă.

5. Sa se prezinte o modalitate de evaluare (pentru frecventa de executie a instructiunilor sau prin program) sau de imbunatatire (prin utilizarea benzilor de asamblare sau a memoriilor intermediare „cache” incorporate) a performantei unui sistem cu microprocesor.

a. $MIPS = I / (T_{cpu} * 10^6) = R / (CPI * 10^6)$ [milioane instructiuni/secunda]

I - arhitectura setului de instructiuni si tehnologia de compilare T_{cpu} – timp sistem CPU [sec/program]

R – tehnologia hardware si organizarea lui [ciclii ceas/sec]

CPI – organizarea si arhitectura setului de instructiuni (nr mediu de cicluri de ceas per instructiune) [ciclii ceas/instructiune]

MIPS este dependent de setul de instructiuni, facand astfel dificil de comparat MIPS-urile calculatoarelor cu diferite seturi de instructiuni.

MIPS variaza intre programe pe aceeasi masina si poate varia invers cu performanta. De obicei se calculeaza un

MIPS relativ = $T_{exec \text{ program masina de referinta}} / T_{exec \text{ pe alta masina}} \times MIPS_{\text{masina de referinta}}$

b. $MFLOPS = Nr. \text{operatii VM din program} / (T_{cpu} * 10^6)$ [milioane instructiuni/secunda]

MFLOPS este dependent de masina si de program. Pentru ca nu este dependent de instructiuni, este destinat cu brio compararii diferitelor μ sisteme. Din pacate, MFLOPS considera in general operatiile in VM, netinand cont de setul diferit de operatii in VM al diferitelor masini sau de viteza diferitelor operatii in VM ale aceluiasi set. De aceea, in general un benchmark (program de evaluare a performantelor) calculeaza un numar normalizat de operatii in VM

6.Sa se prezinte capabilitățile unui sistem cu microprocesor oferite pentru a lucra cu memoria virtuala.

Memoria virtuala reprezinta o tehnica de organizare a memoriei prin intermediul careia programatorul vede un spatiu virtual de adresare foarte mare si care, fara ca programatorul sa simta, este mapat în memoria fizic disponibila.

În cazul Memoriei Virtuale, memoria principala este analoaga memoriei cache între CPU (Central Processing Unit) si memoria principala, numai ca de aceasta data ea se situeaza între CPU si discul hard. Deci MP se comporta oarecum ca un cache între CPU si hard disk.

Prin mecanismele de MV se mareste probabilitatea ca informatia ce se doreste a fi accesata de catre CPU din spatiul virtual (disc), sa se afle în MP, reducându-se astfel dramatic timpul de acces. Spatiul virtual de adresare este împartit în entitati de capacitate fixa (4 sau 64 Kb actualmente), numite pagini.

O pagina poate fi mapata în MP sau pe disc. În general, prin mecanismele de MV, MP contine paginile cel mai recent accesate de catre un program, ea fiind dupa cum am mai aratat, pe post de cache între CPU si discul hard.

Segmentarea constituie o alta varianta de implementare a MV, care utilizeaza în locul paginilor de lungime fixa, entitati de lungime variabila zise segmente. În segmentare, adresa virtuala este constituita din 2 cuvinte: o baza a segmentului si respectiv un deplasament (offset) în cadrul segmentului.

7. Sa se prezinte modurile si tehnicile de translatare a adresei într-un sistem cu microprocesor.

Mecanismul de translatare a adreselor divide memoria in blocuri si mapeaza blocuri logice la blocuri fizice.

Sunt **3 moduri de baza** pt translatarea adresei:

a) paginare

Memoria principala este impartita in blocuri de marime fixa (cadre de pagini – page frames).

Subimpartirea spatiului de adresare logica a programului in pagini este facuta de sistem. Paginile sunt uzual de marime relativ mica fata de totalul memoriei. Dimensiunile tipice: 256 sau 2048 octeti

b) segmentare

Paginile sunt de marime variabila (segmente) si au dimensiuni mari 64K. Incarcarea programului in memoria principala nu il leaga la pozitia in spatiul de adresare logica. Segmentarea difera de organizarea memoriei.

c) paginare / segmentare combinata

Partea segmentata a schemei administreaza spatiul virtual prin impartirea programului in segmente, in timp ce cea paginata administreaza spatiul fizic care este impartit in cadre de pagini

Tehnici:

a) **Tehnica tabelii de translatare** se bazeaza pe translatarea tabelelor localizate in memoria principala: harta tabelilor de paginare in sistem cu paginare si harta tabelilor de segmente in cele cu segmentare. Intrarile in tabela contin informatii pentru a translata adrese logice in fizice ca si date suplimentare pentru protectie si pentru alegerea de plasare si replasare.

b) **Tehnica registrului de translatare** se bazeaza in principal pe registre individuale care contin parametrii de translatare. In cazul registrelor accesibile dupa adresa, campul de adresa logica identifica reg care pastreaza adresa fizica de baza. Aceasta tehnica solicita incarcarea noilor valori in registrul de translatare ori de cate ori un comutator de task-uri este executat, ceea ce nu este eficient pentru sistemele mari, deoarece necesita overhead in SO.

8.Sa se prezinte tehnicile de protecție si securitate oferite de un sistem cu microprocesor.

Tehnici

- **Sistem cu protectie ierarhizata / Sistem cu protectie inel ->**

Consta intr-o ierarhie de nivele de protectie **de la cel mai privilegiat pana la cel mai putin privilegiat**. un program poate accesa numai date care reprezinta acelasi inel sau intr-un inel mai putin privilegiat - un program poate chema servicii care rezida in acelasi inel sau unul mai privilegiat.

- **Sistem cu protectie neierahizata / sistem cu protectie bazata pe capabilitate** Se defineste o tabela de operatii pt fiecare task, tabela care specifica operatiile care pot afecta celalate taskuri din sistem. **Pentru a executa o operatie care ar putea afecta alt task, un task trebuie sa aiba capabilitati corespunzatoare in tabela sa de operatie**. metoda este implementata de regula in SO.

Alte metode:

- **moduri de lucru CPU de diferite nivele de privilegiu:**

modul supervizor (kernel, executiv) în care un proces poate sa execute orice instructiuni si sa acceseze oricare resurse si respectiv

modul user în care un proces are o multime de restrictii legate de protectia si securitatea sistemului

- **control strict al S.O. asupra tranzitiilor din user în kernel** (prin CALL GATES-uri la o anumita adresa determinata din spatiul de cod kernel)
- protectie a paginilor prin drepturi de acces la pagina (read only, read/write).

9.Sa se prezinte modalitatea de diagnosticare si depanare a unui subsistem dintr-un sistem cu microprocesor.



10.Sa se realizeze o comparație între magistralele de interconectare dintr-un sistem cu microprocesor de la prima standardizare si pana in stadiul actual sau pentru un sistem actual cu microprocesor.

ISA

- +cost scazut
- +foarte utilizat
- viteza mica
- mult mai vechi

Microchanel

- +mai rapid ca ISA
- invechit

PCI

- +viteza foarte mare, la nivel de placa dominanta
- incompatibil cu sisteme vechi
- scump

USB1.1

- +cost redus
- +ideal pentru sisteme portabile
- mai lent ca PCI
- necompatibil cu periferice vechi

USB2.0

- +identic cu USB1.1
- +viteza mare
- +compatibile cu mai multe periferice
- mai lent ca PCI
- nu este compatibil cu periferice vechi

11.Sa se prezinte sintetic o magistrala standard dintr-un sistem actual cu microprocesor: caracteristici, categorii de semnale, protocol de funcționare, posibilități de conectare.

Semnale PCI

FRAME

- i/o tristated
- Cycle Frame
- este determinat de un master pentru a indica inceputul si sfarsitul unei tranzactii (activat la inceput si dezactivat la sfarsit)

PGNT

- input
- PCI Grant
- indica permiterea accesului la magistrala PCI de catre arbitrul de magistrala

PREQ

- output
- PCI Request
- este activat spre arbitrul PCI pentru a cere magistrala PCI

MEMREQ

- input
- Memory Request
- este generat de un bridge standard de magistrala PCI pentru a garanta accesul intarziat de master standard de magistrala la memoria principala
- in conjunctie cu FLSHREQ#
- odata ce bufferele au fost golite, MEMACK# este activat continuu pana la dezactivare

MEMREQ#

MEMACK

- output
- Memory Acknowledge
- este generat ca raspuns la FLSHREQ# sau MEMREQ# de la un bridge de magistrala

Protocolul de functionare: Un agent cere magistrala PCI prin activarea PREQ#. Cand arbitrul determina ca un agent poate utiliza magistrala PCI, el activeaza PGNT# catre agent.

Reguli:

- a) daca PGNT# este dezactivat odata cu activarea FRAME, tranzactia pe magistrala este valida si va continua
- b) un PGNT # poate fi negat odata cu activarea altuia, daca magistrala nu este in stare - idle (No action); totusi, o intarziere de 1 ceas este inserata
- c) cat timp FRAME# este negat, PGNT# poate fi negat in orice moment pentru a servi masterul cel mai prioritar
- d) daca MEMREQ# si MEMACK# sunt activate, odata ce masterul este permis pe magistrala PCI, arbitrul nu va schimba permisia pana ce masterul isi anuleaza cererea.

12.Sa se prezinte si sa se caracterizeze funcționalitățile oferite de o magistrala standard de interconectare dintr-un sistem cu microprocesor

Magistrala ISA este prima magistrala specializata pentru PC. Este pe 8/16 biti si ofera interfatarea cu multe de dispozitive. Actualmente ISA este una din cele mai folosite magistrale secundare intr-un PC(ca magistrala principala sunt folosite PCI, VME).

Semnale:

- de date:
 - SD (System data)
- de adrese
 - LA (Unlatched data) | => ambele sunt intrari pentru un master extern ISA/
 - SA (System address) | iesiri pentru un master intern ISA
- de selectie ciclu curent:
 - MEMR (Memory read)
 - MEMW (Memory write)
 - IOR (I/O read)
 - IOW (I/O write)
- de sincronizare si control:
 - SYSCLK (System clock)
- intreruperi (Interrupt request)
 - IRQ8
 - IRQ12/M
- DMA:
 - DREQ (DMA Request) - cere magistrala de la DMA
 - DACK (DMA Acknowledge) - cerere de raspuns de la DREQ

13.Sa se prezinte evolutia microprocesoarelor utilizate intr-un sistem cu microprocesor.

-primul microprocesor I4004 pe 4 biti

-primul pe 8 biti I8008, dar primul de succes pe 8 biti a fost 8080

-primul uP pe 16 biti a fost 8086,

-primul uP pe 32 biti a 80386

-Pentium a fost primul cu pipeline

-primul pe 64 biti a fost Itanium

Alte familii de microprocesoare:

- Motorola
- Texas Instruments: -procesoare de semnal: TMS320c10/20/30/50/80
- Microchip: microcontrolare: PIC12/16/18
- MIPS
- ARM

14.Sa se clasifice sistemele cu microprocesor inainte si dupa aparitia procesorului pe 16 biti.

• **Microcalculatoare:**

- sunt calculatoare a caror unitate centrala este un microprocesor.
- lucreaza pe 8, 16 sau 32 de biti
- sunt sisteme interactive, monoutilizator
- primele microcalculatoare au fost calculatoare personale. Aceste calculatoare puteau lucra direct în limbajul Basic fiindca memoria lor interna cuprindea o parte care nu se stergea în absenta curentului electric si în care utilizatorul nu putea scrie (ROM), unde era înregistrat din fabricatie un interpretor Basic
- cel mai raspândit sistem de operare al microcalculatoarelor este DOS, care are la baza primul sistem de operare al microcalculatoarelor, CP/M, îmbunatatit cu principii din UNIX.

• **Minicalculatoare:**

- erau sisteme interactive
- sistem incorporat multiutilizator: la un moment dat mai multi utilizatori (câteva zeci sau chiar câteva sute) putea folosi calculatorul prin intermediul terminalelor.
- sistem de operare RSX
- sistemul de calcul lucra în regim de multitasking (multiprogramare): utilizatorii aveau acces pe rând la resursele sistemului
- din punctul de vedere al arhitecturii, caracteristica pentru minicalculatoare este existenta unei "magistrale de informatii", numite BUS, prin intermediul careia se realizeaza comunicarea între procesor, memorie, terminale si alte periferice
- comunicarile sunt arbitrate de controler-ul de BUS, care preia astfel functiile canalului de intrare-iesire. Acesta da dreptul de initiere a unei comunicari în functie de prioritatea asociata celei care initiaza comunicare
- minicalculatoarele lucrau pe 16 si 32 de biti

• **Mainframe-uri:**

- calculatoare folosite în principal de organizații mari pentru aplicații critice, de obicei procesări de date masive, cum ar fi recensămintele, statisticile industriei

-termenul este folosit pentru a distinge calculatoarele comerciale de nivel înalt de calculatoarele mai puțin puternice.

• **Supercalculatoare:**

-au o memorie internă și o viteză de lucru foarte mari, fiind cele mai rapide tipuri de calculatoare

-sunt utilizate pentru aplicații specifice, care necesită calcule matematice complexe, mari consumatoare de timp și memorie, cum ar fi, grafică animată, fizică nucleară.

-cel mai cunoscut tip de supercalculator este CRAY

-lucrează pe 32 și 64 de biți și au o arhitectură performantă (de exemplu, conține sisteme multiprocesor – cu mai multe unități centrale)

-pentru arhitectură se folosește conceptul de arhitectură paralelă. În mod obișnuit, există un procesor principal, numit master, care le coordonează pe celelalte, dându-le spre execuție sarcini independente din programul utilizatorului sau punându-le în așteptare.

-sistemele proiectate ca arhitecturi paralele pot fi clasificate:

1. După fluxurile de date și de instrucțiuni:

-> (SISD) sisteme cu un flux de instrucțiuni și un flux de date

-> (SIMD) sisteme cu un flux de instrucțiuni și mai multe fluxuri de date

-> (MISD) sisteme cu mai multe fluxuri de instrucțiuni și un flux de date

-> (MIMD) sisteme cu mai multe fluxuri de instrucțiuni și mai multe fluxuri de date

2. După numărul de procesoare (granularitatea): zeci de procesoare autonome conectate ("ferme") sau sute, până la zeci de mii de procesoare conectate în paralel ("cuburi")

3. După modul de folosire a memoriei interne de către procesoare:

-> partajate

-> proprii.

15.Sa se prezinte tipurile de placi de baza si de socket-uri dintr-un sistem cu microprocesor.

I. Placi de baza

- **PC/XT** - standardul original de placa de baza creat de IBM pentru primul home computer, IBM-PC
- **AT** factor de forma (Advanced Technology) - primul factor de forma care a prins acceptare la scara larga, succesor pentru PC/XT
- **Baby AT** - succesorul de la IBM pentru placa de baza AT, echivalenta functional cu AT, dar a prins popularitate datorita dimensiunii semnificativ mai mici, de obicei fara port AGP
- **ATX** - "evolutia" de la factorul de forma Baby AT, e cel mai populat factor de forma disponibil astazi
- **ETX** - folosit in sisteme embedded si computere single-board
- **Mini-ATX** - la fel ca ATX, dar cu un 'footprint' mai mic
- **microATX** - ATX miniaturizat
- **FlexATX** - subset al microATX care ofera flexibilitate de design pentru placa de baza, pozitionarea componentelor si forma
- **LPX** - bazat pe un design de la Western Digital, permite carcase mai mici bazate pe placa de baza ATX prin rearanjarea cardurilor de expansiune intr-un riser(un card de expansiune in sine atasat la placa de baza)
- **Mini LPX** - un subset mai mic al LPX
- **NLX** - o placa de baza 'low-profile', care incorporeaza un 'riser', proiectata pentru a fi la curent cu trendurile din market. NLX nu a prins niciodata popularitate
- **BTX** (Balanced Technology Extended) - un nou standard propus de Intel ca un eventual succesor pentru ATX
- **microBTX/picoBTX** - subset mai mic al standardului BTX
- **Mini-ITX** - placa de baza compacta proiectata pentru computere de cost relativ redus in spatii mici
- **WTX** (Workstation Technology Extended) - o placa de baza mare(mai mare decat ATX) proiectata pentru workstation-uri de putere mare

II. Socket-uri

- **80486**: 486 Socket, Socket 1, Socket 2, Socket 3, Socket 6
- **Celeron**: Socket 370 (old Celeron), Socket 478, Socket T/LGA 775
- **Itanium**: PAC 418, PAC 611
- **Pentium**: Socket 4, Socket 5, Socket 7
- **Pentium Pro**: Socket 8
- **Pentium II**: Slot 1
- **Pentium III**: Slot 1, Socket 370 • **Pentium 4**: Socket 423, Socket 478, Socket T/LGA 775
- **Pentium M**: Socket 479 PGA & BGA
- **K5**: Socket 7
- **K6**: Socket 7, Super Socket 7
- **Athlon**: Slot A, Socket A, Socket 563
- **Athlon 64**: Socket 754, Socket 939, Socket AM2(940-pin)
- **Athlon 64 FX**: Socket 939, Socket 940, Socket AM2(940-pin)
- **Duron**: Socket A
- **Opteron**: Socket 939, Socket 940, Socket F(1207-pins)
- **Sempron**: Socket A, Socket 754

16. Sa se caracterizeze din punct de vedere functional resursele unui sistem cu microprocesor.

Nx586 citeste instructiuni CISC de la cache-ul de instructiuni si le memoreaza in bufferul prefetch care este împărțit în 3 parti, permitand procesorului sa foloseasca simultan 3 fluxuri de instructiuni. Acestea ajuta la tinerea ocupata a benzilor de asamblare. Din prefetch, instructiunile sunt duse intr-un decodicator programatic, unde pentru fiecare cache, o instructiune CISC este translatata in una sau mai multe instructiuni RISC 86. Instructiunile RISC 86 sunt trimise unitatii de executie.

In timp ce Nx586 este un procesor scalar din punct de vedere CISC, este superscalar din punct de vedere RISC. Ca o limitare, doar o instructiune RISC 86 poate iesi la unitatea de executie partiala per ciclul. Unitatile de executie sunt diferite: una genereaza adresele de load si store, una prelucreaza intregi inclusiv in hardware, cealalta unitate de intregi este simpla, iar a 4-a lucreaza in VM. Fiecare unitate de executie, inclusiv FPU, are o coada de instructiuni cu 14 intrari, toate instructiunile trebuie sa petreaca cel putin un ciclu de ceas in coada chiar daca unitatea de executie nu e ocupata. Din cauza timpului variabil de asteptare in coada a unei instructiuni, benzile de asamblare au o adancime nedeterminata => cel mai scurt timp de fetch e de 7 cicli de ceas.

17. Sa se enumere si sa se caracterizeze tehnologiile introduse de Intel(sau AMD) intr-un sistem cu microprocesor CISC.

- **Deep branch prediction** permite procesorului sa decodifice urmatoarele instructiuni pentru a menține pipeline-ul de instructiuni la capacitate maximă
- **Dynamic data flow analysis** presupune analiza in timp real a datelor care trec prin procesor pentru a determina dependintele si pentru a detecta oportunități de executie a instructiunilor in alta ordine.
- **Speculative execution** se referă la capacitatea procesorului de a executa instructiuni care se află dincolo de o ramură condiționată care nu a fost încă rezolvată și la final sa trimita rezultatele în ordinea în care se aflau în fluxul de instructiuni inițial.

18.Sa se prezinte sintetic una dintre tehnologiile actuale Intel CISC (sau AMD CISC).

Tehnologia Hyper-Threading a fost dezvoltată pentru a îmbunătăți performanța procesoarelor IA-32 atunci când execută sisteme de operare și cod multithread sau aplicații single thread sub medii multitasking. Tehnologia permite unui singur procesor fizic să execute concurent două sau mai multe threaduri separate de cod. Din punct de vedere al arhitecturii, un procesor IA-32 care suportă tehnologia HT constă în două sau mai multe procesoare logice. Fiecare procesor logic are un set întreg IA-32 de regiștri de date, de segment, de control și de debug. De asemenea, fiecare are propriul APIC(advanced programmable interrupt controller). Spre deosebire de configurația unui sistem MP tradițional care utilizează două sau mai multe procesoare fizice IA-32 separate, procesoarele logice sunt procesoare IA-32 care suportă tehnologia de hyper-threading și împart resursele nucleului unui procesor fizic. Incluzând motorul de execuție și interfața magistrală de sistem.

19.Sa se prezinte sintetic un microprocesor CISC(Intel Pentium, AMD K5 sau Motorola 68020).

Pentium 4 se bazează pe noua microarhitectură **Intel NetBurst**.

Microarhitectura Intel NetBurst oferă:

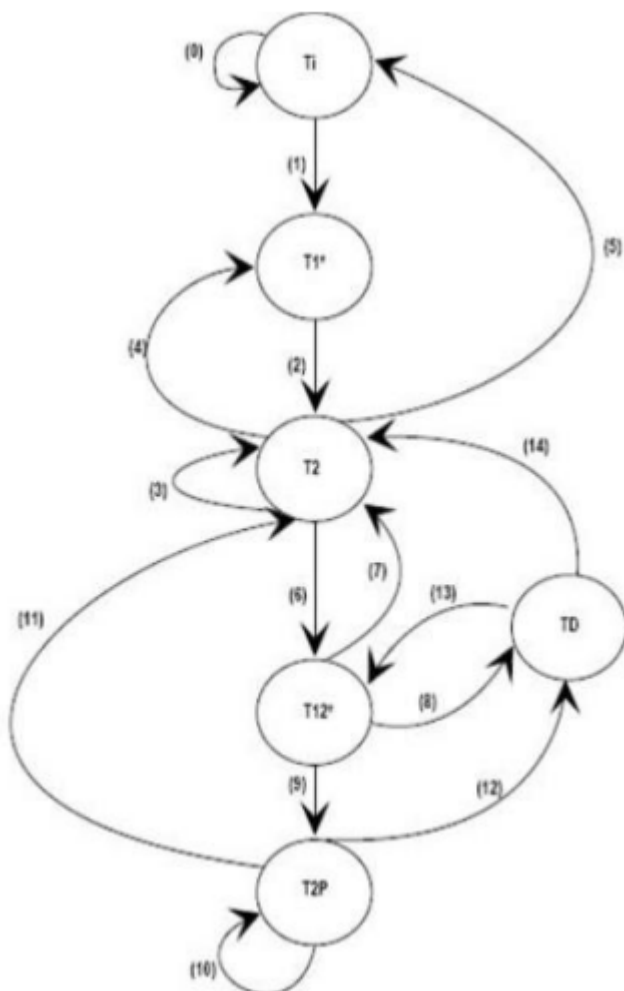
- **Motorul de execuție rapidă**
 - Unitățile aritmetice logice (ALU) rulează la o frecvență de două ori mai mare decât frecvența procesorului
 - Operațiunile de bază cu numere întregi pot fi expediate în 1/2 tick de ceas al procesorului
 - Oferă un randament mai mare și o latență redusă de execuție
- **Tehnologie Hyper-Pipelined**
 - Conductă profundă pentru a permite frecvențe de ceas de lider în industrie pentru computere desktop și servere
 - Margine de frecvență și scalabilitate pentru a continua conducerea în viitor
- **Execuție dinamică avansată**
 - Motor de execuție profund, în afara ordinului, speculativ
- **Până la 126 de instrucțiuni în zbor**
- **Până la 48 de încărcături și 24 de depozite în conductă**
 - Capacitate îmbunătățită de predicție a ramurilor
- **Reduce penalizarea de predicție greșită asociată cu conductele mai adânci**

20.Sa se clasifice smnalele unui procesor CISC (Intel Pentium, AMD K5 sau Motorola 68020) pe categorii functionale.

Funcție	Semnale
Ceas	CLK
Initializare	RESET, INIT, BF1-BF0
Magistrala de adresa	A31-A3, BE7#-BE0#
Masca de adresa	A20M#
Magistrala de date	D63-D0
Paritate adresa	AP, APCHK#
Suport APIC	PICCLK, PICD0-1
Paritate date	DP7-DP0, PCHK#, PEN#
Eroare paritate interna	IERR#
Eroare sistem	BUSCHK#
Pagina memorie cache	PCD, PWT
Definirea ciclului de magistrala	M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK#
Control magistrala	ADS#, ADSC#, BRDY#, BRDYC#, NA#
Control cache	KEN#, WB/WT#
Consistentă cache	AHOLD, EADS#, HIT#, HITM#, INV
Flush cache	FLUSH#
Ordine scrieri	EWBE#
Arbitrare magistrala	BOFF#, BREQ, HOLD, HLDA
Control magistrala Private Dual Processing	PBGNT#, PBREQ#, PHIT#, PHITM#
Intreruperi	INTR, NMI
Raportare erori floating point	FERR#, IGNNE#
Mod management sistem	SMI#, SMIACK#
Verificare redundanta functionala	FRCMC#, IERR#
Port TAP	TCK, TMS, TDI, TDO, TRST
Monitorizare performanta	PM0/BP0, PM1/BP1, BP3-2

Funcție	Semnale
Management putere	STPCLK#
Dual Processing	CPUTYP, D/P#
Debugging	R/S#, PRDY
Detectie tensiune	VCC2DET#

21.Sa se prezinte organigrama de functionare a unui procesor CISC (Intel Pentium, AMD K5 sau Motorola 68020) cu specificarea semnalelor de tranzitie intre stari.



I. Descriere tranzitii

- (0) Niciun request in asteptare
- (1) Request Pending
- (2) Se trece in T2 pentru a procesa transferul de date.
- (3) Sta in T2 pana se termina transferul
- (4) Un nou request cand ciclul curent s-a terminat
- (5) Nu exista request nou cand ciclul curent s-a terminat
- (6) apare un ciclu in asteptare
- (7) ciclul curent se termina si nu e nevoie de asteptare
- (8) ciclul curent se termina si e nevoie de asteptare
- (9) ciclul curent nu este terminat
- (10) se asteapta terminarea transferului pentru primul ciclu
- (11) termina ciclul si nu mai e nevoie de timp mort
- (12) termina ciclul si este nevoie de timp mort
- (13) request nou in asteptare

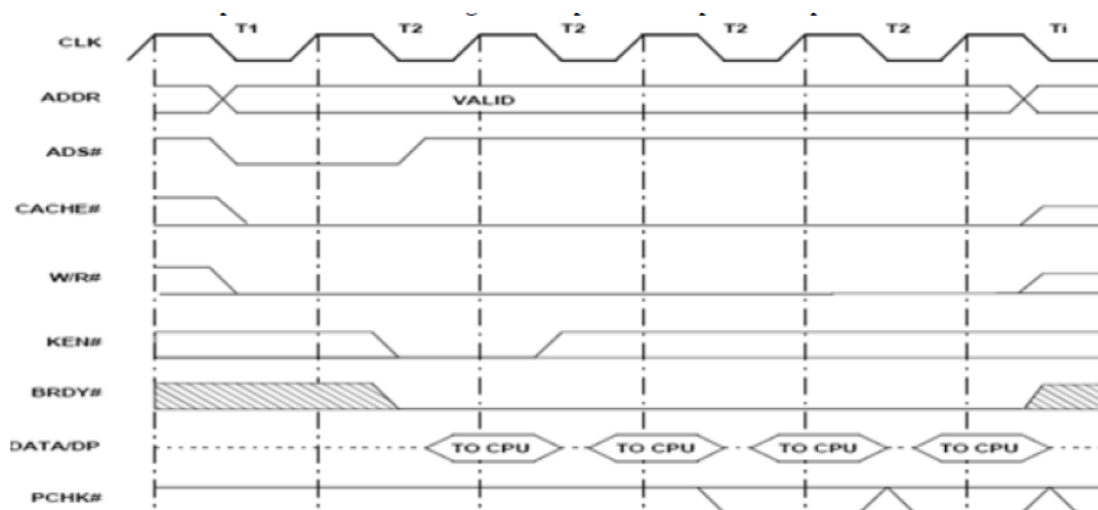
(14) nu exista request nou

II. Descriere stari

- Ti: idle => nu este rulat niciun ciclu pe magistrala
- T1: primul ceas din ciclul pe magistrala => se trimit adresele valide si statusul
- T2: al doilea ceas din ciclul pe magistrala => daca e write, se trimit datele si daca e read, se asteapta datele
- T12: stare care indica faptul ca exista 2 cicluri semnificativi pe magistrala => procesorul incepe al II-lea ciclu in acelasi timp cu transferarea datelor pentru primul ciclu
- T2P: stare care indica faptul ca sunt 2 cicluri de magistrala, ambii in al 2-lea ceas. Pentru primul ciclu se transfera datele, iar pentru al doilea adresa si statusul au fost trimise in T12.
- TD: un singur ciclu pe magistrala. Adresa si statusul au fost trimise in T12. Nu sunt testate datele si BRDY. E necesar un timp mort pentru ca nu se poate intoarce intre citiri si scrieri consecutive

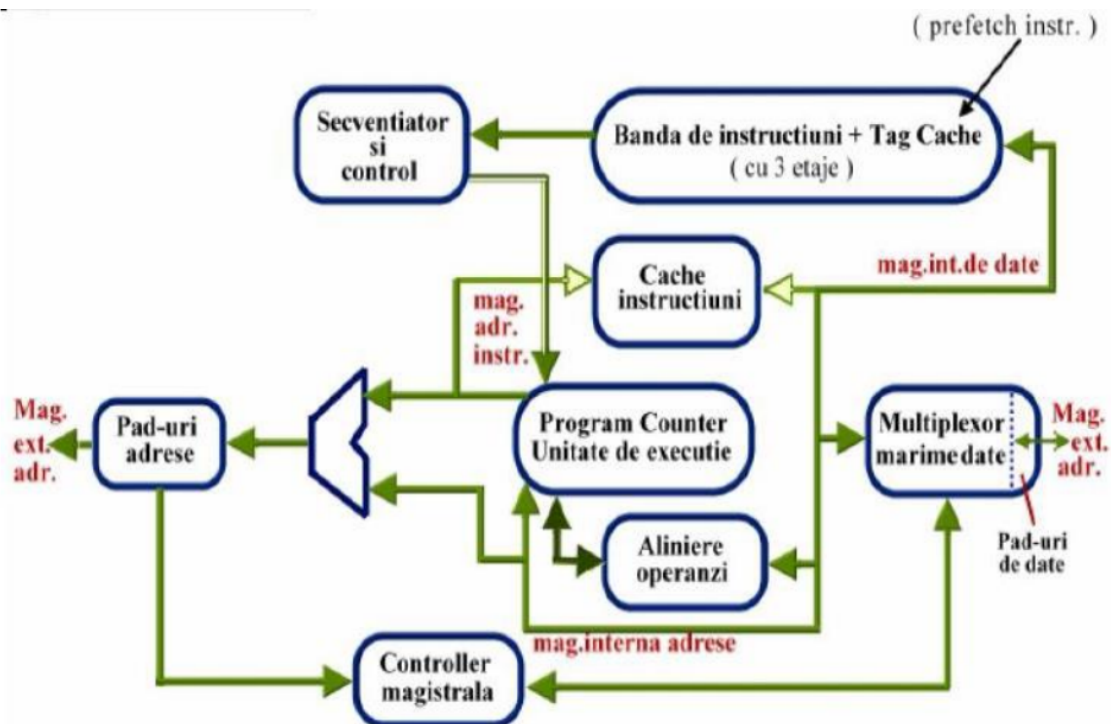
22. Sa se prezinte un ciclu de magistrala reprezentativ pentru un procesor CISC (Intel Pentium, AMD K5 sau Motorola 68020).

Burst Read cycle: Datele vor fi eșantionate numai în ceasul în care este returnat BRDY #, ceea ce înseamnă că datele nu trebuie trimise procesorului Pentium la fiecare ceas din ciclul burst.



23.Sa se prezinte sintetic un sistem cu microprocesor CISC(Intel Pentium, AMD K5 sau Motorola 68020).

Diagrama Bloc MC68020



Caracteristici:

- cod obiect compatibil cu MC68000;
- interfatare VIDE Graphics;
- Cache de instructiuni on-chip;
- Interfatare cu coprocesor (MC68881 si MC68882) si cu MMU paginat (MC68851)
- Microarhitectura pipelined; - Suport total pentru memorie virtuala;
- 4G direct adresabil x 8 spatii de adresare;
- 16 registre pe 32 biti, cu scop general;
- 2 registri pe 32 biti pentru 'stack pointer' si 5 registre de control speciale;
- 8 moduri de adresare si 7 tipuri de date;
- frecventa de lucru selectabila: 12.5, 16.67, 20, 25, 33.33 MHz;
- 2 moduri de lucru: utilizator (user) si supervizor (supervisor).

Descriere semnale:

Semnal	Descriere
CLK	Semnal ceas
FC0:2 (Function Code)	Identifica spatiul de adresare pentru fiecare ciclu de magistrala.
A0:31(A1:23)	Iesiri de adresa cu trei stari. La MC 68000 bitul A0 este folosit intern pentru a genera /UDS si /LDS.

/UDS, /LDS	Indica daca datele sunt transferate pe octetul superior sau inferior sau pe ambii octeti.
D0:31(D0:15)	Semnale bidirectionale de date cu trei stari; Se pot transfera 8b, 16b, 24b sau 32b per ciclu.
Semnale de control	
/OCS (Operand Cycle Start)	Indica inceputul primului ciclu extern pentru o instructiune prefetch sau transfer de date. Este activat impreuna cu /ECS
/ECS (External Cycle Start)	Indica inceputul unui ciclu de magistrala de orice tip
R(w)	citire (=1) / scriere (=0)
/RMC (Read-Modify-Write Cycle)	Este activat la inceputul primului ciclu de operatii RMC si ramane activ pe timpul tuturor ciclurilor de operatii RMC.
AS (Address Strobe)	Indica adresa valida pe magistrala de adrese
DS (Data Strobe)	Indica date valide pe magistrala de date
/DBEN (Data Buffer Enable)	Semnal de activare buffere de date externe; La citire este activat 1 ciclu de ceas dupa inceperea unui ciclu de magistrala si dezactivat odata ce /DS este dezactivat; la scriere este activat odata cu activarea lui /AS si tinut activ pe durata ciclului.
/DACK1, /DSACK0 (Data Transfer and Size Acknowledge)	Indica terminarea unei operatii cerute de transfer de date
Semnale de intrerupere	
/IPL2:0 (Interrupt Priority Level)	- cereri de intrerupere codificate pe nivelul de intrerupere de la circuitul prioritar; - sunt decodificate intern; - /IPL2 reprezinta cel mai semnificativ bit.
/IPEND (Interrupt Pending)	- intrerupere in asteptare, ce a depasit masca prioritatii intreruperii curente din registrul de stare (SR).
AVEC (Autovector)	- cere un autovector in timpul unui ciclu de recunoastere intrerupere (cand dispozitivul care a dat intreruperea nu poate furniza un numar de vector).
Semnale control arbitrare magistrala	
/RESET	Semnal bidirectional de initializare (ca iesire initializeaza doar dispozitivele externe)
/HALT	Indica suspendarea activitatii procesorului; sau cand este utilizat cu /BERR indica reintrarea procesorului in ciclul curent care a produs eroarea magistralei. La MC68000 poate fi si iesire pentru a indica o dubla conditie de eroare.
/BERR (Bus Error)	semnal de eroare; indica operatie nevalida, sau cu /HALT duce la reintrare in ciclul curent.
Semnale de control cache	
/CDIS (Cache Disable)	Dezactiveaza cache-ul on-chip pentru un utilizator

24. Sa se prezinte si sa se caracterizeze modurile de lucru ale unui processor CISC(Intel Pentium, AMD K5 sau Motorola 68020).

Motorola 68020 are doua moduri de lucru: utilizator (user) si supervizor (supervisor).

a) **Supervizor**

- cel mai privilegiat; toate instructiunile sunt executabile.
- $S = 1$
- Intr-un SO multitasking este mai eficient sa avem un spatiu de stiva supervizor asociata cu fiecare task utilizator (MSP) si un spatiu separat de stiva pentru intreruperi asociate task-urilor (ISP) ($M = 1 \Rightarrow \text{MSP}$; $M = 0 \Rightarrow \text{ISP}$)
- ISP poate fi utilizata in informatia de control intreruperi si spatiu de lucru asa cum cer rutinele de intreruperi.
- Cu $M = 0$ avem aceleasi operatii ca la modulul supervizor al MC 68000, 68008, 8010.
- M este afectat de instructiunile: MOVE la SR, ANDi la SR, EORi la SR, Ori la SR si RTE;
- Procesorul salveaza automat valoarea lui M si o sterge in SR, ca parte a procesarii unei exceptii pentru intreruperi.
- Toate exceptiile sunt executate la nivel supervizor.
- Toate ciclurile de magistrala generate in timpul procesarii exceptiei sunt referinte supervizor si toate accesele la stiva utilizeaza indicatorii stivei supervizoare active.

b) **Utilizator**

- Cel mai putin privilegiat;
- Majoritatea instructiunilor se executa la orice nivel de privilegiu, dar anumite instructiuni sunt privilegiate si se pot executa numai la nivel supervizor (ex. STOP, RESET).
- Pentru a preveni ca un program utilizator sa intre pe nivel supervizor, exceptand o maniera controlata, instructiunile care pot altera bitul S din registrul de stare SR sunt privilegiate.

Pentru a schimba nivelul de priilegiu de la utilizator la supervizor trebuie sa apara una din conditiile care fac ca procesorul sa execute procesarea unei exceptii.

Pentru reintoarcere in nivel privilegiat utilizator, o rutina sistem trebuie sa execute una din instructiunile (urmatoarele instructiuni pot modifica bitul S din SR):

- MOVE la SR
- ANDi la SR
- EORi la SR
- Ori la SR sau RTE

25. Sa se prezinte resursele unui procesor CISC (Intel Pentium, AMD K5 sau Motorola 68020) din punct de vedere al programatorului.

1. **Registre generale** (pot fi utilizate si ca registre index)

- a) D0 ÷ D7 = 8 registre de date pe 32 biti (pot fi operanzi pe 8, 16 sau 24 de biti);
- b) A0 ÷ A6 = 7 registre de adrese pe 32 biti (pot fi registre baza adresa)
- c) A7 - registru adresa pe 32 biti sau indicator stiva utilizator ;

2. **PC** - registru control program pe 32 de biti;

3. **CCR** - registru conditie cod

4. **A7' (ISP)** - registru adresa sau indicator stiva intrerupere in nivel supervisor

5. **A7'' (MSP)** - registru adresa sau indicator stiva master in nivel supervisor - SSP

6. **Registru de stare SR** :

7. **Registru vector baza** (VBR - Vector Base Register)

8. **Registre cod functional alternant**

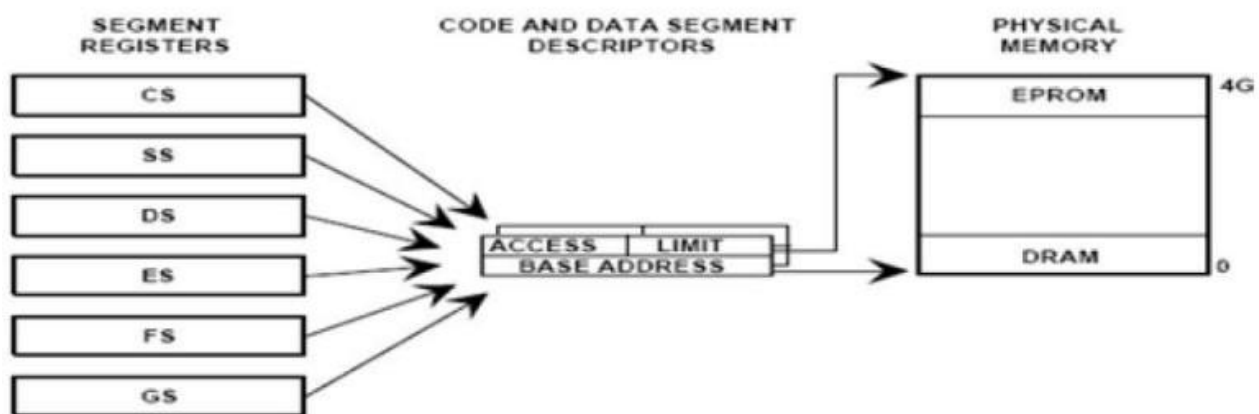
- a) SFC (System Function Code)
- b) DFC (Data Function Code)

9. **Registre cache:**

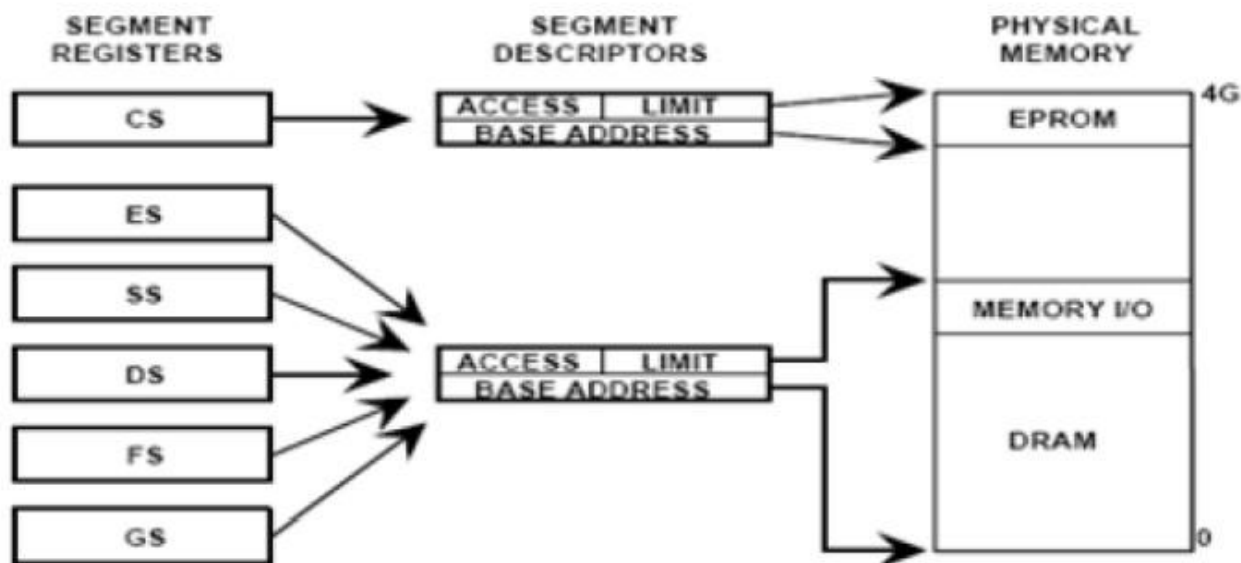
- a) Control (CARC - Address Control Register);
- b) Adresa (CAAR - Cache Address Register)

26.Sa se prezinte modelul de adresare segmentata pentru procesorul Intel Pentium, cu precizarea modelelor de segmentare.

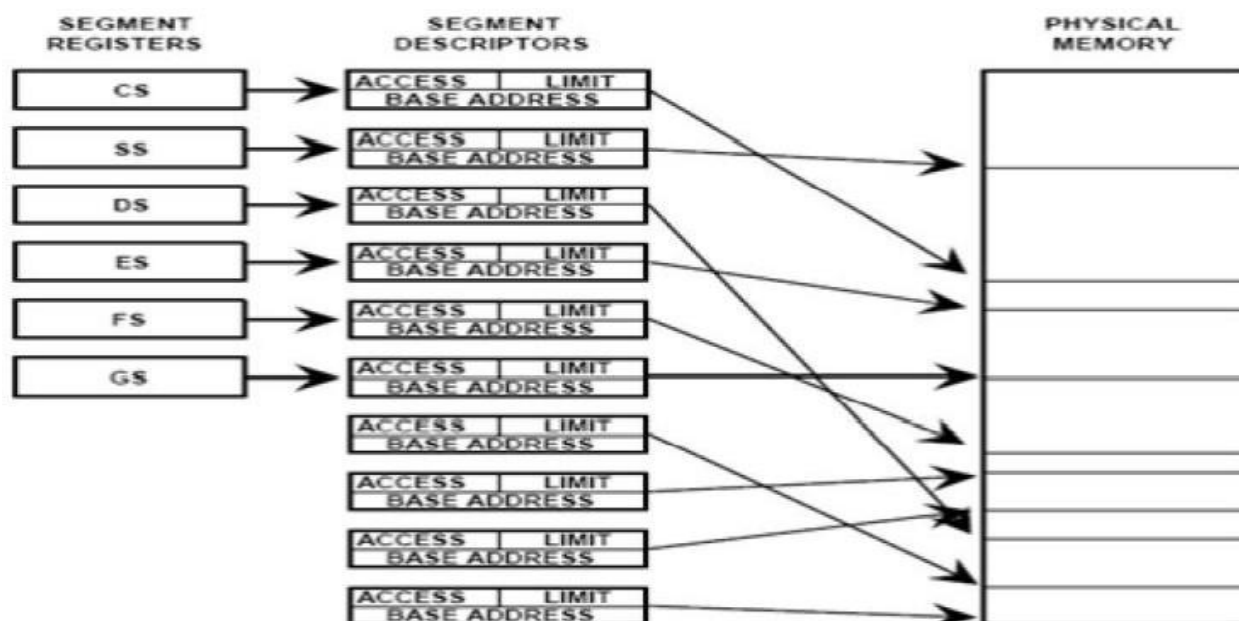
Model plat: Cel mai simplu model este modelul plat. În acest model, toate segmentele sunt mapate la întregul spațiu de adrese fizice. Un decalaj de segment se poate referi la zone de cod sau date. În cea mai mare măsură posibilă, acest model elimină mecanismul de segmentare din arhitectura văzută fie de proiectantul de sistem, fie de programatorul aplicației. Acest lucru se poate face pentru un mediu de programare precum UNIX, care acceptă paginarea, dar nu acceptă segmentarea.



Model plat protejat: modelul plat protejat este ca modelul plat, cu excepția faptului că limitele de segment sunt setate să includă doar intervalul de adrese pentru care există de fapt memoria. O excepție generală de protecție va fi generată la orice încercare de a accesa memoria neimplementată. Acesta poate fi folosit pentru sistemele în care mecanismul de paginare este dezactivat, deoarece oferă un nivel minim de protecție hardware împotriva unor tipuri de erori de program.



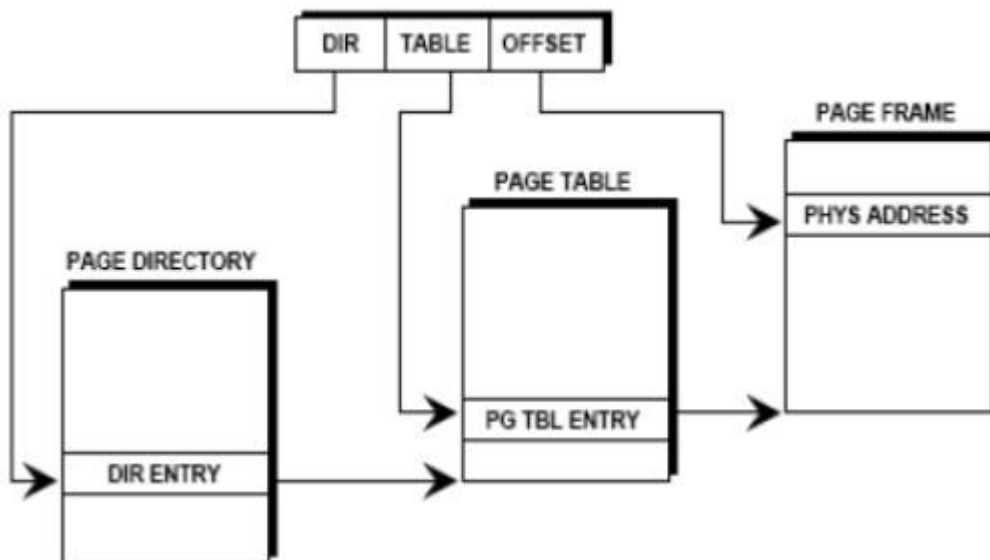
Modelul multisegment: Cel mai sofisticat model este modelul multisegment. Aici sunt utilizate toate capacitățile mecanismului de segmentare. Fiecare program are propriul său tabel de descriptori de segment și propriile sale segmente. Segmentele pot fi complet private pentru program sau pot fi partajate cu alte programe specifice. Accesul între programe și anumite segmente poate fi controlat individual.



27.Sa se prezinte modelul de adresare paginata pentru procesorul Intel Pentium.

Un **page table** este o matrice de intrări pe 32 de biți.

- Un tabel de pagini este el însuși o pagină și conține 4096 de octeți de date sau cel mult 1K intrări pe 32 de biți.
- Paginile de patru kiloocteți, inclusiv directoarele și tabelele de pagini, sunt aliniate la limitele de 4K octeți.
- Două niveluri de tabele sunt folosite pentru a adresa o pagină de memorie.
- La cel mai înalt nivel este un director de pagini.
- Un director de pagini conține până la 1K intrări care se adresează tabelelor de pagini de al doilea nivel.
- Un tabel de pagini de al doilea nivel se adresează până la 1K pagini din memoria fizică.



28.Sa se prezinte modul de lucru multitasking pentru procesorul Intel Pentium.

Suportul multitasking este o capacitate oferită de arhitectura Intel pe care designerul de sistem poate alege să nu o folosească. Multe dintre sistemele de operare multitasking disponibile care rulează pe arhitectura Intel nu folosesc suportul hardware multitasking furnizat. Unul dintre motivele cheie este că funcționalitatea comutatorului de sarcini hardware poate fi complet acceptată folosind comutarea sarcinilor software. Deoarece multe dintre sistemele de operare multitasking necesită, de asemenea, o procesare suplimentară la schimbarea sarcinilor, efectuarea lucrărilor suplimentare este puțin îngrijorătoare. De fapt, comutarea sarcinilor poate fi de fapt mai rapidă din cauza cantității de informații despre procesor care trebuie păstrată între comutatoarele de sarcini.

Funcționarea multitasking-ului hardware depinde de aceste **structuri suplimentare de date** ale sistemului:

- Segmentul de stare a sarcinii (TSS); segment pentru salvarea și restabilirea stării procesorului.
- TSS și trebuie să fie în GDT.
- Registrul sarcinilor (TR); ține selectorul pentru TSS-ul sarcinii curente.

Descriptorii TSS arată la fel ca un descriptor de tip de cod, cu excepția bitului D și a conținutului câmpului de tip. Un TSS este utilizat pentru a salva și pentru a încărca starea procesorului ori de câte ori este inițiată o schimbare de sarcină. TSS va păstra starea completă a procesorului plus harta de biți I/O.

O schimbare a sarcinii hardware are loc în următoarele cazuri:

- Sarcina curentă execută o instrucțiune JMP sau CALL
- O întrerupere (software sau hardware)
- Sarcina curentă execută o instrucțiune IRET atunci când este setat steag-ul Sarcină imbricată (NT).

Un comutator de activitate are acești pași:

- 1) **Verifica dacă sarcina curentă poate trece la noua sarcină.**
- 2) **Verifica dacă descriptorul TSS al noii sarcini este marcat prezent și are o limită validă.**
- 3) **Salvați starea sarcinii curente.**
- 4) **Încărcați registrul TR**
- 5) **Încărcați starea noii sarcini din TSS-ul său și continuați execuția.**

29. Sa se prezinte protectiile oferite de procesorul Intel Pentium.

Dacă oricare dintre aceste reguli este încălcată de o aplicație, va apărea o defecțiune generală de protecție .

1. Când un selector pentru un descriptor este încărcat într-un registru de segment.

Anumite registre de segmente pot conține doar anumite tipuri de descriptori; de exemplu:

- Numai registrul CS poate fi încărcat cu un selector pentru un segment executabil.

2. Anumite segmente pot fi folosite de instrucțiuni numai în anumite moduri predefinite; de exemplu:

- Nicio instrucțiune nu poate scrie într-un segment executabil [cod].
este setat.

Mecanisme de protecție utilizate:

1. Verificarea limitelor: Verificarea limită împiedică accesul unui program în afara limitelor unui segment. Verificarea limitelor este efectuată automat de procesor de fiecare dată când se face referire la memorie fie la un cod, fie la un segment de date.

2. Niveluri de privilegii: Arhitectura Intel oferă o modalitate de a controla ce tipuri de aplicații au acces; obiecte de sistem, segmente și memorie sau hardware mapat I/O.

3. Instrucțiuni protejate: Mecanismul de protecție împiedică executarea instrucțiunilor sensibile de sistem de la alte niveluri de privilegii decât cele de încredere de către sistemul de operare.

30. Sa se realizeze o comparație CISC-RISC.

Paralela RISC vs. CISC:

- **La RISC** - instructiunile trebuie sa fie in numar redus (<100), la CISC avem un set cat mai larg de instructiuni (>100).
- La RISC accesese de memorie sunt limitate la 2 instructiuni(load si store), la CISC accesese de memorie se fac prin tipuri variate de instructiuni, avand mai multe formate de instructiuni(>4)
- La RISC avem un numar mare de registre interne, pe cand la CISC avem un numar relativ mic de registre interne
- La RISC ~100MIPS, la CISC ~10 MIPS
- viteza de executie (performanta) RISC poate fi crescuta printr-o proiectare atenta a caii de date, benzii de asamblare si a celorlalte resurse ale CPU
- μsistemele RISC sunt mai orientate pe aplicatii si nu sunt utile in anumite medii de programare; ele au insa eficienta ridicata pe aplicatii
- compilatoarele optimizate pentru RISC solicita uzual mai mult timp de executie decat cele neoptimizate utilizate de CISC
- spre deosebire de CISC, in care multe registre interne sunt cu scop special, in RISC exista multe registre cu scop general
- spre deosebire de RISC, la CISC cele mai multe instructiuni se executa in mai mult de un ciclu masina

31.Sa se prezinte sintetic o caracteristica RISC (arhitectura setului de instructiuni; arhitectura benzii de asamblare regulate; unitățile de execuție).

I. Caracteristici:

- instructiunile trebuie sa fie in numar redus si sa se execute intr-un singur ceas
- accesele la memorie sunt limitate la 2 instructiuni, botezate load si store
- decodificarea instructiunilor este cablata si nu microprogramata
- jocul de instructiuni trebuie sa se bazeze pe un format fix (de regula 32 de biti) cu moduri de adresare simple
- fiecare operatie complexa trebuie sa fie rejectata la nivelul compilatorului
- ansamblul arhitecturii trebuie sa profite la maximum de pe urma modelului pipeline
- procesorul trebuie sa posede un numar mare de registre interne, cu sau fara tehnica de ferestre
- procesorul trebuie sa fie adaptat adaugarii sistemelor de memorii cache si coprocesoare
- setul de instructiuni este in general conceput pentru un domeniu de aplicatii perfect determinat

II. Categorii de instructiuni

Pentru orice procesor RISC vom regasi trei categorii de instructiuni indispensabile:

- 1) instructiuni aritmetice si logice
- 2) instructiuni de control secvential
- 3) instructiuni de acces la memorie

III. Formatele instructiunilor

Se adopta un format fix pentru ansamblul instructiunilor. Marimea este in general echivalenta marimii unui cuvant de memorie, adica in majoritatea cazurilor 32 biti. Pe baza acestei marimi fixe se determina numarul exact si compunerea formatelor instructiunilor.

IV. Un set de instructiuni orientat pe registre

O crestere sensibila de performanta poate fi obtinuta daca se limiteaza accesele la memoria externa, obiectiv atins crescand numarul de registre interne si limitand explicit accesele externe.

32.Sa se prezinte sintetic un processor RISC(VLIW).

Unitatea de instructiuni : Cuprinde coada de instructiuni, logica de iesire si BPU. Ea determina adresa urmatoarei instructiuni pentru a fi citita (fetch) utilizand fiecare din cele 3 componente si multe concepte fundamentale arhitecturii PowerPC: instruction prefetch, branch prediction, out-of-order operation si branch folding.

Unitatea de intregi primeste toate instructiunile cu intregi, load/store si anumite instructiuni in VM de la pozitia Q0 a IQ.

Unitatea in VM FPU primeste instructiunea de la IQ. Pentru a se asigura ca unitatea de instructiuni nu asteapta terminarea instructiunii in VM, FPU are propria sa coada de 2 instructiuni.

Unitatea de gestiune mem suporta nu numai translatarea adresei bazata pe segmentare si paginare dar si translatare orientate pe blocuri.

Unitatea Cache contine un cache de 32 K cu 8 cozi set - asociative unificat pentru cod si date si algoritmul de inlocuire LRU. Cache-ul furnizeaza 8 cuvinte pe magistrala interna pentru a putea fi incarcate in unitatea de instructiuni pana la 8 intrari in IQ.

Unitatea de memorare (MU) si interfata de sistem (SIU) Unitati functionale separate pentru interfatare cu accesele externe la memoria principala. MU consta din buffere de citire/scriere pe care procesorul le utilizeaza cand acceseaza memoria principala.

33.Sa se prezinte sintetic un sistem cu microprocesor RISC(VLIW).

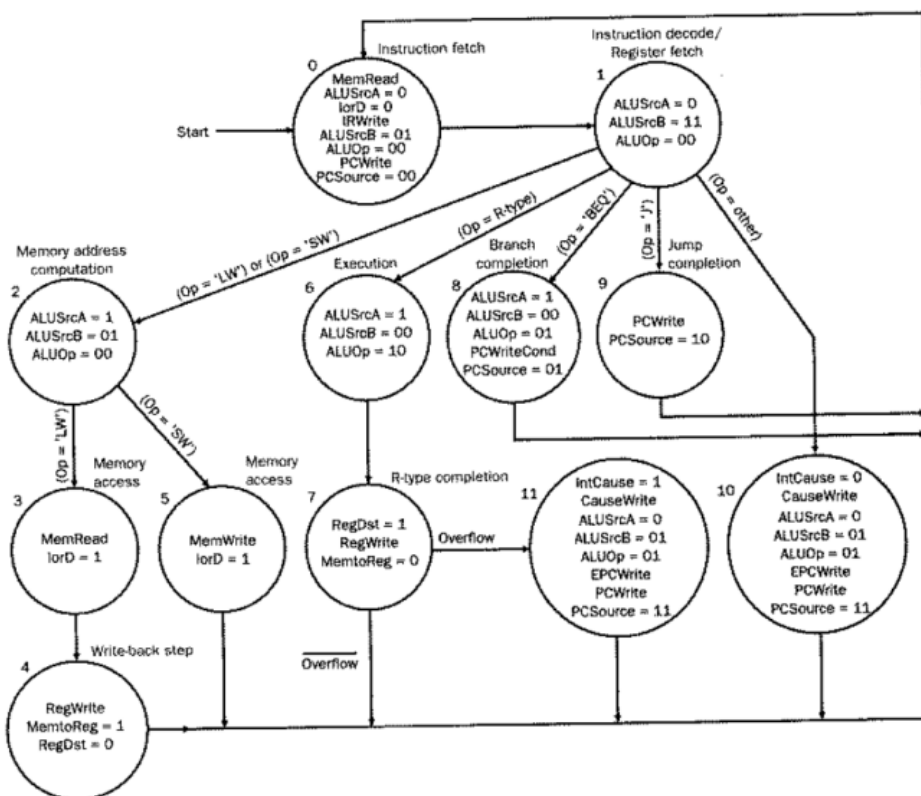
- are un sasiu tip 680x0 Mac ceea ce face posibil upgrade-ull prin schimbarea placii de baza
- au un pachet de emulare care poate rula atat DOS cat si Windows.
- o magistrala de date de 64b leaga procesorul 601 de ROM, RAM, slotul cache si sloturile de expandare.
- suporta un nivel 2 de cache de 128 KB – 1 MB.
- Subsistemul de memorie utilizeza un ciclu (pana la 64 b sau 8 octeti) sau patru cicli (32 octeti) burst posibil pentru 601.
- magistrala CPU este esantionata la jumatate din frecventa procesorului.
- au de regula 2 porturi seriale GeoPort cu 9 pini compatibili cu conectorii mini-DIN-8.
- oricare port poate fi configurat pentru modem/decodificator de sunet digital pe 16b pentru sunet I/O.
- Pentru un throghput mare de I/O se foloseste un I/O ASIC pentru conectare Ethernet serial si SCSI I/O;
- pentru serial I/O sunt furnizate doua buffere de 8b FIFO. I/O ASIC implementeaza controller SCSI-2, maxim 5 MB/sec. al doilea chip controller SCSI-2, NCR 53CF96, conectat la o magistrala SCSI interna de 50 pini.
- Utilizand un SCSI-2 rapid, magistrala interna poate functiona la o rata de 10MB/sec. conector intern CDROM.

34.Sa se clasifice semnalele unui microprocesor RISC(VLIW)pe categorii functionale.

Semnalele pentru un microprocessor MPC7451 MPX sunt grupate astfel:

- *Arbitrarea adreselor* – folosite la arbitrarea conducerii magistralei de adrese
- *Start transfer adrese* – indica faptul ca o magistrala master a inceput o tranzactie pe bus-ul de adrese
- *Transfer de adresa* – includ semnalele pentru magistrala de adrese si pe cele pentru paritatea adresei
- *Atributele transferului* – ofera informatii despre tipul transferului, dimensiunea lui
- *Terminare transfer adresa* – folosita pentru confirmarea incheierii unui transfer si necesitatii sau nu de repetare a acestuia
- *Arbitrare date* – folosite la arbitrarea conducerii magistralei de date
- *Transfer de date* – folosite pentru a transfera date si pentru a asigura integritatea transferului
- *Terminare transfer date* – confirma incheierea transferului de date si necesitatea sau nu de repetare a acestuia

35.Sa se prezinte organigrama de functionare a unui procesor RISC (VLIW) cu specificarea semnalelor de tranzitie intre stari.



36.Sa se prezinte un ciclu de magistrala reprezentativ pentru un procesor RISC (VLIW).

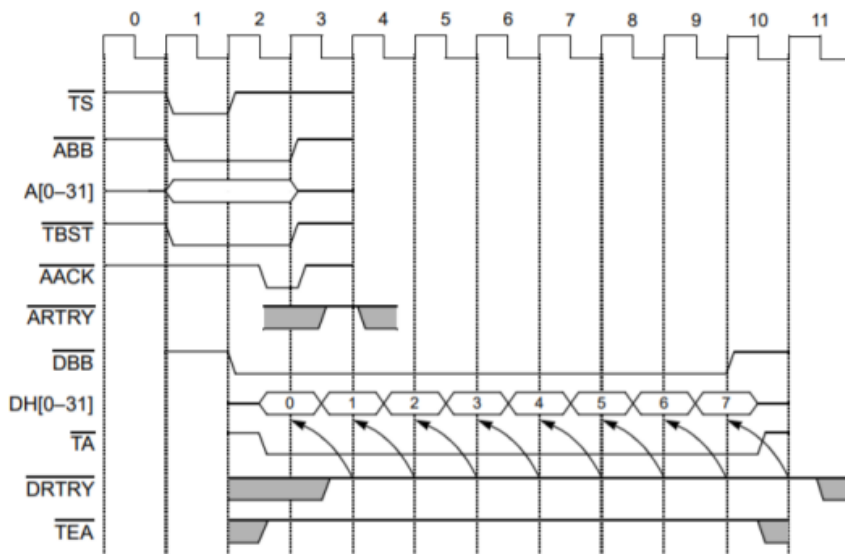


Figure 6-2. 32-Bit Data Bus Transfer (Eight-Beat Burst)

37.Sa se prezinte si sa se caracterizeze modurile de lucru ale unui processor RISC.

PowerPC au 2 nivele de privilegiu:

- a) modul supvizor: utilizat exclusiv de sistemul de operare.
- b) modul user: utilizat de catre aplicatiile software si SO.

Exista 3 nivele sau medii de programare:

a) **PowerPC User Instruction Set Architecture (UIA)**: defineste setul de instructiuni de baza la nivel user, registrele la nivel user, tipuri de date, conventii de memorie, modele de memorare si programare vizate de aplicatiile program.

b) **PowerPC Virtual Environment Architecture (VEA)**: este cea mai mica componenta a arhitecturii PowerPC ce defineste functionalitati aditionale la nivel user care cad de regula in afara cerintelor de software user; VEA descrie modelul de memorare pentru un mediu in care multiple procesoare sau alte dispozitive pot accesa memoria externa si defineste aspecte ale modelului cache si instructiunile de control cache din perspectiva userului.

c) **PowerPC Operating Environment Architecture (OEA)**: defineste resursele la nivel supvizor tipic cerute de un SO; OEA defineste modelul de gestiune a memoriei PowerPC, registrele de nivel supvizor, cerinte de sincronizare si modelul de exceptii. Implementarile OEA adera si la UIA si VEA. Tranzitia de la UIA sau VEA la OEA se fac via mecanisme de exceptii.

38.Sa se prezinte resursele unui procesor RISC (VLIW) din punct de vedere al programatorului.

3 seturi de registre:

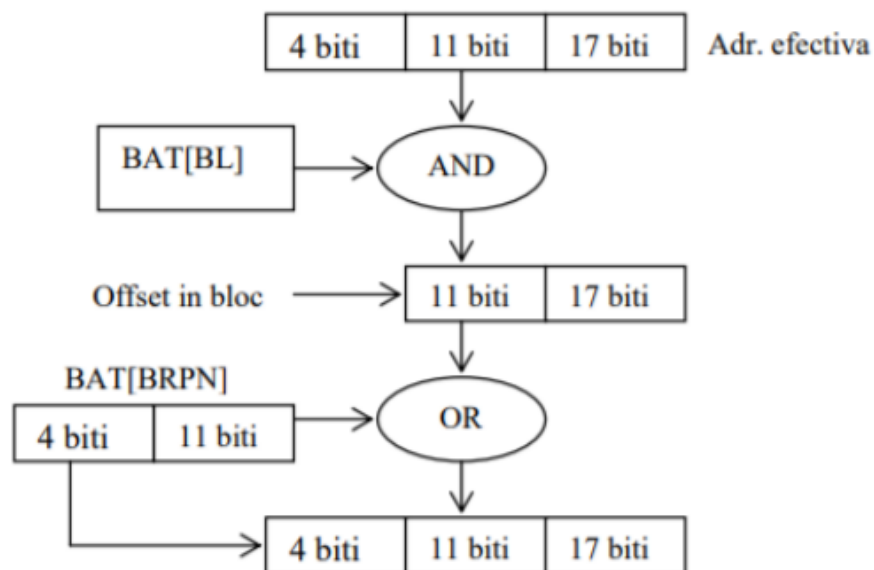
- **Registre UISA:** pot fi accesate de intregul software in ambele nivele de privilegiu; includ 32 GPR, 32 FPR, alte registre utilizate pentru intregi, VM si instructiuni de ramificatie.
- **Registre VEA:** definesc portiunea user a facilitatii bazei de timp (TB) care consta din 2 registre TB pe 32 biti care pot fi citite la nivel user dar pot fi scrise doar la nivel supervizor.
- **Registre OEA:** sunt utilizate pentru operatii la nivel sistem (supervizor).

Registre UISA si VEA: 32b sau 64b

- **Registre cu scop general:** 32 de registre (ri) utilizate de unitatea de intregi pentru a efectua operatii aritmetice
- **Registre in VM :** 32 de registre (fri) utilizate in unitatile de VM
- **Registrul de conditii :** similar cu registrul EFLAGS de la x86. Are 8 campuri de 4 biti: CR0:CR7.
- **Registrul de control si stare VM):** inregistreaza, activeaza/dezactiveaza exceptiile generate de operatiile in VM, controleaza modul de rotunjire.
- **Registrul de Exceptii Intregi :** inregistreaza, activeaza/dezactiveaza exceptiile generate de operatiile intregi
- **Registrul Legaturi :** furnizeaza adresa de ramificatie pt utilizare cu instructiunea bclr
- **Registrul Contor :** registru de contor de bucle
- **Registrul Baza de Timp :** singurul definit pentru VEA si poate fi accesat in ambele moduri (user, supervizor)

39.Sa se prezinte un mecanism de translatare a adreselor pentru un procesor RISC (VLIW)

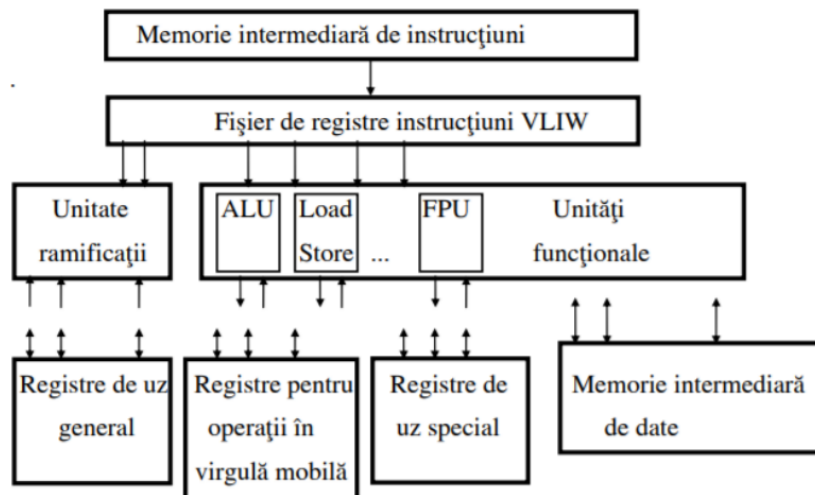
Maparea BAT este activata numai cand translatarea adresei este activata in MSR(MSR[IR]=1 si/sau MSR[DR]=1). Daca o regiune particulara de mem. este mapata utilizand BAT si segmentare/paginare, BAT are prioritate. In translatarea BAT, fiecare bloc de mem. este definit printr-o pereche de reg. BAT(BATU, BATL). Cand procesorul recunoaste o adr. efectiva ca fiind mapata la o pereche de reg. BAT apare un BAT Hit. Procesul de translatare BAT pentru PowerPC pe 32 biti se desfasoara astfel:



Bitii de validare acces din BAT realizeaza protectia memoriei.

40.Sa se prezinte din punct de vedere funcțional arhitectura VLIW.

O arhitectura generala VLIW ar fi:



- Procesoarelor VLIW pot implementa vechile seturi de instrucțiuni CISC pentru ca programarea acestora este foarte asemănătoare cu scrierea de microcod, putându-se conserva astfel programul.
- Caile de date constau într-un număr de unități funcționale, conectate prin registre globale și speciale multiport, prin intermediul unei rețele de comutare de tip "crossbar" la o memorie intermediară de date
- Fiecare unitate funcțională este controlată de un set independent de câmpuri ale instrucțiunii și de registre specifice de adresă pentru operanți și rezultate
- Există o unitate de salt controlată tot de instrucțiune
- Un procesor VLIW poate avea unul sau mai multe fișiere de registre globale
- Un procesor VLIW poate avea unități omogene sau heterogene
- Compilatorul este mai complex decât la sistemele RISC, motiv pentru care se poate realiza o imbinare "hardware-software" a implementării VLIW
- Datorită lungimii foarte mari a cuvântului de instrucțiune și pentru a elimina creșterea complexității "hardware"-ului, acest compilator preia instrucțiunile și le pune într-un cuvânt de instrucțiune foarte lung din care ele să poată fi executate în paralel și folosind benzi de asamblare multiple
- Compilatorul este cel care se ocupă de planificarea ramificațiilor prin intermediul blocurilor de cod folosite la generarea instrucțiunii foarte lungi
- Un "hardware" adecvat pe procesorul VLIW poate oferi un anumit suport pentru compilator (ex. operația de salt cu cai multiple)
- Deoarece compilatorul trebuie să cunoască microarhitectura procesorului VLIW, compatibilitatea cu noile generații poate fi menținută numai prin folosirea unor tehnici hardware sau software adecvate

41.Sa se realizeze o comparatie intre modelul masinii de stare al arhitecturii VLIW si modelele masinii de stare pentru alte arhitecturi de procesare (SISD; SIMD; MIMD).

Modelul traditional SIMD este o simplificare a modelului VLIW. Astfel, iesirea unei functii singulare m va fi distribuita la fiecare unitate functionala. Daca functiile m_1, \dots, m_n ale unui model VLIW sunt identice si egale cu functia m a unui model SIMD, atunci cele doua masini au o functionare echivalenta; aceasta implica faptul ca se poate programa un procesor VLIW sa emuleze functionarea unui procesor SIMD. De aceea se poate considera masina VLIW ca un superset functional al masinii SIMD.

42.Sa se enumere si sa se caracterizeze succint tehnicile de crestere a paralelismului pentru arhitecturile VLIW.

Tehnici hardware de crestere a paralelismului:

- Utilizarea unui hardware care **sa monitorizeze fluxul de instructiuni si sa grupeze** mai multe instructiuni din flux intr-o instructiune VLIW prin procesorul insusi
- **Fill Unit** – FU - Un hardware pentru compactarea microoperatiilor generate prin citirea instructiunilor intr-un tampon de instructiuni decodificate

Tehnici software de creste a paralelismului:

- **Software pipelining** - distributia unei bucle pe mai multe procesoare sau unitati functionale prin atribuirea de stari independente la procesoare/unitati functionale si apoi inlanturea continua a operatiilor
- **Desfasurarea buclelor** - pentru bucle fara dependente de date intre iteratii, se expandeaza iteratiile unei bucle de multiple ori si se planifica instructiunile din bucla desfasurata crescand distanta lexicala intre instructiuni cu dependente.

43.Sa se prezinte succint modalitatile de pastrare a compatibilitatii arhitecturale VLIW.

Tehnici hardware

Aceste tehnici folosesc o planificare dinamica hardware a codului. Un model de executie este urmatorul:

a) Metoda split-issue

Tehnica de planificare dinamica hardware prin divizarea executiei (splitissue), elaborata de B. Rau, se caracterizeaza prin urmatoarele: Fiecare operatie este divizata într-o pereche de operatii de tipul read-and-execute (RE), destination-writeback (DW).

b) Metoda fill-unit

- Procesorul executa un flux de operatii UAL sau NUAL.
- Concurent cu executia, unitatile de umplere compacteaza operatiile într-o multioperatie, care se memoreaza într-o linie de memorie intermediara ascunsa (shadow instruction cache).
- Formarea unei noi multioperatii de catre unitatea de umplere este terminata când se înregistrează o instructiune de ramificatie

Tehnici software

a) Recompilarea statică

Aceasta tehnica recompileaza întregul program în mod static (offline) si poate lua avantajele compilatoarelor optimizate. Recompilarea completa poate fi evitata prin mentinerea copiilor multiple de programe pentru diferitele arhitecturi destinatie într-un fisier obiect partitionat; un modul adecvat acestei cerinte poate fi planificat la instalare.

b) Replanificarea dinamică

Tehnica de replanificare dinamica recurge la o versiune limitata de planificare prin program (software scheduling) fara a necesita resurse hardware suplimentare.

Incarcatorul (loader-ul) sistemului de operare citeste antetul (header-ul) programului binar si detecteaza versiunea generatiei arhitecturii. Dupa ce prima pagina a programului este încarcata pentru executie, agentul de gestiune a erorii de pagina invoca modulul de replanificare dinamica, care replanifica executia pe masina curenta; procesul este repetat pentru fiecare noua eroare de pagina.

Paginile translate sunt salvate pe un spatiu de înlocuire (swap); sunt replanificate numai paginile care sunt executate în timpul duratei de viata a programului.

44.Sa se prezinte sintetic modelul architectural VLIW pentru un procesor de semnal.

Un procesor de semnale video poate fi proiectat continuând mai multe elemente de procesare (PE- Processing Elements) lucrând în paralel. Semnalul de prelucrat este digitalizat si introdus la procesare prin intermediul unui comutator de tip crossbar (cu bare încrucisate). Elementele de procesare executa operatiile în paralel pe baza unor grafuri de flux de semnal SFG (Signal Flow Graph).

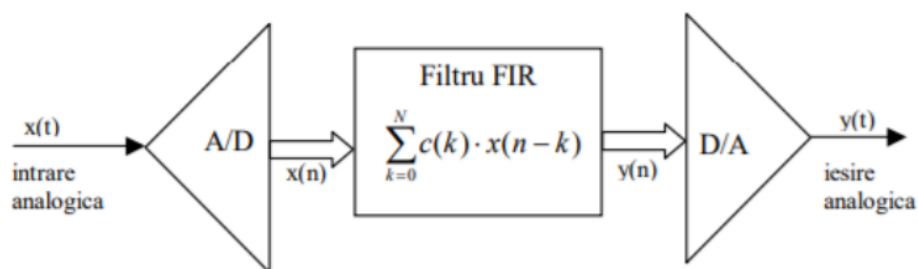
Cele 10 PE corespund urmatoarelor unitati:

- 3 elemente logice aritmetice (ALE – “Arithmetic Logic Element) în banda de asamblare;
- 2 elemente de memorare (ME – “Memory Element);
- 5 elemente de iesire.

Instructiunile pentru ALE au 56 biti, cele pentru ME au 36 biti iar cele pentru elementele de iesire au 12 biti. O instructiune VLIW va avea: $3 \times 56 + 2 \times 36 + 5 \times 12 = 300$ biti.

45.Sa se prezinte, pe baza schemei echivalente, funcțiile si aplicațiile unui procesor de semnal.

Schema echivalenta cu DSP este:



Funcții DSP:

a) filtrare digitala

- FIR (Finite Impulse Response);
- IIR (Infinite Impulse Response);

b) procesare de semnal

c) procesare de date

- criptari;
- codari;
- decodari;

d) procesare numerica

e) **modulari**

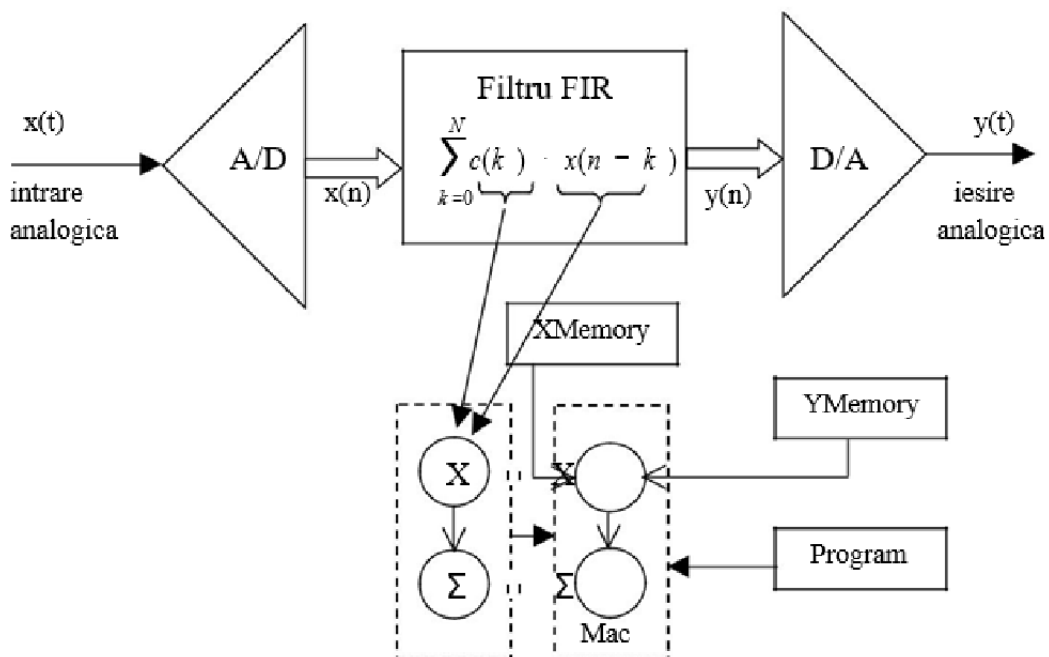
- in amplitudine;
- in frecventa;
- in faza;

DSP sunt folosite in aplicatii ca:

- a) telecomunicatie: generare ton, speaker-full-duplex, teleconferinte, posta vocala, transcodare, transceiver-e ISDN, securitate telefonica
- b) comunicatii de date: modemuri de inalta viteza si frecventa pe biti multipla, facsimile de inalta viteza
- c) radiocomunicatie: comunicatii sigure, punct la punct sau radio-emisie, telefonie mobila celulara
- d) calculatoare: procesoare, statii de lucru, calculatoare personale, acceleratoare grafice

46.Sa se prezinte, pe baza schemei de structura,caracteristicile procesoarelor de semnal Motorola DSP56K (DSP96002).

Structura unui DSP este:



Caracteristicile DSP56K:

- a) **Viteza:** destul de ridicata;
- b) **Precizia:** lucru pe 24 biti cu rezultat intermediar pe 56 biti;
- c) **Paralelism:** permite ca o sectiune de filtru IIR cu coeficient 4 sa poata fi executata in numai 4 cicli. In acelasi timp doua controllere seriale pot trimite si primi full-duplex date si portul gazda poate trimite si primi date simplex

d) Flexibilitate: interfete seriale si paralele on-chip care pot suporta configuratii variate de memorie si module periferice;

e) Depanare: tehnologia Motorola de emulare on-chip(OnCE - On-Chip Emulation) permite acces simplu, ieftin si rapid la registrele interne pentru depanare;

f) Ceas bazat pe bucla blocata in faza (PLL – Phase-Locked Loop): permite utilizarea aproape a oricarui sistem de ceas extern disponibil si ceas extern sincronizat intern;

g) Banda de asamblare invizibila: banda de asamblare pe trei etaje este invizibila programatorului, permitand programe de dezvoltare in limbaj de asamblare sau nivel inalt (ex: C);

h) Setul de instructiuni: asemanator unitatilor centrale Motorola: sintaxa ortogonala controleaza unitatile de executie paralela;

i) Compatibilitati intre membrii familiei;

j) Consum scazut: poseda instructiuni de trecere in mod de lucru cu consum scazut

47.Sa se prezinte sintetic un procesor de semnal (DSP56000; DSP96002; TMS320C50x(60x); ...) sau un procesor embedded..

DSP96002

- **Nucleu de procesare semnale digitale (DSP core)**

- motor DSP eficient pe 32 biti
- respecta standardul IEEE 754-1985 pentru operatiile aritmetice cu 'single precision' (32-bit) si 'single extended precision' (44-bit)
- ~30 milioane de instructiuni pe secunda (MIPS) la frecventa 60 MHz
- operatii executate in paralel la nivelul Data ALU, Address Generation Unit (AGU), iar Controlerul de program din CPU permite mai multă procesare per ciclu de instructiuni
- Multiplicator paralel cu un singur ciclu de 32×32 de biti
- set de instructiuni bine paralelizat cu moduri unice de adresare a DSP-ului
- bucle hardware inlantuite DO(Nested hardware DO loops)
- Instruction cache extended to operate as 4 K byte (1 K word)
- intreruperi auto-return rapide
- Bus de adrese:
 - un bus de adrese intern, unidirectional de 32 biti de tip X(XAB)
 - un bus de adrese intern, unidirectional de 32 biti de tip Y(YAB)
 - un bus de program intern de 32 biti(PAB)
 - 2 bus-uri de adrese externe
- Bus de date:
 - un data bus intern, bidirectional de 32-bit X(XDB)
 - un data bus intern, bidirectional de 32-bit Y(YDB)
 - un Global memory Data Bus de 32 biti bidirectional(GDB)
 - un DMA Data Bus 32 biti intern, bidirectional(DDB)
 - un Program Data Bus 32 bit bidirectional, intern (PDB)
 - 2 bus-uri de date externe de 32 biti
- MCU setul de instructiuni mnemonice este asemanator cu MCU, ceea ce face progarmarea mai usoara

- **Memorie**

- de program on-chip 1024×32 -bit RAM
- 2 X mem. de date independente on-chip 512×32 -bit RAM
- 2 X mem. de date independente on-chip 512×32 -bit ROMs (1024×32 -bit mem. virtuala)
- bootstrap on-chip 64×32 -bit ROM

48.Sa se prezinte sintetic un sistem cu procesor de semnal sau cu procesor embedded.

Power Macintosh :

- are un sasiu tip 680x0 Mac.
- au un pachet de emulare care poate rula atat DOS cat si Windows.
- au 4 MB de ROM la 100 nsec motati pe un socket SIMM.
- o magistrala de date de 64b leaga procesorul 601 de ROM, RAM, slotul cache si sloturile de expandare.
- suporta un nivel 2 de cache de 128 KB – 1 MB.
- Subsistemul de memorie utilizeza un ciclu (64 b/8 octeti) sau patru cicli (32 octeti) burst posibil pentru 601.
- magistrala CPU este esantionata la jumatate din frecventa procesorului.
- Canalele DMA nu pot fi utilizate in acelasi timp.
- Pentru un throghput mare de I/O se foloseste un I/O ASIC pentru conectare Ethernet serial si SCSI I/O;
- pentru serial I/O sunt furnizate doua buffere de 8b FIFO. I/O ASIC implementeaza controller SCSI-2, maxim 5 MB/sec. al doilea chip controller SCSI-2, NCR 53CF96, conectat la o magistrala SCSI interna de 50 pini.
- Utilizand un SCSI-2 rapid, magistrala interna poate functiona la o rata de 10MB/sec. conector intern CDROM.

49.Sa se clasifice semnalele unui procesor de semnal sau un procesor embedded pe categorii functionale.

Semnalele pentru un microprocessor MPC7451 MPX sunt grupate astfel:

- **Arbitrarea adreselor** – folosite la arbitrarea conducerii magistralei de adrese
- **Start transfer adrese** – indica faptul ca o magistrala master a inceput o tranzactie pe bus-ul de adrese
- **Transfer de adresa** – includ semnalele pentru magistrala de adrese si pe cele pentru paritatea adresei
- **Atributele transferului** – ofera informatii despre tipul transferului, dimensiunea lui
- **Terminare transfer adresa** – folosita pentru confirmarea incheierii unui transfer si necesitatii sau nu de repetare a acestuia
- **Arbitrare date** – folosite la arbitrarea conducerii magistralei de date
- **Transfer de date** – folosite pentru a transfera date si pentru a asigura integritatea transferului
- **Terminare transfer date** – confirma incheierea transferului de date si necesitatea sau nu de repetare a acestuia

50. Sa se prezinte porturile de interfata ale unui procesor de semnal sau ale unui procesor embedded.

- **Address (A0–A15)** - Acesti pini three-state specifica adresele bus pentru accesul la programe externe și memorie de date. Pentru a minimiza disiparea de putere, A0-A15 nu isi modifica starea atunci cand spatii de memorie externe sunt accesate.

- **Data (D0–D23)** - Acesti pini furnizeaza data bus-ul bidirectional pentru accesul la programe externe și memorie de date. D0-D23 sunt in stare de mare impedanta atunci cand pe bus circula date.

- **Port A Bus Control Signals** - Aceste semnale ofera posibilitatea de conectare la bus masters aditionali (DSP-uri, microprocesoare, controller DMA etc.) la port A. Ele sunt three-stated in timpul resetului si ar putea avea nevoie de rezistente de pullup pentru a preveni operatiile eronate.

51. Sa se prezinte resursele unui procesor de semnal sau embedded din punct de vedere al programatorului.

- **Registre generale de date:** X0, X1, Y0, Y1 pe 24 de biti (4 intrari) sau 2 intrari de 48 de biti X si Y

- **Registre acumulator:** A si B pe 48 de biti cu extensie de 8 biti pentru lucrul cu 56b

52.Ciclul de viata al unui sistem incorporat.

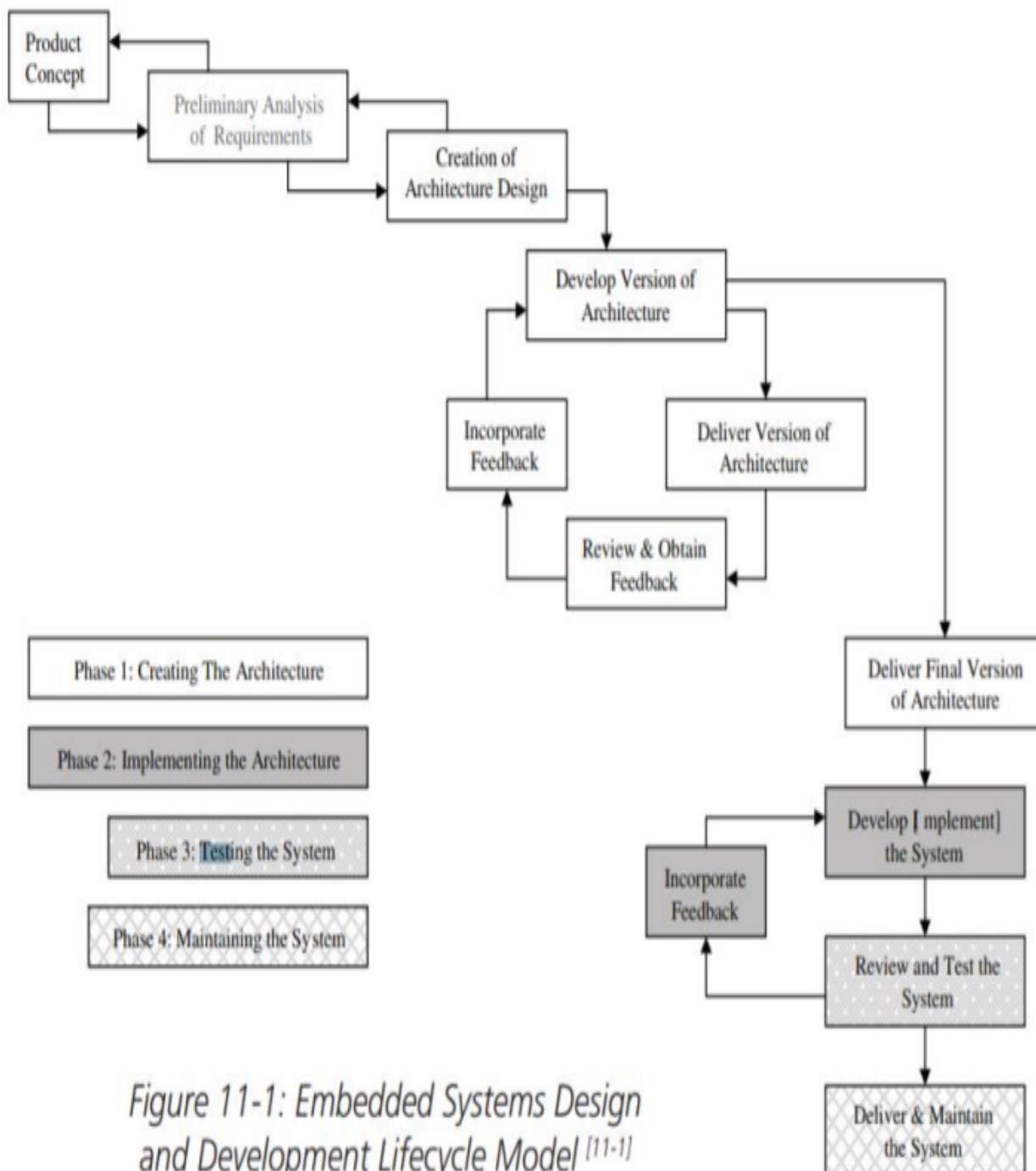


Figure 11-1: Embedded Systems Design and Development Lifecycle Model ^[11-1]

53.Enumerati si caracterizați fazele de proiectare ale unui sistem incorporat.

Modelul (poza de la ex 52) indica faptul ca procesul de proiectare a unui sistem embedded are 4 etape importante:

1. Crearea arhitecturii, adica procesul de planificare a proiectarii sistemului
2. Implementarea arhitecturii = procesul de dezvoltare a sistemului embedded
3. Testarea sistemului = procesul de testare a sistemului embedded cu scopul de a descoperi probleme, care vor fi rezolvate ulterior
4. Mentenanta sistemului = procesul de lansare a sistemului pe piata si asigurarea asistentei tehnice pentru utilizatorii sistemului pe toata durata de functionare a acestuia

54.Prezentati succint faza de creare a unui sistem incorporat.

Aceasta faza consta in 6 stagii, avand proprietatea ca fiecare dintre acestea se bazeaza pe rezultatul obtinut in cadrul stagiului precedent. Acestea sunt:

1. Detinerea unei baze tehnice solide
2. Intelegerea ciclului de afaceri al arhitecturii (Arhitectural Bussiness Cycle - ABC) Ciclul de afaceri al arhitecturii contine toate tipurile de influente care genereaza cerintele sistemului.
 - 2.1 Intelegerea faptului ca ABC influenteaza cerintele sistemului embedded si ca aceste influente pot fi si non-tehnice
 - 2.2 Identificarea tuturor ABC-urilor care influenteaza design-ul
 - 2.3 Implicarea diferitelor influente cât mai curând posibil în ciclul de viață al proiectării și dezvoltării și colectarea cerințelor sistemului
 - 2.4 Determinarea tuturor elementelor software si hardware necesare pentru indeplinirea cerintelor
3. Definirea unor tipare arhitecturale si a modelelor de referinta. Un tipar arhitectural pentru un anumit device reprezinta un profil high-level pentru sistemul embedded. Acest profil contine o descriere a elementelor software si hardware pe care device-ul le poate contine, functiile acestora in cadrul sistemului, un layout topologic al elementelor (numit si model de referinta) si interactiunile dintre interfetele externe si diferite elemente.
4. Crearea unei structuri arhitecturale
5. Documentarea arhitecturii
 - 5.1 Un document care sa reprezinte outline-ul pentru intreaga arhitectura
 - 5.2 Document pentru fiecare structura din arhitectura
 - 5.3 Glosar al arhitecturii
6. Analizarea si evaluarea arhitecturii

- statica
 - sistemul nu ruleaza
 - testeaza specificatiile produsului
- dinamica
 - sistemul ruleaza
 - testeaza date despre utilizator, situatii limita, limite interne, input-uri

57. Prezentati succint faza de intretinere a unui sistem incorporat.

Dupa lansarea unui dispozitiv incorporat exista unele responsabilitati care trebuie indeplinite, precum instruire, suport tehnic, punerea la dispozitie a update-urilor, solutii pentru bug-uri, etc. Documentatia arhitecturala, poate fi folosita ca o baza pentru manualele tehnice, de utilizare si de instruire sau pentru a determina impactul unui update asupra sistemului. Pentru a asigura succesul design-ului unui sistem incorporat este important sa fim familiari cu etapele parcurse pentru a ajunge la produsul finit, in special cu arhitectura. Altfel spus, inginerii hardware trebuie sa inteleaga soft-ul, iar cei software trebuie sa inteleaga hardware-ul. De asemenea, este important ca toti dezvoltatorii sa se puna de accord asupra unei metodologii de implementare si de testare a sistemului pe care sa o respecte.