* 1. **Princípy a základné koncepcie počítačových systémov. Definícia číslicového počítača. Koncepcia programového riadenia číslicového počítača: von Neumannova architektúra číslicového počítača. Výpočtové modely (CF,DF,DD). Flynnova klasifikácia (+ obr.)**

**Číslicový počítač** (ČP) je zložitý univerzálny číslicový systém (automat) určený na samočinné vykonávanie postupnosti operácií (výpočtov) nad údajmi zobrazenými číslicovým kódom, na základe vopred pripraveného a v pamäti uloženého programu (algoritmu).

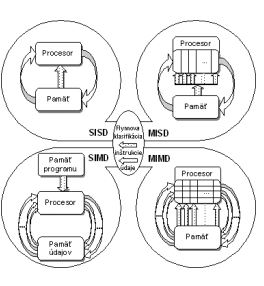
**VN:**

1. *Pamäť* je použitá na uloženie *inštrukcií*aj*údajov*.
2. *Riadiaca jednotka* j e použitá na výber inštrukcií z pamäte.
3. *Aritmetická jednotka* je použitá na vykonávanie špecifikovaných operácií nad údajmi.
4. *Vstupná jednotka* je použitá na vstup údajov, *výstupná jednotka*na výstup údajov.

**Výpočtové modely** vo všeobecnosti slúžia pre popísanie systému respektíve problému. ⁄ Spôsob akým sú prevádzkované celé aplikácie a ich časti. ⁄ Spôsob akým s nimi komunikujú užívatelia respektíve iné aplikácie. ⁄ spôsob akým sa ukladajú, posielajú a využívajú dáta

* **CF (Control Flow)**, pri ktorom sa riadenie výpočtového procesu uskutočňuje prostredníctvom interpretácie sériového prúdu inštrukcií programu.
* **DF (Data Flow**), pri ktorom sa riadenie výpočtového procesu uskutočňuje prostredníctvom prúdu operandov (údajov), definujúcich pripravenosť inštrukcií na vykonanie.
* **DD (Demand Driven**), pri ktorom sa riadenie výpočtového procesu uskutočňuje na základe požiadaviek inštrukcií programu na vyslanie operandov. Výpočtové modely založené na báze organizácie výpočtu typu DD sa nazývajú redukčné modely.

**Flynn-ova klasifikácia**



* SISD ν Single Instruction Single Data:
  + Jedna riadiaca jednotka (RJ)
  + Jedna vykonávacia jednotka (VJ)
  + Čítanie/zápis do pamäte len jednej hodnoty
  + Von Neumann architektúra
* MISD ν Multiple Instruction Single Data
  + Jedna RJ, jedna inštrukčná cache pamäť ⁄ Viac (identických) synchronizovaných VJ ⁄ Rovnaké operácie nad rôznymi dátami ⁄ „Von Neumann front-end computer + processor array“ ⁄ Vektorové procesory
* SIMD ν Single Instruction Multiple Data
  + Rôzne operácie nad rovnakými dátami ν Single data flow ν Micro-pipeline
  + Napr. Carnagie-Mellon, systolické polia, data flow architektúry
* MIMD ν Multiple Instruction Multiple Data
  + Procesory môžu pracovať asynchrónne (na rozdiel od MISD) ⁄ Zdieľaná pamäť na výmenu dát ⁄ Napr. multiprocessory (SM-MIMD) a multipočítače (DM-MIMD)
  1. **Princípy a základné koncepcie počítačových systémov. Definícia číslicového počítača. Klasifikácia číslicových počítačov (všeobecne). Kuck-ova klasifikácia. Shore-ova klasifikácia (+ obr.). Fengova klasifikácia. Erlangenova klasifikácia (+obr.).**

**Číslicový počítač** (ČP) je zložitý univerzálny číslicový systém (automat) určený na samočinné vykonávanie postupnosti operácií (výpočtov) nad údajmi zobrazenými číslicovým kódom, na základe vopred pripraveného a v pamäti uloženého programu (algoritmu).

**Klasifikácia číslicových počítačov**

* Flynnova klasifikácia (1966)
* Používateľsko-aplikačná klasifikácia
* Klasifikácia podľa použitého výpočtového modelu ν Programové riadenie (CF – control flow) ν Riadenie tokom dát (DF – data flow) ν Podľa požiadavky (DD – data demand)

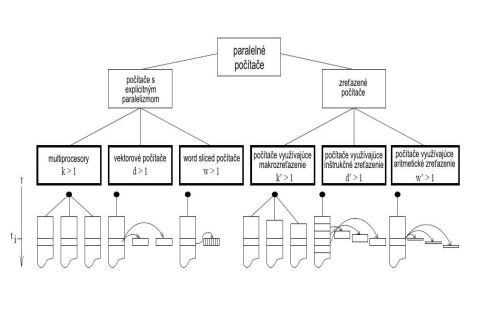
**Kuck-ova klasifikácia**

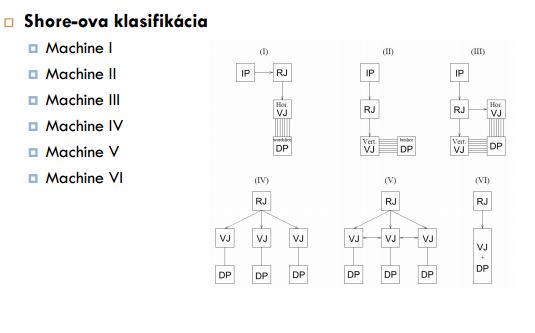
* SISSES ν Single Instruction Scalar, Single Execution Scalar
* SISSEA ν Single Instruction Scalar, Single Execution Array
* SIASEA ν Single Instruction Array, Single Execution Array
* MISMES ν Multiple Instruction Scalar, Multiple Execution Scalar
* MISMEA ν Multiple Instruction Scalar, Multiple Execution Array

**Fengova klasifikácia**

* Systémy popisuje pomocou dvojice čísel. Prvé číslo reprezentuje veľkosť slova a druhé počet slov, ktoré je možné spracovať v jednom čase.
* WSBS – word-serial, bit-serial ( bit-serial ) ⁄ WSBP – word-serial, bit-parallel (word-slice ) ⁄ WPBS – word-parallel, bit-serial ( bit-slice ) ⁄ WPBP – word-parallel, bit-parallel ( fully paralel )

**Erlangen-ova klasifikácia** Systém je popísaný trojicou ( k\*k’, d\*d’, w\*w’ ) kde **k** – počet riadiacich jednotiek **d** – počet výpočtových jednotiek na jednu riadiacu jednotku **w** – veľkosť jedného dátového slova **k’** - počet špecializovaných riadiacich jednotiek **d’** - počet špecializovaných výpočtových jednotiek **w’** - počet zreťazených stupňov



****

**2.1 Stavebné prvky číslicových počítačov. Návrh PS a jeho komponentov. Špecifikácia ISA. Špecifikácia IA. Špecifikácia fyzickej realizácie. Definícia číslicového systému na abstraktnej úrovni. Dekompozícia ČS na operačnú a riadiacu časť (+obr.). Príklad zobrazenia IA na systémovej úrovni (+obr.). Príklad zobrazenia IA na úrovni prenosov medzi registrami (+obr.).**

**Návrh PS a jeho komponentov**

* Systémová úroveň (SÚ)
* Úroveň prenosov informácií medzi registrami (PÚ)
* Úroveň logických obvodov (LÚ)
* Úroveň technologickej realizácie (TÚ)

**Špecifikácia inštrukčno orientovanej architektúry (oblasť správania sa)**

* **SÚ**: inštrukcie, údaje, programy, algoritmy, operačný systém, jazyky HLL, **PÚ**: operácie, údajové typy, aplikačné programy, používateľské programy, **LÚ**: boolovské výrazy, konečné automaty, **TÚ**: funkcie spojitých obvodov, prenosové funkcie

**Špecifikácia implementačne orientovanej architektúry (oblasť štruktúry)**

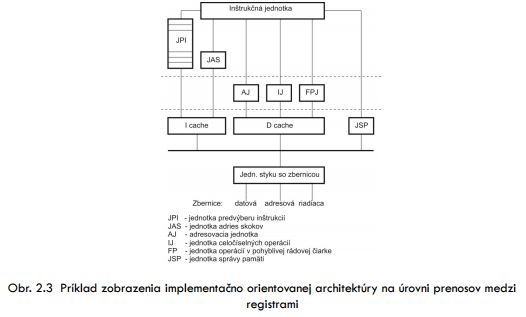
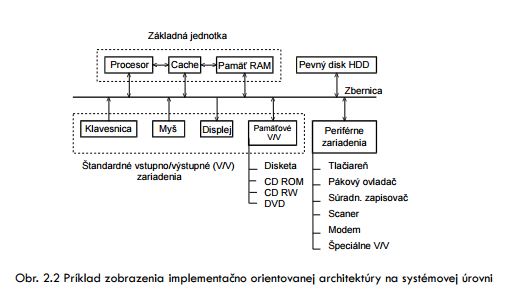
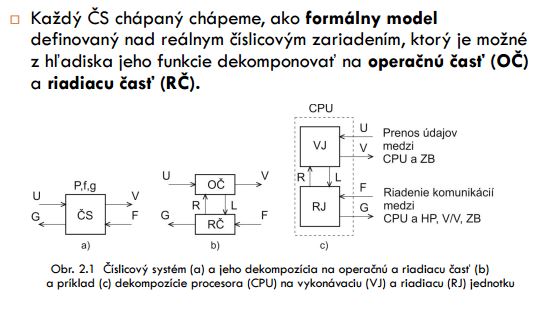
* **SÚ:** procesory, pamäte, V/V jednotky, zbernice, **PÚ:** operačné, komunikačné a riadiace funkčné jednotky, zbernice, funkčné moduly, **LÚ**: sekvenčné a kombinačné logické obvody (členy), **TÚ**: tranzistory, rezistory, prepojenia, kontakty

**Špecifikácia fyzickej realizácie (oblasť realizácie, konštrukcie, výroby)**

* **SÚ:** fyzické rozčlenenie systému na makromoduly, plánovanie technologicky odlišných modulov, **PÚ:** plánovanie rozdelenia plochy čipu pre jednotlivé moduly, **LÚ:** bunky a plány ich umiestnenia v moduloch, **TÚ**: rozmiestnenie a prepojenie prvkov v čipe, návrh masiek (geometria čipu, technológia výroby čipu), **konštrukčná úroveň:** zdroje, mechanické časti, kabeláž, dizajn

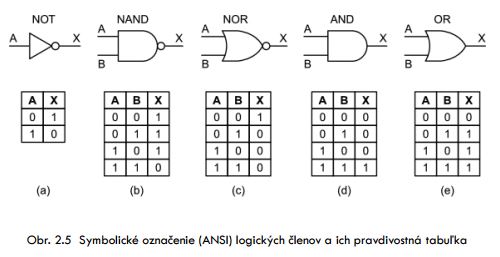
**Číslicový (digitálny) systém na abstraktnej úrovni**

* **definovaný sedmicou ČS = (U, V, P, F, G, f, g)** kde
  + **U, V, F, G** sú premenné definované vo vonkajších kanáloch ČS, určených na jeho styk s okolím, ktoré môžu byť*: údajové* (zobrazujú spracúvané informácie), *komunikačné* (zobrazujú informácie o spracúvaných údajoch), prezentované ako riadiace, informačné a stavové premenné,
  + **P** – premenná, vyjadrujúca stav ČS,
  + **f, g** – funkčné vzťahy, ktoré vyjadrujú reakciu ČS na podnety definované hodnotami nasledujúcich premenných, pôsobiacich v príslušných kanáloch:
  + V = f( U, F), G = g(P)



**2.2 Stavebné prvky číslicových počítačov. Boolova algebra, elementárne a odvodené logické funkcie. Symbolické označenie logických členov a ich pravdivostná tabuľka. Klasifikácia základných stavebných prvkov ČS (charakteristika).**

**Elementárne logické funkcie** Logický súčin - AND Logický súčet - OR Negácia - NOT ♦ **Odvodené (vektorové) logické funkcie** Negácia logického súčinu - NAND Negácia logického súčtu – NOR Nerovnosť - XOR Rovnosť – NXOR



**Klasifikácia základných prvkov**

* **Obvody na riadenie prenosu údajov a vnútorných komunikácií** 
  + Hradlovacie obvody a budiče, Prevodníky kódov, Kódery a dekódery, Multiplexory a demultiplexory.
* **Obvody na realizáciu predikátov**
  + Pre riadenie ľubovolného procesu v ČP je potrebné priebežne získavať rad informácií o stave, priebehu, korektnosti, požiadavkách a výsledkoch kontroly vykonávania tohto procesu. ⁄ Tieto informácie sú spravidla definované ako predikáty nad obsahmi prvkov a komponentov (funkčné jednotky, registre, zbernice a pod.) ČP. ⁄ Nosičom informácií je v tomto prípade stavovo-informačná premenná 𝐿, ktorá nadobúda jedničkovú hodnotu v prípade, ak predikát (napríklad, výrok nad obsahom registra) je splnený a nulovú hodnotu, ak splnený nie je.
* **Obvody na realizáciu operácií (výpočtov)** 
  + Sčítačky a akumulátory, ⁄ Inkrementory a počítadlá, ⁄ Posúvacie obvody, ⁄ Obvody logických a špeciálnych operácií (napr. invertor priameho kódu).
* **Obvody na pamätanie údajov**
  + sú určené na ukladanie (pamätanie) údajov v tvare jedno a viacbitových slov ν preklápacie obvody, ν registre, ν pamäte

**2.3 Stavebné prvky číslicových počítačov. Hradlovacie obvody a budiče. Prevodníky kódov. Kodéry a dekodéry. Multiplexor. Demultiplexor.**

* **Hradlovacie obvody a** **budiče**  Hradlo je základný logický člen mikroarchitektúry ČP, ktorého priechodnosť medzi vstupným (𝑋) a výstupným (𝑌) kanálom označeným rovnakopomenovanou vstupnou a výstupnou premennou 𝑋 a 𝑌 je definovaná vzťahom:

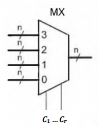


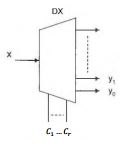
kde **𝑋** – je vstupná premenná v kanále 𝑋, **𝑌** – výstupná premenná v kanále 𝑌, **𝑅** – riadiaci signál

* Významnou aplikáciou hradla pri riadení komunikačných operácií je **riadenie prenosu údajov z viacerých zdrojov do jedného cieľového zdroja, ktorým je zbernica**.
* V súčasností sa pri obvodovom riešení používajú **trojstavové hradlá**, ktoré poskytujú úspornejšie riešenie v porovnaní s klasickými hradlami a vyššie spínacie rýchlosti v porovnaní s hradlami s otvoreným kolektorom.
* Ak trojstavové hradlá nie sú súčasťou príslušných výstupných zdrojov informácií pripájaných ku zbernici, ale tvoria samostatné prvky, nazývajú sa **budiče**. Obvody, ktoré sú určené na obojsmernú komunikáciu so zbernicou sa nazývajú **budiče/prijímače**

♦ **Prevodníky kódov**,

Funkčná schéma prevodníka ľubovoľného typu predstavuje kombinačný obvod, ktorý sa v ČS aktivuje spravidla výberovým signálom CS pôsobiacim na riadiacom vstupe prevodníka.

* **Kódery a dekódery,**
  + Dekodér s 𝑛 vstupnými a 𝑚 výstupnými premennými (𝐷𝐾 (𝑛 × 𝑚)) definovaný vyššie uvedenými vzťahmi sa nazýva **úplný dekodér**.
  + Dekodér, v ktorom medzi počtom 𝑛 vstupných a 𝑚 výstupných premenných platí 𝑚 < 2 𝑛 – 1, sa označuje ako dekodér **neúplný**.
* **Multiplexor (MX)**
  + Je to kombinačný logický obvod, ktorý pracuje ako riadený elektronický prepínač. Podľa kombinácie bitov na adresových vstupoch (na obr. sú to vstupy 𝐶1 a 𝐶2) prepína jeden z 𝑁 údajových vstupov (na obr. sú to vstupy 𝑆1 až 𝑆4 na jeden údajový výstup (𝐷).
* **Demultiplexor (DX)** 
  + Je opakom multiplexoru. Má jeden vstupný kanál a niekoľko výstupných kanálov. Podobne, ako v prípade MX, sa distribúcia jednobitového vstupného slova 𝑋 na jeden z jeho výstupných kanálov uskutočňuje prostredníctvom riadiacich vstupov 𝐶1, 𝐶2, ..., 𝐶𝑟 , kde 𝑟 = log2 n



**3.1 Stavebné prvky číslicových počítačov. Preklápacie obvody: typy, charakteristika. Asynchrónne preklápacie obvody : RS (+zapojenie na báze NOR aj NAND + pravd. tabuľka), T (+pravd. tabuľka). Synchrónny preklápací obvody D (+pravd. tabuľka).**

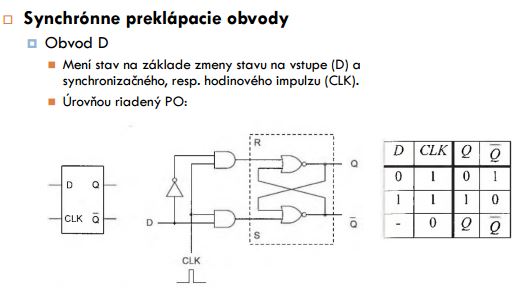
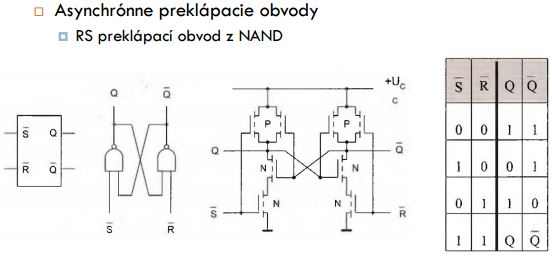
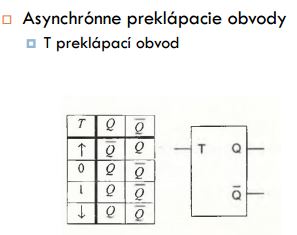
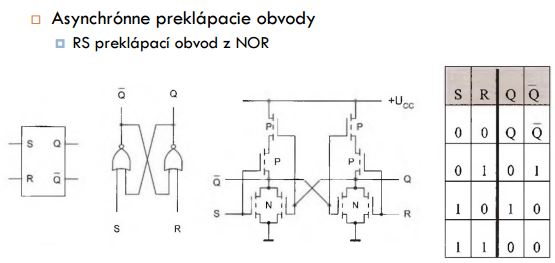
**Preklápacie obvody (PO)**

* sú určené na pamätanie jednobitových slov ⁄ jeden alebo viac vstupov I1 , I2 , ..., In (vstupné kanály) ⁄ dva navzájom komplementárne výstupy Q a Q (výstupné kanály). ⁄ pamäťové správanie rôznych typov PO je interpretované spravidla Mooreovým automatom. ⁄ astabilné, monostabilné, bistabilné ⁄ synchrónne vs. Asynchrónne

**Asynchrónne preklápacie obvody**

* Najjednoduchšie sekvenčné súčiasty ⁄ Reagujú na zmenu vstupných signálov okamžite ⁄ Preklápací obvod vznikne spojením dvoch negujúcich log. členov do okruhu. Takto je vytvorená spätnoväzobná slučka s celkovým fázovým posunom 360° , t.j. ide o kladnú spätnú väzbu

.



**3.2 Stavebné prvky číslicových počítačov. Základné vlastnosti hranou riadeného PO. Nulovanie a nastavenie PO. Blokovanie preklápacieho obvodu. Synchrónny preklápací obvody T a JK (+pravd. tabuľky). Registre: charakteristika a základné operácie. Pamäte, funkcia pamäte v riadiacej časti ČS.**

**Základné vlastnosti hranou riadeného PO**

* Rozhodujúci je stav na vstupoch v tesnom okolí nábežnej hrany hodinového impulzu. ν V iných okamžikoch sú zmeny na vstupoch bezvýznamné. ν Nevhodné časovanie hodinových impulzov a vstupných signálov môže vyvolať metastabilný stav. ν Metastabilný stav má za následok nespoľahlivú funkciu preklápacieho obvodu. PO sa môže rozkmitať, alebo sa môže preklopiť do nesprávneho stavu, alebo sa môže preklopiť do správneho stavu s veľkým oneskorením. Dobu oneskorenia nie je možné predvídať.

**Nulovanie a nastavenie PO**

* Niektoré PO D, T a JK majú okrem vstupov CLK, D, T, J a K aj asynchrónne vstupy nulovania a nastavenia výstupu. ⁄ Sú účinné vždy bez ohľadu na ostatné vstupy. ⁄ Vstup nulovania, ktorým je vynútený stav 𝑄 = 0, je označovaný ako 𝑅 (reset) alebo 𝐶𝐿𝐾 (clear) aktívny v stave 0. ⁄ Vstup nastavenia, ktorým je vynútený stav Q = 1, je označovaný ako 𝑆 (set) alebo 𝑃𝑅𝐸 (preset) aktívny v stave 0. ⁄ Súčasné vyvolanie nulovania a nastavenia môže viesť na nedefinovaný stav PO.

**Blokovanie preklápacieho obvodu**

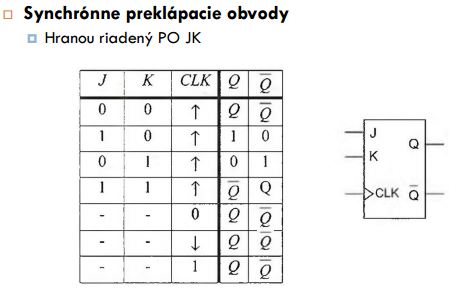
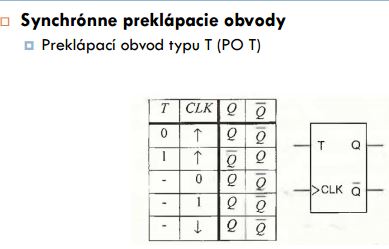
* Pre spoľahlivú činnosť PO je bezpodmienečne nutné dodržať podmienky správneho časovania vstupných signálov vzhľadom na hodinové impulzy. ⁄ Často je potrebné hodinové impulzy prepúšťať alebo blokovať. ⁄ Synchrónny spínač impulzov

**Registre**

* je usporiadaný súbor preklápacích obvodov (PO) ⁄ je určený na zápis a pamätanie slov binárnych informácií ⁄ pamäťový register vs. registrov všeobecného použitia

**Pamäte**

* na zapamätanie polí binárnych reťazcov
* **funkcia pamäte v riadiacej časti ČS** ν *statická pamäť typu ROM* (Read Only Memory) ν jednorázovo programovateľná pamäť *typu PROM* (Programmable ROM), ν preprogramovateľná pamäť *typu EPROM* (Erasable PROM), ν RWM (Read Write Memory)



**3.3 Stavebné prvky číslicových počítačov. Obvody na generovanie predikátov: komparátor, generátor parity, generátor priority.**

**Obvody na generovanie predikátov**

* Pre riadenie ľubovolného procesu v ČP je potrebné priebežne získavať rad informácií o stave, priebehu, korektnosti, požiadavkách a výsledkoch kontroly vykonávania tohto procesu. ⁄ Tieto informácie sú spravidla definované ako predikáty nad obsahmi prvkov a komponentov (funkčné jednotky, registre, zbernice a pod.) ČP. ⁄ Nosičom informácií je v tomto prípade stavovo-informačná premenná 𝐿, ktorá nadobúda jedničkovú hodnotu v prípade, ak predikát (napríklad, výrok nad obsahom registra) je splnený a nulovú hodnotu, ak splnený nie je.

**Komparátor**

* je funkčný prvok ČP určený na vyhodnocovanie relácií „ = , ≠, , ≤ , ≥“ medzi dvoma hodnotami binárnych slov 𝑋 = 𝑥𝑛𝑥𝑛−1 … 𝑥1𝑥0 a 𝑌 = 𝑦𝑛𝑦𝑛−1 … 𝑦1𝑦0.

**Generátor parity**

* je funkčný prvok komunikačných obvodov ČP určený na zabezpečenie korektného prenosu informácii po prenosovej ceste, resp. na indikáciu nekorektného prenosu. ⁄ Obvodové riešenie GP je založené na kontrole párneho alebo nepárneho počtu jedničiek v binárnom zobrazení slova prenášaného medzi jeho vysielačom a prijímačom. ⁄ Na základe uvedeného generátor parity generuje predikát „dané n-bitové slovo obsahuje párny, resp. nepárny počet jedničiek“.

**Generátor priority**

* Generátor priority je obvod na určenie priority signálov reprezentovaných hodnotami binárnych premenných daného slova podľa vopred zadaných kritérií. Napríklad, umožňuje v danom slove identifikovať rád s najvyššou prioritou.

**3.4 Stavebné prvky číslicových počítačov - FPGA. Čo je FPGA. Funkcia a štruktúra FPGA. Jazyky HDL. Vznik konfigurácie FPGA z HDL opisu.**

**Čo je FPGA**? Field Programmable Gate Array. Programovateľné hradlové pole umožňuje vytvoriť „ľubovoľný“ digitálny obvod. Sám o sebe nedokáže vykonávať žiadnu funkciu na rozdiel od mikrokontroléra, ktorý disponuje rôznymi integrovanými modulmi

**Funkciu** definuje „programovateľná logika“ υ Sú väčšinou volatilné zariadenia υ FLASH pamäť υ po pripojení na zdroj stiahne sa konfigurácia do FPGA υFPGA nie je okamžite po zapnutí funkčné a trvá niekoľko ms (~10-100) kým „nabehne“

**Štruktúra**

* FPGA disponujú vysokým počtom vývodov (~100-1000), preto sú bežne FPGA najčastejšie v BGA (Ball grid array) púzdrach, čo práve znižuje možnosti amatérskeho použitia
* Základná štruktúra Logické bloky (Logical Block) υ Programovateľná prepojovacia matica (Switch Matrix, Switch Boxes) υ Programovateľné I/O (Input Output Block)

**HDL** (Hardware Description Language) υNa rozdiel od konvenčných jazykov sa tu kód nevykonáva (výnimkou je simulácia), ale je z neho odvodená konfigurácia FPGA. υĎalším rozdielom je masívne paralelný priebeh. K tomuto účelu sa využíva konceptu tzv. procesova signálov

**HDL – Low level**

* VHDL - Európa
* Verilog – USA
* SystemVerilog

**HDL – High level**

* SystemC (C++)
* Catapult C (ANSI C, C++)
* Vivado HDL (ANSI C, C++)
* Visual HDL (C++)
* MyHDL (Python)

**Vznik konfigurácie FPGA z HDL opisu**

* **KROK 1 : Syntéza**: Táto časť analyzuje HDL opis a hľadá zodpovedajúce hardvérové štruktúry v podobe kombinačných a sekvenčných logických obvodov.
* **KROK 2 : Mapovanie na cieľovú technológiu** Na základe vybraného typu FPGA „mapper“ sa snaží nájsť najúspornejšie/najrýchlejšie ekvivalenty v syntéze nájdených komponentov. υ Zisťuje sa, či je návrh realizovateľný na zvolenom FPGA (počet dostupných komponentov, časové charakteristiky)
* **KROK 3 : Rozmiestnenie a prepojenie** Hľadá sa vhodné usporiadanie a prepojenie jednotlivých komponentov FPGA v prepojovacej matici. υ Rieši sa hlavne časovanie tzv. static timing analysis, ktorá určí či návrh bude fungovať na zvolenej frekvencii (a iných časovacích obmedzeniach napr. hold/setup time).

**4.1 Počítačová aritmetika a algoritmy. Reprezentácie údajov. Základné pojmy. Priamy, inverzný, doplnkový kód. Kód s posunutou nulou. Veľký a malý endián. Sprístupňovanie slov v pamäti.**

**Reprezentácia údajov**

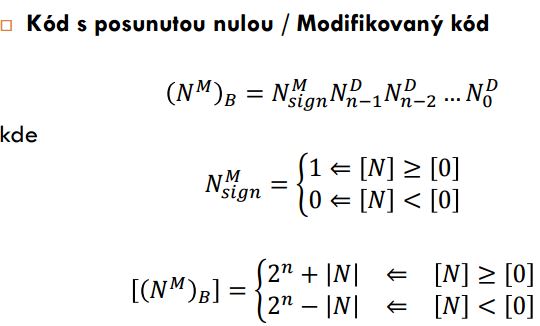
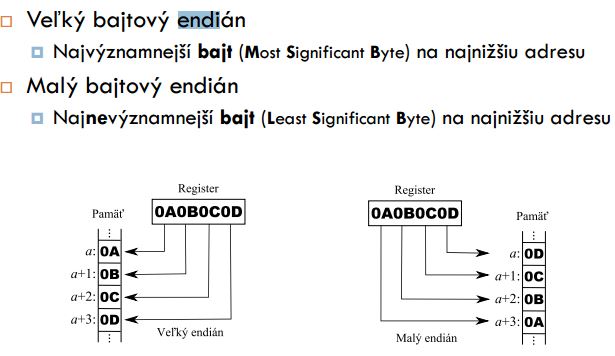
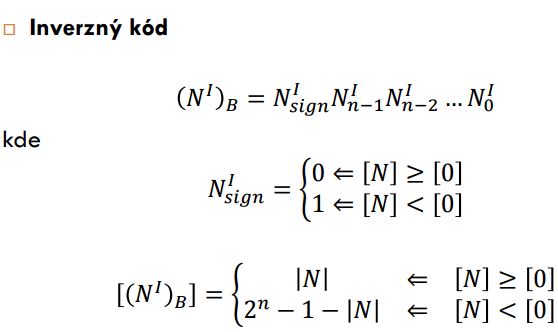
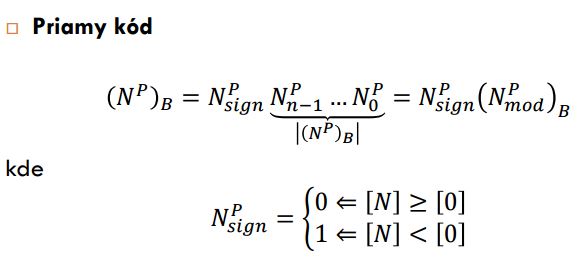
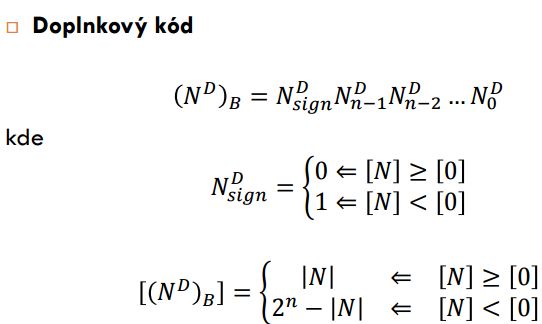
* Znaky ⁄ Celé čísla ⁄ Binarizované dekadické čísla (Binary Coded Digit - BCD) ⁄ Čísla v pohyblivej rádovej čiarke

**Základné pojmy**

* Ľubovoľnú diskrétnu informáciu v číslicovom počítači zobrazujeme prostredníctvom sústavy symbolov, ktorým môžeme priradiť rôzne číselné hodnoty. ⁄ Reprezentácia číselných hodnôt vopred definovanou sústavou číslic sa nazýva číselná sústava(Pozičné ⁄ Nepozičné ⁄ Symetrické ⁄ Nesymetrické)

**Sprístupňovanie slov**

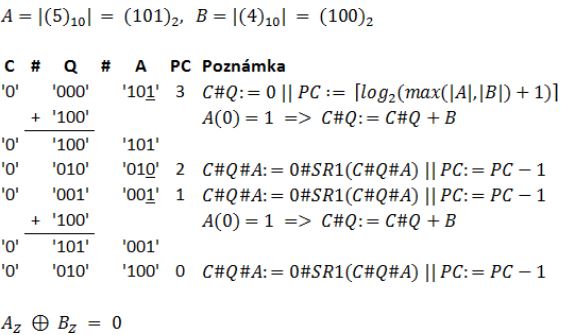
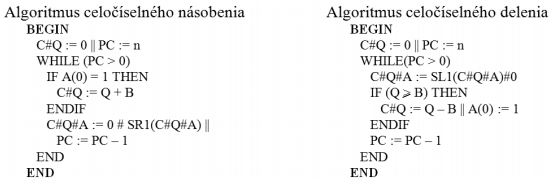
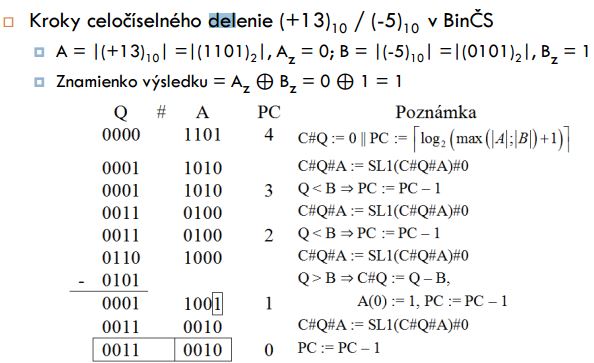
* Zarovnané sprístupňovanie bajtov ν Jednoduchosť návrhu HW ν Jednoduché sprístupňovanie ν Využitie v RISC ⁄ Nezarovnaný výber bajtov ν Zložitý návrh HW ν Zložité sprístupňovanie ν Využitie v CISC



**4.2 Počítačová aritmetika a algoritmy. Reprezentácie údajov. Operácie s celými číslami bez znamienka: násobenie a delenie (algoritmy + príklady).**

**Reprezentácia údajov**

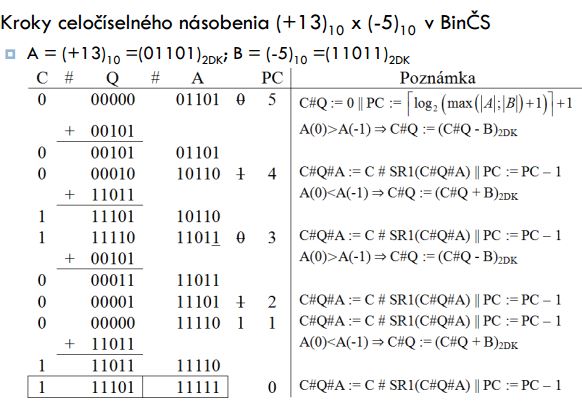
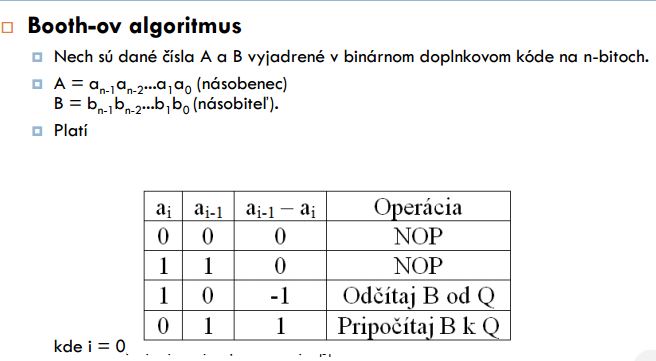
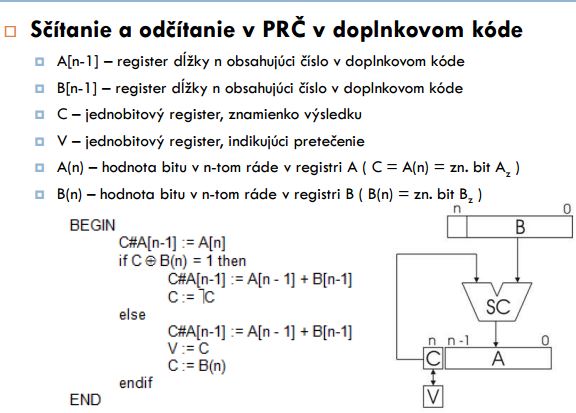
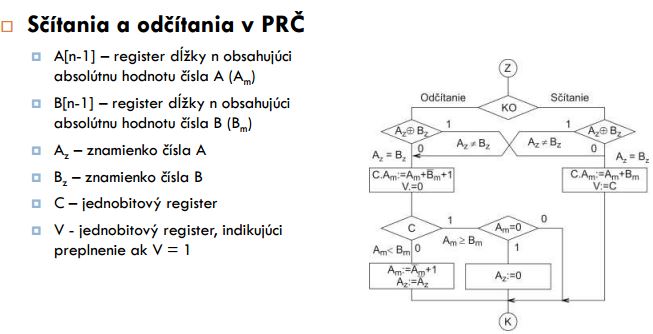
* Znaky ⁄ Celé čísla ⁄ Binarizované dekadické čísla (Binary Coded Digit - BCD) ⁄ Čísla v pohyblivej rádovej čiarke



**4.3 Počítačová aritmetika a algoritmy. Reprezentácie údajov. Operácie s celými číslami bez znamienka: sčítanie a odčítanie v PK a DK (algoritmy), Boothov algoritmus násobenia (algoritmus + príklad).**

**Reprezentácia údajov**

* Znaky ⁄ Celé čísla ⁄ Binarizované dekadické čísla (Binary Coded Digit - BCD) ⁄ Čísla v pohyblivej rádovej čiarke



**4.4 Počítačová aritmetika a algoritmy. Pohyblivá rádová čiarka, IEEE-754. Násobenie v PHRĆ (algoritmus + príklad). Časové charakteristika algoritmov.**

**Časové charakteristiky algoritmov**

Čas vykonania operácie závisí od operačnej rýchlosti procesora

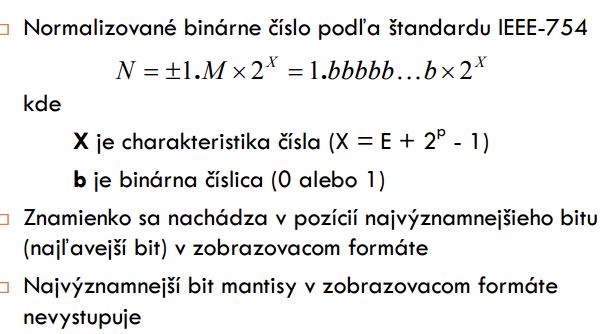
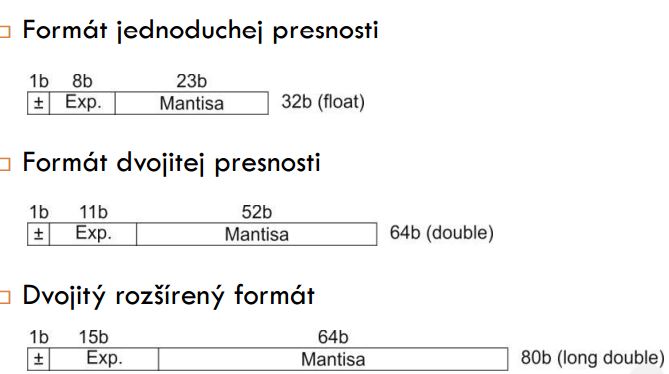
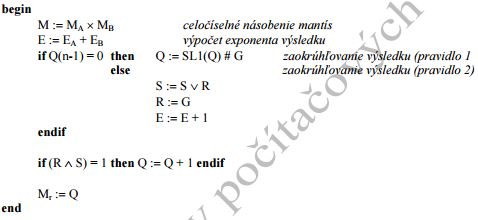
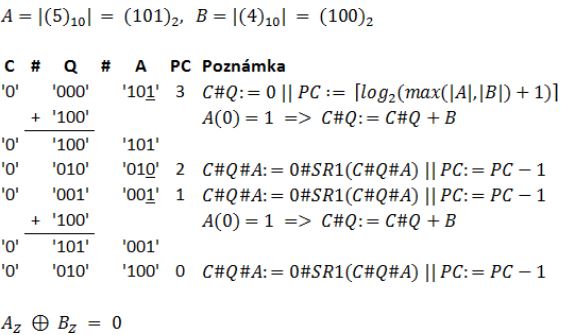
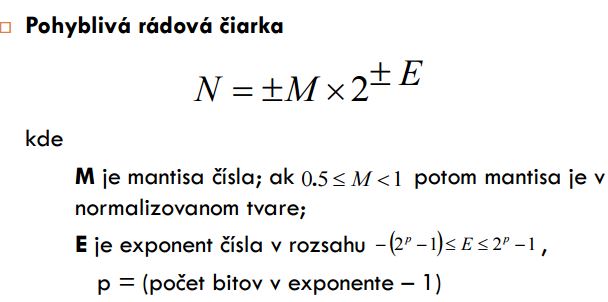
Čas potrebný na vykonanie daného algoritmu sa určí z jeho časového diagramu, v ktorom:

Operačným a rozhodovacím vrcholom sú postupne priradené vrcholy grafu s poradovými číslami 0, 1, 2, . . . ,

Hranám (i, j) sú priradené pravdepodobnosti p prechodu z jedného vrcholu do druhého, pričom platí:



kde Di je množina všetkých hrán vychádzajúcich z vrcholu (i).



**5.1. Inštrukčne orientovaná architektúra. Inštrukčno-orientované architektúry CISC, RISC (+obrázky CISC a RISC). Inštrukcie a ich funkcie. Súbor inštrukcií. Formát inštrukcií. Adresovacie módy.**

**CISC (Complex Instruction Set Computer)**

* veľký počet zložitých inštrukcií, ν veľký počet formátov inštrukcií a adresovacích módov, ν špeciálne inštrukcie.

**RISC (Reduced Instruction Set Computer)**

* malý počet jednoduchých inštrukcií, ν jednoduchý formát inštrukcií, ν malý počet adresovacích módov, ν LOAD/STORE.

**Inštrukcie a ich funkcia**

* súbor inštrukcií, formát inštrukcie, vnútorné registre, adresovacie módy, riadenie inštrukčného toku.

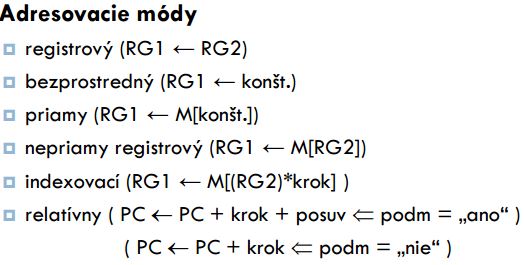
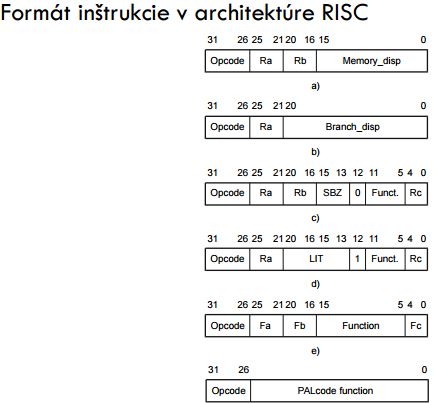
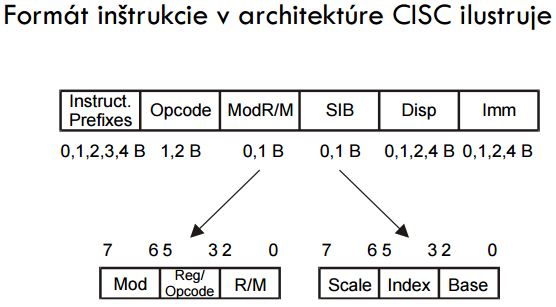
**Súbor inštrukcií** – kľúčový komponent ISA

* prenos („mov“), logické operácie („and“) ,PRČ aritmetika („add“), PHRČ aritmetika („fmul“), Bitové manipulácie („test“), multimediálne inštrukcie („padd“), riadenie prenosu („lea“), riadenie systému („hlt“), špeciálne inštrukcie (“rdtsc”)

**Formát inštrukcie** – informácie potrebné k vykonaniu inštrukcie

<FI>::=<KO><PKO>{<PAO>[,AO]}

* **FI** – formát inštrukcie **KO** – kód operácie **PKO** – príznak kódu operácie **PAO** – príznak adresy operandov **AO** – adresa operandov
* Väčšina RISC ISA používajú 3-adresový FI , v ktorom všetky inštrukcie majú jednotnú dĺžku 32-bitov. ⁄ Táto hodnota sa posúva do výšky 64 -128 bitov.
* CISC ISA vychádzajú z architektúr typu register-pamäť s premenlivou dĺžkou inštrukcií.
* Premenlivú dĺžku inštrukcií používajú i zásobníkové architektúry, ktorých predstaviteľmi sú napr. procesory Java (PicoJava, MicroJava).
* Akumulátorové typy architektúr nachádzajú najväčšie uplatnenie v riadiacich jednotkách (mikrokontroleroch) mikroriadenia v systémoch diskrétnej automatiky.



**5.2. Výkonnostné parametre ČP. Základné charakteristiky ČP: výkonnosť, systémové vlastnosti. Hodnotenie výkonnosti (TCPU, ACPI). Jednotky výkonnosti.**

**Základné charakteristiky ČP**

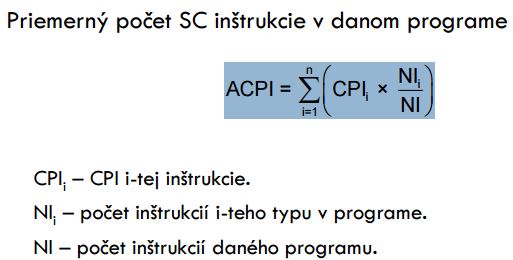
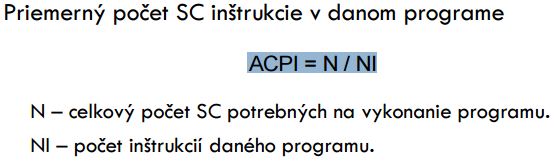
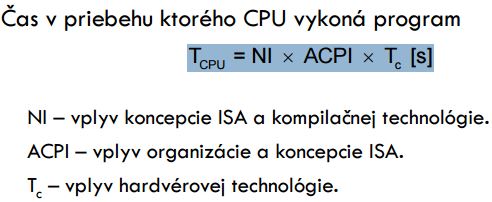
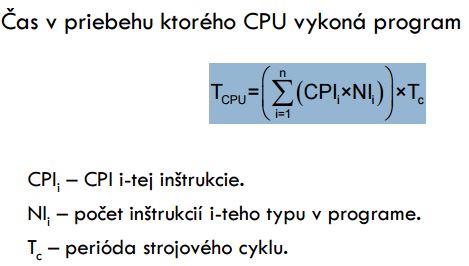
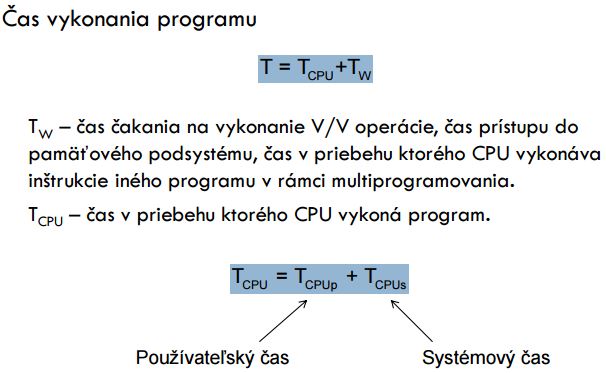
* **Výkonnosť ČP** – kvantitatívne hodnotenie ν čas odozvy (Response Time), ν priepustnosť (Throughput), ν koeficient vyťažiteľnosti (Efficiency).
* **Systémové vlastnosti** – kvalitatívne hodnotenie ν súbor inštrukcií, ν kompilačná technológia, ν implementácia a riadenie CPU, ν technológia a hierarchia pamäťového podsystému.

**Hodnotenie výkonnosti počítačových** **architektúr**

* popis všeobecných poznatkoch ⁄ metriky a pracovná záťaž

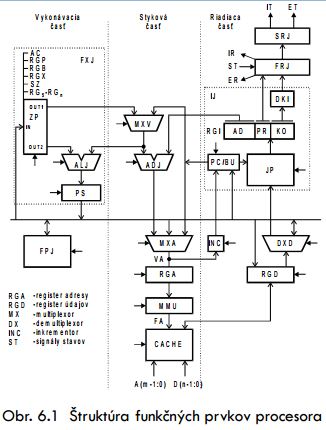
**Jednotky výkonnosti**

* MIPS ν 106 inštrukcií za sekundu (Milion Instruction Per Second, odvodené jednotky: GIPS)
* FLOPS ν počet operácií za sekundu vykonaných v pohyblivej rádovej čiarke (odvodené jednotky: MFLOPS, GFLOPS, TFLOPS),
* KLIPS ν 103 logických vývodov za sekundu (Logic Inference Per Second, určené na definovanie výkonnosti počítačového prostredia na riešenie úloh umelej inteligencie (UI), jeden vývod zahrňuje približne 1000 inštrukcií)
* účelovo vytvorené „benchmarky“
  + **Dhrystone** ν určené na meranie výkonnosti ČP pre vykonávanie programov, v ktorých sa nevyskytujú operácie FP, vytvorených na základe benchmarkov, ktoré používajú inštrukčné mixy (určenie: systémové programovanie, testovanie CPU a kompilátorov),**Whestone** ν určené na meranie výkonnosti ČP pre vykonávanie "fortranovských" programov s operáciami FP, vytvorených na základe benchmarkov, ktoré používajú celočíselné a FP operácie, vrátane indexovania polí, volanie procedúr, podmienené skoky a trigonometrické a transcedentálne funkcie.**LINPACK** ν určené na testovanie aplikačných programov (riešenie lineárnych rovníc: 100 x 100, 1000 x 1000, N x N), **SPEC** určené na porovnávanie výkonnosti voči etalónu najmä série počítačov pri riešení vedeckých a inžinierskych aplikácií



**6.1. Centrálna procesorová jednotka. Definícia procesora. Funkčné jednotky vykonávacej časti + nakreslite VČ z obr. 6.1. Komponenty vykonávecaj časti: operačné jednotky, polovičná sčítačka (+obr.), úplná sčítačka (+obr.), sčítačka so zrýchleným prenosom. Typy sčítačiek (vymenovať, viď. téma 3). Riadenie vykonávacej časti..**

**Procesor** je multifunkčná jednotka číslicového počítača (ČP) alebo počítačového systému (PS), ktorá riadi vykonávanie inštrukcií programu, uloženého v pamäti, v procese spracovania informácií.



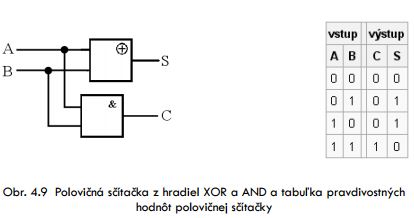
**Vykonávacia časť CPU**

* je určená na vykonávanie operácií definovaných kódom operácie inštrukcie.
* **Pozostáva z**
  + pola vnútorných registrov, konštrukčne riešených spravidla ako zápisníková pamäť,
  + rôzneho počtu operačných jednotiek (OJ),
    - aritmeticko-logická jednotka ALJ,sčítačka, násobička, delička a pod./posúvací obvod, koprocesor(y) (napr. FPU – Fixed Point Unit).

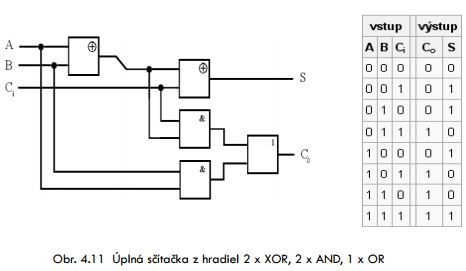
**Spontánne riadenie** OJ je realizovaná ako KLO (kombinačný logický obvod) ν OJ nepotrebuje osobitné riadenie ν Výpočet je závislý na rýchlosti šírenia signálu(/-ov) kombinačným log. obvodom

**Algoritmické riedenie** Osobitné riadenie → Riadiaca jednotka (RJ) ν RJ generuje riadiace príkazy (signály) pre univerzálnu funkčnú jednotku (FJ)

**Polovičná sčítačka** realizuje sčítanie dvoch jednomiestnych binárnych čísel



* **vstup**: dva jednobitové sčítance A, B, **výstup**: jednobitový súčet a jednobitový príznak prenosu do vyššieho rádu (Carry flag)
* odovzdáva ďalej príznak prenosu, ale sama ho nedokáže spracovať
* S = A ⊕ B; C = AB



**Úplná sčítačka** realizuje sčítanie dvoch jednomiestnych binárnych čísel, pripočítava prenos z predchádzajúceho rádu

* **vstup:** tri jednobitové sčítance Ai , Bi , Ci (Carry-in) **výstup**: jednobitový súčet a jednobitový príznak prenosu do vyššieho rádu Ci+1 (Carry-out)



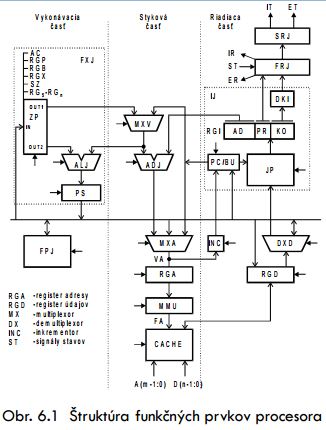
**??sčítačka so zrýchleným prenosom??**

**Typy sčítačiek** ⁄ Polovičná, úplná sčítačka a sčítačka so sériovým prenosom (Ripple carry-adder) ⁄ Sčítačka so zrýchleným prenosom (Carry-look-ahead adder; CLA) ⁄ Sčítačky s blokovým prenosom: Block carry look-ahead adder (BCLA), Ripple block carry look-ahead adder (RCLA) ⁄ Paralelné prefixové sčítačky (Parallel prefix adders: Ladner-Fisher adder, Kogge-Stone adder, Brent-Kung adder, Han-Carlson adder) ⁄ Sčítačka s výberovým prenosom (Carry-select adder; CSA), Sčítačka s tranzitívnym prenosom (Carry-skip adders) ⁄ Sčítačka s podmieneným prenosom (Conditional sum adder), sčítačka s uchovaným prenosom (Carry-save adder), ⁄ Polovičná odčítavačka (Half subtractor), Úplná odčítavačka (Full subtractor) BCD sčítačka, BCD odčítavačka

**6.2. Centrálna procesorová jednotka. Definícia procesora. Funkčné jednotky riadiacej časti + nakreslite RČ z obr. 6.1. Klasifikácia riadiacej jednotky, komponenty riad. jednotky a ich funkcia. Inštrukčný cyklus.**

**Procesor** je multifunkčná jednotka číslicového počítača (ČP) alebo počítačového systému (PS), ktorá riadi vykonávanie inštrukcií programu, uloženého v pamäti, v procese spracovania informácií.

**Riadiaca časť CPU**



* je určená na časovanie a riadenie postupnosti príkazov, vykonávaných jednotlivými prvkami procesora pri vykonávaní inštrukcií.
* **Pozostáva** z
  + inštrukčnej jednotky (IJ), určenej na zreťazený predvýber inštrukcií (jednotka predvýberu - JP) a organizáciu postupnosti ich vykonávania v súčinnosti s programovým počítadlom (PC - Program Counter), jednotkou vetvenia (BU - Branch Unit), registrom inštrukcií RI a dekodérom inštrukcií (DKI),
  + synchronizačnej (SRJ) a funkčnej (FRJ) riadiacej jednotky, určenej na generovanie interných a externých signálov IT, ET, IR, ER.

**Klasifikácia RJ**

* RJ s pevnou logikou - spontánne riadenie, RJ s programovateľnou logikou - algoritmické riadenie

**Komponenty RJ**

* Jednotka predvýberu inštrukcií (JPI) - výber ν dekódovanie
* Inštrukčná jednotka (IJ) - rozklad na mikrooperácie , vykonanie mikrooperácie
* Jednotka adries skokov (JAS) - riadenie a plánovanie vykonávania mikrooperácie

**Jednotka predvýberu inštrukcií** výber postupnosti inštrukcií programu z pamäte cache ⁄ ich ukladanie do vyrovnávacej pamäte FIFO vo funkcii frontu vykonateľných inštrukcií ⁄ dekódovanie v dekodéri inštrukcií (DKI)

**Inštrukčná jednotka** Riadi a organizuje vykonanie inštrukcie ⁄ Generuje mikrooperácie ν Riadiace slová / signály ν Dĺžka vykonania mikrooperácie je daná mikrotaktom ⁄ Superskalárne architektúry ν Preusporiadanie inštrukcií

**Jednotka adries skokov (JAS)** riadi a organizuje poradie vykonávania inštrukcií po výskyte podmieneného skoku v lineárnej postupnosti inštrukcií programu. ⁄ pri prúdovom spracovaní inštrukcií sú „skokové“ inštrukcie, v procese ich predvýberu vo vopred definovanom počte, umiestnené v pamäti FIFO jednotky predvýberu inštrukcií.

**Inštrukčný cyklus**

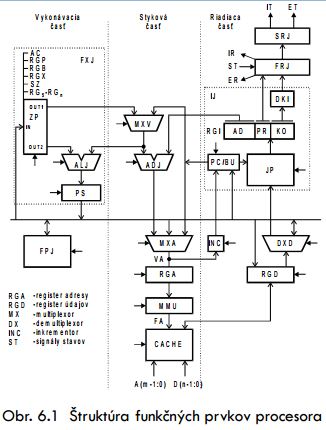
* je postupnosť elementárnych krokov (strojových cyklov - SC), v priebehu ktorých procesor vykoná operáciu definovanú inštrukciou programu.
* Čas TSC vykonania SC je TSC = n X TC
* TC – perióda taktovacej (hodinovej) frekvencie procesora, n – počet periód TC .
* Inštrukčný cyklus CPI pozostáva z m fáz Fm z ktorých každá trvá ni  SC, kde i = 1, 2, ... , m.

CPI = F1+F2+...+Fn

* Jednotlivé fázy vykonania inštrukcie sa realizujú prostredníctvom postupnosti (elementárnych) operácií, ktoré sa nazývajú mikrooperácie.

**6.3. Centrálna procesorová jednotka. Definícia procesora. Funkčné jednotky stykovej časti + nakreslite SČ z obr. 6.1. Komponenty stykovej časti a ich funkcia.**

**Procesor** je multifunkčná jednotka číslicového počítača (ČP) alebo počítačového systému (PS), ktorá riadi vykonávanie inštrukcií programu, uloženého v pamäti, v procese spracovania informácií.



**Styková (komunikačná) časť CPU** je určená na organizáciu styku procesora pri sprístupňovaní komponentov pamäťového a V/V podsystému ν zabezpečuje styk s ostatnými jednotkami ČP pri prenose informácií, ν zabezpečuje prenos informácií medzi procesorom a hlavnou pamäťou resp. V/V podsystémom.

**Komponenty stykovej časti**

* Jednotka správy pamäte - adresová jednotka (ADJ), určená na výpočet virtuálnej adresy (VA), ν jednotka riadenia pamäte (MMU - Memory Management Unit), určená na prevod virtuálnej adresy (VA) na fyzickú adresu (FA) pamäťového priestoru v hlavnej pamäti.
* Vyrovnávacia pamäť typu cache, určená na rýchlu komunikáciu procesora s pamäťovým podsystémom.
* Zbernicová jednotka

**Jednotka správy pamäte**

* Prevod logických adries (LA) na fyzické adresy (FA)
* Logický adresový priestor ν stránky ν adresa slova = bázová adresa stránky + posuv
* Fyzický adresový priestor ν stránkové rámy ν adresa slova = bázová stránkového rámu + posuv
* Prevodovú mapu reprezentuje pamäť preložených adries (PPA), označovaná tiež ako TLB (Translation Look aside Buffer) a tabuľka stránok (TS).

**Pamäť preložených adries**

* Obsahuje adresy najviac používaných stránok vo vopred definovanom časovom intervale práve prebiehajúcej časti programu.
* Tabuľka stránok -pamäť zodpovedajúcich dvojíc (LA, FA)
* Obe môžu byť uložené v osobitnej asociatívnej pamäti, v pamäti cache alebo v hlavnej pamäti ČP.
* TLB a TS tvoria hlavné komponenty štruktúrnej organizácie JSP.

**Zbernicová jednotka (ZJ)**

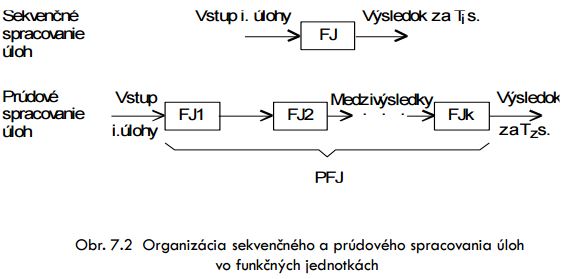
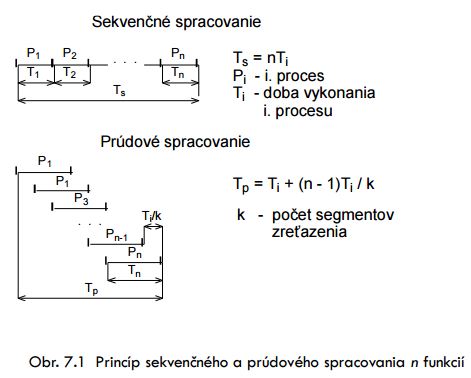
* Zabezpečuje rozhranie medzi procesorom a ostatnými komponentmi ČP pri vykonávaní V/V a pamäťových inštrukcií.
* Organizuje prenos informácií medzi procesorom V/V, resp. pamäťovým podsystémom ČP.
* Synchronizuje prenos informácií prostredníctvom zbernicových cyklov (V/V-ných, pamäťových, prerušovacích a pod.).

**7.1. Teória prúdového spracovania výpočtového procesu. Definícia zreťazenia. Podmienky zreťazenia, sekvenčné a prúdové spracovanie. Stupne systému zreťazenia. Základné pojmy: inicializácia zreťazenia, rezervačná tabuľka. latentnosť, stavový graf inicializácií.**

**Zreťazenie** je všeobecný princíp spracovania informácií založený na rozložení výpočtového procesu ľubovoľnej funkcie (úlohy, operácie, postupu a pod.) na oddelené kroky, ktoré sa prostredníctvom samostatných funkčných modulov, označovaných ako stupne (segmenty) zreťazenia, realizujú na princípe súčasného prekrývania týchto krokov v priebehu vykonávania za sebou nasledujúcich výpočtových procesov.

**Podmienky prúdového spracovania**

* Inštrukčný cyklus je rozložený na niekoľko fáz, ktoré sa môžu konkurenčne (prekrývane) vykonávať s inými fázami predchádzajúcich inštrukcií.
* Každá fáza sa vykonáva v priebehu jedného alebo viacerých strojových cyklov.
* Jednotlivým fázam vykonávania zodpovedajú na úrovni obvodového riešenia príslušné stupne (segmenty) zreťazenia.



**Stupne systému zreťazenia**

* Stupeň F(etch) - načítava inštrukcie z pamäte
* Stupeň D(ecode) - dekóduje inštrukcie
* Stupeň E(xecute) - vykonáva operáciu definovanú v IF KO
* Stupeň S(tore), resp. WB (Write Back) - zapisuje výsledok operácie do registra/pamäte

**Inicializácia zreťazenia** Začiatok procesu zreťazeného spracovania vstupného prúdu údajov zobrazeného v RTB.

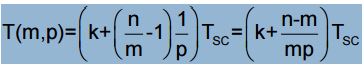
Postupnosť aktivácie jednotlivých stupňov v priebehu zreťazeného spracovania danej funkcie v PFJ sa zobrazuje v jej **rezervačnej tabuľke (RTB).**

**Latentnosť (L)** Počet synchronizačných intervalov (krokov) medzi začiatkami dvoch za sebou nasledujúcich inicializácií zreťazenia.

**Stavový graf inicializácií**  Vyjadruje prípustné stavy prechodov medzi postupnosťou následných inicializácií. Umožňuje vypočítať Lmin .

**7.2. Teória prúdového spracovania výpočtového procesu. Doba vykonania prúdu n inštrukcií. Efektívnosť prúdového spracovania. Základné úlohy syntézy prostriedkov na zreťazené spracovanie úloh. Základné typy zreťazených systémov. Základné pojmy: inicializácia zreťazenia, rezervačná tabuľka. latentnosť, stavový graf inicializácií.**

**Doba vykonania prúdu n inštrukcií**



**TSC** – perióda SC. **n** – počet prúdovo spracúvaných inštrukcií. **k** – počet fáz dekomponovanej inštrukcie. **m** – počet súčasne spracúvaných prúdov inštrukcií. **p** – počet inštrukcií v spracúvanom prúde šírky m, vykonanie ktorých začína postupne, v priebehu jedného SC.

**Efektívnosť prúdového spracovania** je definovaná vzťahom



t.j. prúdové spracovanie je k násobne rýchlejšie ako sekvenčné spracovanie, pričom spracovanie jedinej úlohy, rozloženej na k vykonávacích fáz, môže pri prúdovom režime trvať dlhšiu dobu, ako pri sekvenčnom režime (Ti ≤ kTk ).

**Základné úlohy syntézy prostriedkov na zreťazené spracovanie úloh** -Analýza procesu zreťazenia , Stratégia riadenia procesu zreťazenia , Návrh technických prostriedkov

**Inicializácia zreťazenia** Začiatok procesu zreťazeného spracovania vstupného prúdu údajov zobrazeného v RTB.

Postupnosť aktivácie jednotlivých stupňov v priebehu zreťazeného spracovania danej funkcie v PFJ sa zobrazuje v jej **rezervačnej tabuľke (RTB).**

**Latentnosť (L)** Počet synchronizačných intervalov (krokov) medzi začiatkami dvoch za sebou nasledujúcich inicializácií zreťazenia.

**Stavový graf inicializácií**  Vyjadruje prípustné stavy prechodov medzi postupnosťou následných inicializácií. Umožňuje vypočítať Lmin .

**8.1 Prúdová koncepcia skalárnych procesorov. Skalárne procesory (vymenovať). Hazardy prúdového spracovania, údajové hazardy a ich odstránenie. Podstata vzniku hazardu RAW a jeho odstránenie. Štrukturálne hazardy a jeho odstránenie. Hazardy riadenia a jeho odstránenie.**

**Skalárne procesory**

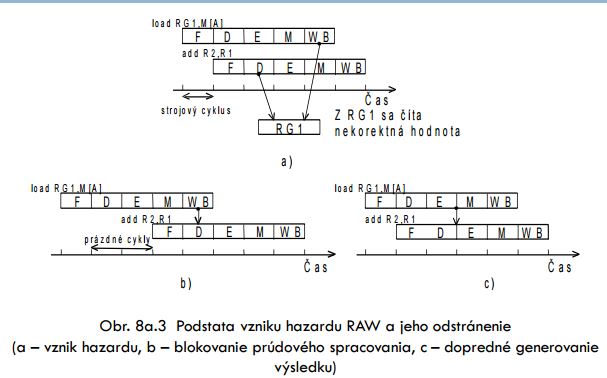
* IBM 801 (1975), Berkley RISC I, II Stanford MIPS,Sun SPARC (1987, microSPARC-II), Intel i860 (1988), MIPS R3000 (1988), Motorola M88000 (1988), AMD 29000 (1988), PicoJava (1998, zásobníková architektúra, hardvérová podpora programov Java)

Základný problém prúdového spracovania inštrukcií je vyriešenie konfliktov, pri požiadavke súbežného vykonávania ich rôznych fáz, ak tomu bránia objektívne dôvody.

Tieto dôvody sa nazývajú **hazardy zreťazenia** (prúdového spracovania), ktoré môžu byť ν Údajové,Štrukturálne,Riadenia

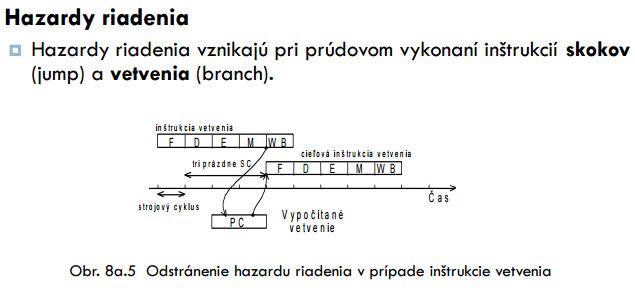
**Údajové hazardy** vznikajú v dôsledku nedostupnosti operandov pri vykonávaní inštrukcií. Nedostupnosť operandov vyplýva z definovania vzájomnej závislosti spoločných premenných v dvoch za sebou nasledujúcich inštrukciách programu. Vo všeobecnosti existujú nasledujúce závislosti: *údajová závislosť*, spôsobuje hazard typu RAW (Read After Write), *údajová antizávislosť*, spôsobuje hazard typu WAR (Write After Read), *výstupná závislosť*, spôsobuje hazard typu WAW (Write After Write).

Pri jednoduchom prúdovom spracovaní inštrukcií môže **vzniknúť** iba **hazard RAW**, ktorý spôsobuje využívanie spoločného registra v dvoch rôznych fázach nasledujúcich inštrukcií.



**Odstránenie:** softvérovými prostriedkami- vloženie prázdnej inštrukcie po inštrukcii, ktorá môže spôsobovať hazard; preusporiadanie vykonávania inštrukcií. hardvérovými prostriedkami- blokovanie prúdového spracovania, v dôsledku ktorého sa generujú prázdne cykly v prúdovom spracovaní inštrukcií; dopredné generovanie výsledku predchádzajúcej inštrukcie pre jeho použitie ako vstupného operandu nasledujúcej inštrukcie.

**Štrukturálne hazardy** ⁄ Štrukturálne hazardy vznikajú, ak dve za sebou nasledujúce inštrukcie využívajú tie isté zdroje technických prostriedkov zreťazeného systému. ⁄ Hazard sa môže riešiť: ν vložením prázdneho SC (oneskorenie vykonania druhej inštrukcie) alebo ν použitím viacnásobných zdrojov (rôznych registrov, ktoré časovo oddeľujú prístup do registrovej pamäte - RGM).



**8.2. Multiprúdové architektúry superskalárnych procesorov. Definícia a delenie multiprúdových architektúr. Superskalárne procesory: charakteristika, časový diagram zreťazeného spracovania inštrukcií, doba vykonania prúdu n inštrukcií, efektívnosť spracovania. Superprúdové procesory. Prúdové vektorové procesory.**

**Definicia**: Významnou črtou prúdových architektúr procesorov je, že vytvárajú podmienky na súbežné (viacprúdové) vykonanie inštrukcií. Stupeň súbežného vykonávania inštrukcií vyjadruje paralelizmus na inštrukčnej úrovni (prúdový paralelizmus) definovaný ako maximálny počet inštrukcií, ktoré sa môžu súbežne vykonávať pri ich prúdovom spracovaní.

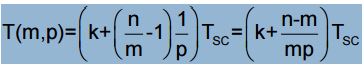
**Delenie: Podľa stupňa súbežného vykonávania inštrukcií**- superskalárne, superprúdové, superprúdovo superskalárne.

**???diagram???**

**Superskalárne procesory** Charakteristickou črtou superskalárneho prístupu vykonania inštrukcií je súbežné spracovanie viacerých prúdov inštrukcií, v dôsledku čoho sa architektúra superskalárneho procesora nazýva multiprúdová

* **bez dopredného prehľadávania** nie je z časového hľadiska úsporné, ⁄ pri prúdovom vykonávaní inštrukcií sa môžu vyskytovať prázdne cykly, resp. blokovanie prúdového spracovania, v dôsledku vzniku rôznych typov hazardov.
* **Dopredné prehľadávanie prúdu inštrukcií**⁄ umožňuje odstrániť vkladanie prázdnych cyklov a blokovanie prúdového spracovania.⁄ Prostredníctvom dopredného prehľadania vstupného prúdu inštrukcií sa tieto preusporiadajú a vykonajú (ukončia) v takom poradí, pri ktorom vzniká minimálny počet hazardov.

**Doba vykonania prúdu n inštrukci**



**TSC** – perióda SC. **n** – počet prúdovo spracúvaných inštrukcií. **k** – počet fáz dekomponovanej inštrukcie. **m** – počet súčasne spracúvaných prúdov inštrukcií. **p** – počet inštrukcií v spracúvanom prúde šírky m, vykonanie ktorých začína postupne, v priebehu jedného SC.

**Efektívnosť prúdového spracovania** je definovaná vzťahom

C:\Users\HoaiThuong\AppData\Local\Microsoft\Windows\INetCache\Content.Word\efekt.jpg

t.j. prúdové spracovanie je k násobne rýchlejšie ako sekvenčné spracovanie, pričom spracovanie jedinej úlohy, rozloženej na k vykonávacích fáz, môže pri prúdovom režime trvať dlhšiu dobu, ako pri sekvenčnom režime (Ti ≤ kTk ).

**Superprúdové procesory**

* Architektúra superprúdových procesorov predstavuje modifikáciu prúdových procesorov. Využívajú zrýchlené zreťazenie pri spracovaní inštrukcií. Pre účely zrýchleného zreťazenia sa používajú dva druhy frekvencie strojových cyklov zreťazenia (SCZ), taktujúcich prúdové vykonanie inštrukcie: ν externé (f e ), ν interné (f i ),
* f = p × f ; p > 1

**Prúdové vektorové procesory**

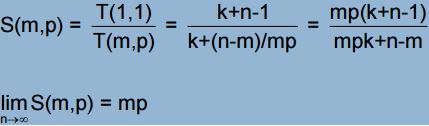
* Prúdový vektorový procesor je špecializovaný procesor určený na vykonávanie vektorových operácií. Vektorová inštrukcia zahrňuje veľký počet operandov (prvky vektora), ktoré sa spracúvajú v multiprúdovom prostredí vektorového procesora. Prúdové spracovanie vektorových inštrukcií, pri ktorom sa uplatňuje paralelizmus rozvinutých operandov, môže byť použité pri ľubovoľnom skalárnom, superprúdovom alebo superprúdovom spracovaní inštrukcií programu, ktorý je vektorizovateľný.

**8.3. Multiprúdové architektúry superskalárnych procesorov. Superprúdové a superprúdovo-superskalárne procesory: charakteristika, časový diagram zreťazeného spracovania inštrukcií, doba vykonania prúdu n inštrukcií, efektívnosť spracovania. Architektúry VLIW a prúdové procesory a ich charakteristika.**

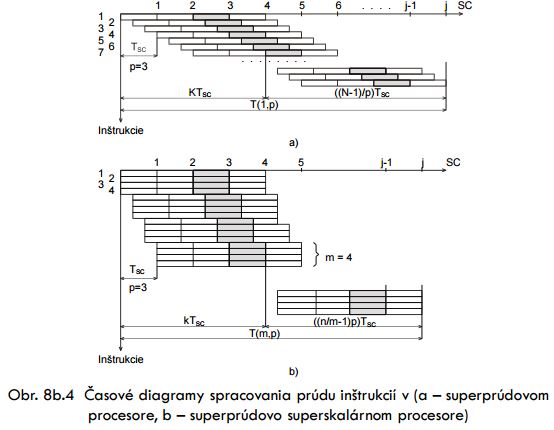
**Superprúdové procesory**

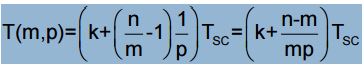
* Architektúra superprúdových procesorov predstavuje modifikáciu prúdových procesorov. Využívajú zrýchlené zreťazenie pri spracovaní inštrukcií. Pre účely zrýchleného zreťazenia sa používajú dva druhy frekvencie strojových cyklov zreťazenia (SCZ), taktujúcich prúdové vykonanie inštrukcie: ν externé (f e ), ν interné (f i ),
* f = p × f ; p > 1

**Superprúdovo superskalárna architektúra** Kombinácia prístupov superskalárneho a superprúdového spracovania inštrukčných. Pre zrýchlenie superprúdovo superskalárneho procesora S(m,p) platí



**Doba vykonania prúdu n inštrukci**





**TSC** – perióda SC. **n** – počet prúdovo spracúvaných inštrukcií. **k** – počet fáz dekomponovanej inštrukcie. **m** – počet súčasne spracúvaných prúdov inštrukcií. **p** – počet inštrukcií v spracúvanom prúde šírky m, vykonanie ktorých začína postupne, v priebehu jedného SC.

**Efektívnosť prúdového spracovania** je definovaná vzťahom

C:\Users\HoaiThuong\AppData\Local\Microsoft\Windows\INetCache\Content.Word\efekt.jpg

t.j. prúdové spracovanie je k násobne rýchlejšie ako sekvenčné spracovanie, pričom spracovanie jedinej úlohy, rozloženej na k vykonávacích fáz, môže pri prúdovom režime trvať dlhšiu dobu, ako pri sekvenčnom režime (Ti ≤ kTk ).

**Procesory VLIW** (Very Long Instruction Word) Predstavuje v súčasnosti kombináciu dvoch koncepcií spracovania inštrukcií, ktorými sú: ν horizontálne mikroprogramovanie (východisko pôvodnej architektúry procesorov VLIW odvodené z architektúr CISC), ν superskalárne spracovanie inštrukcií (nový prístup uplatňovaný v architektúrach procesorov VLIW). ⁄ Uplatnenie v signálových procesoroch, multimediálnych procesoroch a vo formátoch inštrukcií EPICS (Explicitly Parallel Instruction Computing) uplatňovaných firmami Intel a HP v architektúrach procesorov (Merced).

**Prúdové vektorové procesory** Prúdový vektorový procesor je špecializovaný procesor určený na vykonávanie vektorových operácií. Vektorová inštrukcia zahrňuje veľký počet operandov (prvky vektora), ktoré sa spracúvajú v multiprúdovom prostredí vektorového procesora. Prúdové spracovanie vektorových inštrukcií, pri ktorom sa uplatňuje paralelizmus rozvinutých operandov, môže byť použité pri ľubovoľnom skalárnom, superprúdovom alebo superprúdovom spracovaní inštrukcií programu, ktorý je vektorizovateľný.

**9.1. Pamäťový podsystém. Definícia, charakteristika a klasifikácia pamätí. Výkonnosť pamäťového podsystému. Cena pamäťového podsystému. Procesorové pamäte: pamäte s nemeniteľným obsahom, zápisníková pamäť a zásobníková pamäť: Vyrovnávacia pamäť.**

**Definicia,CHAR-**

* Vzhľadom na nevyhnutnosť vyriešenia konfliktu medzi požiadavkami na hodnoty základných parametrov pamäte, ktorými sú kapacita, prístupová doba a cena, používa sa niekoľko druhov pamätí s rozličnými hodnotami uvedených parametrov.
* Zo systémového hľadiska predstavujú tieto pamäte jediný celok, vytvárajúc **pamäťový podsystém ČP.**

**Klasifikácia pamätí**

* Podľa spôsobu sprístupňovania informácií pri ich čítaní/zápise z/do pamäte (pri aktivácii pamäte)
  + Pamäte s postupným prístupom (SAM - Sequential Access Memory). ν Pamäte s priamym prístupom (DAM - Direct Access Memory), s komerčným označením: ν RAM (Random Access Memory) ν RWM (Read Write Memory) ν ROM (Read Only Memory) ν PROM (Programmable ROM) ν REPROM (Reprogrammable PROM) ν Pamäte s asociatívnym prístupom (CAM - Content Access Memory), ktoré sprístupňujú ich obsah podľa výberového kľúča.
* Na základe pamätanej informácie vo vzťahu k pamäťovému nosiču
  + statické pamäte, na báze prvkov z dvoma stabilnými stavmi (SRAM – Static DRAM); dynamické pamäte, na báze časovo ohraničenej reprezentácie bitových hodnôt, ktoré je nutné v pravidelných intervaloch obnovovať (DRAM - Dynamic RAM).

**Výkonnosť pamäťového podsystému** Výkonnosť hierarchicky usporiadaného pamäťového podsystému vyjadruje efektívny prístupový čas Te do jeho ľubovoľnej úrovne. Te závisí na hodnote ν koeficientu úspešnosti a frekvencií prístupu k jednotlivým úrovniam pamäťového podsystému

Pre dvojúrovňový pamäťový podsystém (VP + HP)



kde TV a TH sú príslušné prístupové časy VP a HP a N je počet prístupov do HP, resp. VP.

**Celková cena (C)**

kde Ci a ci sú cena a kapacita i-tej úrovne pamäťového podsystému a platí



C1>C2>…>Cn

c1<c2<….<cn

**Procesorové pamäte** predstavujú rôzne typy registrových pamätí; ⁄ z funkčného hľadiska majú rôznu štruktúrnu organizáciu a technologickú realizáciu.

**Pamäte s nemeniteľným obsahom** Sú reprezentované prostredníctvom pamätí typu ROM, PLA, GA a pod. vo funkcii ν generátora logických funkcií (pamäť pravdivostnej tabuľky), ν generátora kódu (prevodník kódu), ν generátora znakov (pamäť bodov matice rastra znakov), ν generátora funkcií (pamäť tabuliek hodnôt funkcie), ν špecializovanej ALJ (pamäť výsledkov operácií), ν riadiacej pamäte (pamäť riadiacich signálov v riadiacich jednotkách a mikroprogramových automatoch).

**Zápisníková pamäť (ZP)**

Charakteristickou operáciou je prenos údajov medzi jej registrami, určenými adresou a inštrukciou definovanými registrami ALJ, resp.procesora, čo sa formálne vyjadrí v tvare: **R) A := ZP [ADR]**pri čítaní údajov zo ZP a **W) ZP [ADR] := A**

pri zápise údajov do ZP, kde R a W riadiace signály,

inicializujúce registrovú operáciu čítania a zápisu.

**Zásobníková pamäť (ZS)** ⁄ Uskutočňuje zápis a čítanie v inverznom frontovom režime LIFO (Last In First Out). ⁄ Typické aplikácie ZS vyplývajú z bezadresového sprístupňovania jej položiek, ktoré nachádza významné uplatnenie najmä pri: ν výpočte aritmetických výrazov zobrazených v postfixnom bezzátvorkovom tvare (pre účely syntaktickej analýzy a konštrukcie prekladačov); ν organizovanie prerušenia a volanie podprogramov ν odpamätanie návratových adries a stavov prerušeného programu pri prechode do príslušného obslužného programu resp. podprogramu, ν ukladanie globálnych a lokálnych parametrov procedúr.

**Vyrovnávacia pamäť (VP)** ⁄ Komponenty VP ν Adresár VP (AVP), v ktorom sú uložené adresy Ai kópií obsahov pamäťových buniek hlavnej pamäte (HP), ν Údajové pole VP (DVP), v ktorom sú uložené kópie obsahov najčastejšie používaných pamäťových buniek hlavnej pamäte HP[Ai ], kde i = 0,1, ... ,n-1. ⁄ Aktivácia VP sa uskutočňuje na základe asociatívneho prístupu, pri ktorom sa konfrontuje prítomnosť adresovaného slova porovnaním jeho adresy A s jej kópiou Aα v AVP: L | A {A }

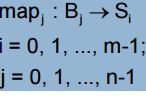
**9.2. Pamäťový podsystém. Mapovanie vyrovnávacej pamäte. Modely adresovania vyrovnávacej pamäte. Vyrovnávacia pamäť s priamym sprístupňovaním fyzickej adresy. Vyrovnávacia pamäť s obmedzeným stupňom asociativity. Skupinovo asociatívna pamäť. Vyrovnávacia pamäť so sektorovou organizáciou.**

**Mapovanie vyrovnávacej pamäte** Sprístupňovanie vyrovnávacej pamäte sa uskutočňuje prostredníctvom

* Fyzickej adresy- VAX 8600, Intel i486, MIPS R3000.
* Virtuálnej adresy Intel i 860.

**VP s priamym sprístupňovaním fyzickej adresy**

Pre n >> m je mapovanie HP do VP definované zobrazením



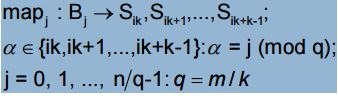
n - počet pamäťových blokov B v HP m - počet stránok S vo VP mapj - mapovacia funkcia Bj - j-ty pamäťový blok s veľkosťou k slov v HP Si - i-ty pamäťový blok s veľkosťou k slov vo VP

**VP s obmedzeným stupňom asociativity**⁄ Prístup do asociatívnej VP prostredníctvom VA je príliš nákladný, pretože VA >> FA, v dôsledku čoho príznakový kľúč asociatívneho výberu má veľký rozmer. Efektívne riešenie poskytuje koncepcia VP s adresovo-asociatívným prístupom, ktorá umožňuje obmedziť stupeň asociativity definovaný koeficientom k.

**Skupinovo asociatívna VP (stupeň asociativity k ≠ 1)**

Mapovanie HP do VP je definované zobrazením

k - stupeň asociativity m - počet stránok S vo VP mapj - mapovacia funkcia Bj - j-ty pamäťový blok s veľkosťou k slov v HP Sik - ik-ta stránka vo VP



**VP so sektorovou organizáciou**

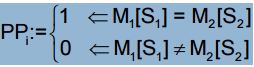
* Umiestnenie blokov predstavuje zovšeobecnenie predchádzajúcich schém sprístupňovania VP.
* VP a HP sú rozdelené na sektory s vopred definovaným rozmerom, ktoré môžu byť umiestnené v ľubovolnom sektore vo VP.
* Požiadavky na aktiváciu pamäte sú formulované na sprístupňovanie blokov a nie sektorov.
  + Ak vyhľadanie rámu sektora je úspešné, je úspešné i vyhľadanie bloku v ráme sektora.
  + Pri takejto organizácii vyhľadávania blokov je dôležité určiť, či blok v danom sektore je platný alebo platný nie je. Za týmto účelom každý rám bloku má pridaný bit platnosti bloku v sprístupňovanom ráme sektora, ktorý indikuje, či blok je platný alebo neplatný.

**9.3. Pamäťový podsystém. Modulárna organizácia hlavnej pamäte. Konzistencia údajov v pamäti. Prístupy organizovania modulárnej hlavnej pamäte. Systém virtuálnej pamäte. Stránková organizácia pamäte. Tabuľka stránok. Metódy vyraďovania stránok. Stratégie LRU vyraďovania stránkových rámov.**

**Modulárna organizácia hlavnej pamäte** ⁄ Adresovanie pamäťových buniek sa uskutočňuje prostredníctvom adresy (ADR), ktorá sa skladá z dvoch segmentov M a D. ⁄ Pri sprístupňovaní pamäte jej modulárna organizácia umožňuje prekrývať adresovanie nasledujúceho modulu s vlastným procesom čítania/zápisu z/do predchádzajúceho modulu.

**Konzistencia údajov**

Zhoda údajov medzi dvoma pamäťovými úrovňami M1 a M2 sa zabezpečuje prostredníctvom príznaku platnosti údajov PP



* Metódy dosiahnutia zhody údajov ν súčasný zápis údajov do M1 a M2 , ν pri zápise údaju do M1 na miesto určené k vyradeniu sa príslušný stránkový rám vždy vyradí do M2 , ν pri zápise údaju do M1 na miesto určené k vyradeniu sa príslušný stránkový rám vyradí do M2 iba vtedy, keď PP indikuje nezhodu medzi originálom v M2 a jeho kópiou v M1 .

**Tri prístupy organizovania modulárnej HP**

* lineárny prístup k súvislej oblasti HP v rámci modulov, ADR(n + m..1) = M (m..1) . D (n..1)
* prekrývaný prístup k súvislej oblasti HP distribuovanej medzi jednotlivými modulmi, ADR(n + m..1) = D (n..1) . M (m..1)
* prekrývaný prístup k súvislej oblasti HP distribuovanej medzi jednotlivými modulmi organizovanými v skupinách ν zvyšuje sa odolnosť proti poruchám v moduloch, v dôsledku izolovanej činnosti jednotlivých pamäťových bánk.

ADR(n + m..1) = M1 (m1..1) . D (n..1) . M2 (m1..1)

**Systém virtuálnej pamäte**

* Fyzická pamäť (FP) ν Je určená na umiestnenie vykonávaných programov ν Označuje sa ako hlavná pamäť ν Je kapacitne obmedzená
* Virtuálna pamäť (VP) ν Rozširuje adresovateľný priestor ν Označuje sa ako sekundárna pamäť

**Stránková organizácia pamäte** Mapovanie adries sa uskutočňuje prostredníctvom stránkovej organizácie pamäte ν Skupiny susedných 2k pamäťových buniek sa zlučujú do pamäťového priestoru, ktorý sa nazýva: ν stránka, v prípade VP, ν stránkový rám, v prípade FP.

**Tabuľka stránok** Jej funkcia vychádza z potreby priebežne registrovať prítomnosť kópií stránok z priestoru VP ako stránkových rámov v priestore FP. ⁄ Každej stránke S = 0,1, ... , M je v tabuľke stránok priradený riadok, v ktorom: Ri - adresa príslušného stránkového rámu, PDi - príznak dostupnosti stránky, PAi - príznak aktivity stránky, PZi - príznak zmeny (platnosti) stránky.

**Metódy vyraďovania stránok** Ak vo FP (v HP) nie je umiestnená stránka z adresovaného priestoru VP

VP[TLB[ ].S] := HP[TLB[ ].R] - vyradenie stránkového rámu z HP

HP[TLB[ ].R] := VP[VA] - zápis kópie stránky do HP

TLB[ ].S := VA - úprava TLB

kde γ je pozícia zápisu v TLB, v ktorej je uložená informácia o stránkovom ráme určenom na vyradenie z HP

**Stratégie LRU** ⁄ Vyradí sa najdlhšie nepoužívaný stránkový rám, k čomu sa využíva počítadlo aktivácií PCi nastavované príznakom aktivity Pi , kde i = 0,1, ... , m je číslo stránkového rámu.



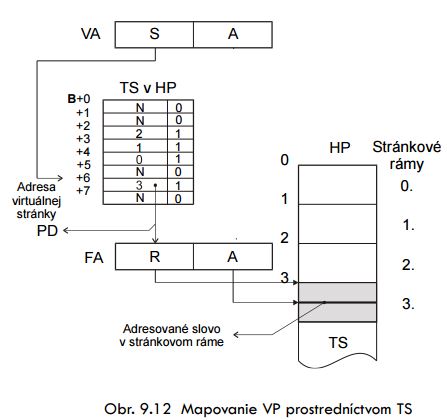
Vyradí sa stránkový rám, ktorého PC je nastavené na najvyššiu hodnotu.

**9.4. Pamäťový podsystém. Základné mechanizmy riadenia logického adresného priestoru. Mapovanie virtuálnej pamäte. Mapovanie prostredníctvom tabuľky stránok (+obr.). Mapovanie prostredníctvom pamäti preložených adries (+obr.). Segmentová organizácia pamäte (+obr.). Metódy vyraďovania stránok. Stratégia FIFO vyraďovania stránkových rámov.**

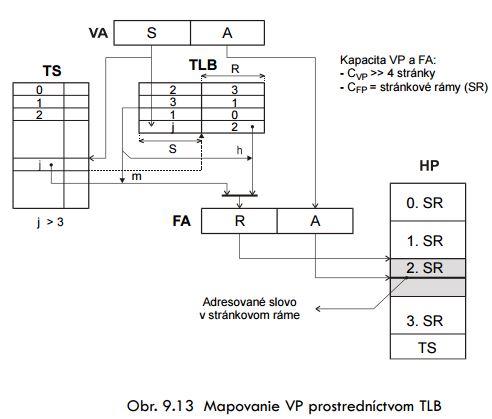
**Základné mechanizmy riadenia logického adresného priestoru**

* mapovanie pamäte -prevod virtuálných adries na fyzické adresy.
* premiestňovanie stránok medzi FP a VP.

**Mapovanie virtuálnej pamäte** ⁄ Mapovanie prostredníctvom tabuľky stránok TS (PT - Page Tables) ⁄ Mapovanie prostredníctvom pamäti preložených stránok PPS (TLB - Translation Lookaside Buffer), ⁄ Mapovanie prostredníctvom registrovej pamäte RP.

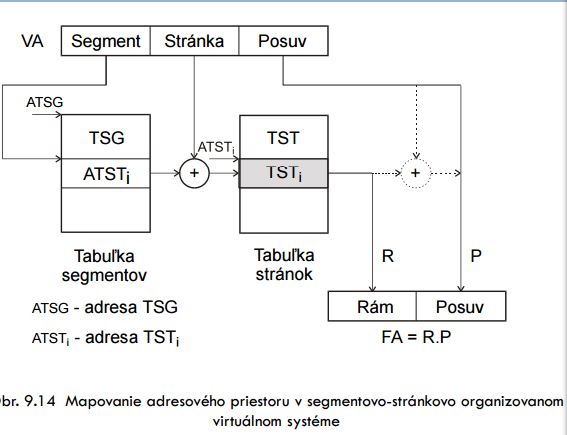


**Mapovanie prostredníctvom TS** ⁄ Ak B je pevne definovaná bázová adresa TS uloženej v HP, tak výsledná adresa (R) stránkového rámu, zodpovedajúceho stránke s virtuálnou adresou (S) je ⁄ Nedostatok ν pomalé mapovanie v dôsledku dvojnásobného prístupu do HP pri každom výpočte FA.



**Mapovanie prostredníctvom TLB** Sú v nej uložené adresy najčastejšie používaných stránok v práve prebiehajúcom intervale spracovania programu. ⁄ Položka v TLB (asociatívna pamäť) má formát TLB= <S><R> Pri asociatívnom prehľadávaní TLB sú definované nasledujúce situácie ν prítomnosť stránky v TLB (h - hit), ν výpadok stránky v TLB (m - miss). h | S TLB, m | S TLB ∈ ∉ TLB = S R

**Segmentová organizácia pamäte** ⁄ Segmentová organizácia pamäte vyplýva z požiadavky vytvoriť v pamäťovom podsystéme priestor premenlivej dľžky. ν Môže byť uložený od ľubovolnej adresy virtuálneho priestoru. ⁄ Spravidla sa používa v kombinácii so stránkovou organizáciou pamäte. ν Segmentovo-stránková (viacúrovňová) organizácia virtuálnej pamäte



**Metódy vyraďovania stránok** Ak vo FP (v HP) nie je umiestnená stránka z adresovaného priestoru VP

VP[TLB[ ].S] := HP[TLB[ ].R] - vyradenie stránkového rámu z HP

HP[TLB[ ].R] := VP[VA] - zápis kópie stránky do HP

TLB[ ].S := VA - úprava TLB

kde γ je pozícia zápisu v TLB, v ktorej je uložená informácia o stránkovom ráme určenom na vyradenie z HP

**Stratégie FIFO** ⁄ Vyradí sa stránkový rám, ktorý bol najdlhšie umiestnený v HP (resp. v cache) ⁄ Vyradí sa stránkový rám, ktorého PC je nastavené na najvyššiu hodnotu.

**10.1. Vonkajšie komunikácie procesora a riadenie vstupno/výstupných operácií. Charakteristika prenosu informácií. Parametre zbernice. Spôsob obsluhy V/V zariadení. Prístupy riadenia V/V operácií: priame riadenie programom, riadenie s prerušením, priamy prístup do pamäte (samoštúdium).**

**Prenos informácií**

* Komunikácia medzi zdrojom a príjemcom správ sa uskutočňuje na základe vopred špecifikovaných syntaktických a sémantických pravidiel a prostriedkov komunikácie.
* Podľa vzdialenosti- Malé vzdialenosti -nevyžaduje osobitné prostriedky na realizáciu komunikácie Stredné vzdialenosti ν vyžaduje prostriedky na ochranu informácií pred ich znehodnotením Veľké vzdialenosti ν vyžaduje ďalšie komunikačné prostriedky na vysokej organizačnej a technickej úrovni

**Parametre zbernice**

* Šírka zbernice - Počet informačných jednotiek (bitov) údajovej zbernice prenášaných súčasne
* Prenosová rýchlosť Počet bitov prenesených za jednotku času , Jednotka prenosovej rýchlosti je 1 bit/s ,Dvojstavový prenos – modulačná rýchlosť (1 Bd)
* Prenosový výkon Počet bitov prenesených za jednotku času ,Jednotka prenosovej rýchlosti je 1 bit/s

**Spôsob obsluhy V/V zariadení (VVZ**) Zabezpečuje ν zbernicová jednotka (ZBJ) ≈ riadiaca jednotka zbernice ν V/V koprocesor ν V/V procesor ν arbiter zbernice (AZB) – rieši aj otázku priorit

**Prístupy riadenia V/V operácií** ⁄Priame riadenie programom ⁄ Riadenie prerušením ⁄ Priamy prístup do pamäte

**Priame riadenie programom** Prenos sa iniciuje samostatnou, v programe definovanou vstupnou, alebo výstupnou inštrukciou. Riadenie vykonania inštrukcie uskutočňuje procesor.

**Riadenie s prerušením** Po naštartovaní V/V operácie procesor zabezpečuje realizáciu ďalších inštrukcií programu až do okamihu, keď VVZ skončí začatú operáciu. ⁄ V závislostí od povahy a výskytu týchto situácií sa rozlišujú ν vonkajšie prerušenia, ktorých príčiny nezávisia od práve prebiehajúceho programu (majú náhodný charakter), ν vnútorné prerušenia, ktorých príčiny sú cieľavedome vyvolávané práve prebiehajúcim programom (možno ich predvídať).

**Priamy prístup do pamäte (DMA**) procesor len inicializuje informácie o prenose (odkiaľ, kam, koľko, a robí iné ) DMA radič zabezpečuje prenos po zbernici priamo medzi V/V zariadením a pamäťou bez účasti procesora DMA radič generuje prerušenie po ukončení prenosu procesor prečíta stavovú informáciu

**10.2. Vonkajšie komunikácie procesora a riadenie vstupno/výstupných operácií. Komunikácia CPU – Hlavná pamäť. Komunikácia CPU – vstupno/výstupné zariadenie. Prerušovací systém.**

**Komunikácia procesor ↔ HP**

* Riadiace činnosti ν pridelenie komunikačnej linky, ν adresovanie HP, ν smer prenosu informácií, ν synchronizácia prenosu informácií.
* Komunikačné linky medzi procesorom a HP sú obyčajne organizované ako zbernicový systém a preto časová postupnosť signálov, ktoré riadia uvedené činnosti sa nazýva zbernicový cyklus.

**Komunikácia procesor ↔ V/V zariadenie**

* pri komunikácii procesora s pomalými VVZ sa jeho činnosť synchronizuje prostredníctvom signálu pripravenosti (READY) a procesor sa uvedie do stavu čakania.
* Zbernicový cyklus INTA (Interrupt Acknowledge) ν Podobá sa zbernicovému cyklu čítania. ν Potvrdenie požiadavky prerušenia sa skladá z dvoch cyklov ν V prvom cykle si procesor rezervuje zbernicu blokovacím signálom LOCK. ν V druhom cykle procesor po údajovej zbernici prijíma z príslušného VVZ informáciu (vektor prerušenia) o type prerušenia.

**Prerušovací systém**

* Vo všeobecnosti je proces prerušenia charakterizovaný vykonaním nasledujúcich činností
  + generovanie signálov prerušenia, ktoré reprezentujú príslušné požiadavky na prerušenie (PP) činnosti procesora; ν identifikácia PP; ν riešenie konfliktnej situácie pri vzniku niekoľkých PP (prioritný výber); ν získanie adresy príslušného obslužného programu; ν odpamätanie obsahu PC registra (návratovej adresy) a registrov s údajmi, potrebnými na pokračovanie hlavného programu po skončení obslužného programu; ν vykonanie obslužného programu; ν návrat do prerušeného programu.

**10.3. Vonkajšie komunikácie procesora a riadenie vstupno/výstupných operácií. Systémové rozhrania (ISA, ATA, PCI). Sériové rozhrania (SCI, SPI, RS-232, USB). Paralelné rozhrania (SCSI, CENTRONICS).**

ISA (Industry Standard Architecture)

* je štandard pre zbernice pre IBM kompatibilné počítače. Rozširovala architektúru [XT](https://sk.wikipedia.org/w/index.php?title=XT&action=edit&redlink=1) zbernice na [16 bitov](https://sk.wikipedia.org/wiki/16-bitov%C3%BD). Zbernica ISA je rozšírením natívnej zbernice procesora 8086 (elektricky posilnená adresná a dátová zbernica) o signály prerušovacieho kontroléra (PIC) a DMA kontroléra. a je určená na pripájanie periférnych kariet k [základnej doske](https://sk.wikipedia.org/wiki/Z%C3%A1kladn%C3%A1_doska).

PCI (Peripheral Component Interconnect)

* štandard pre [zbernicu](https://sk.wikipedia.org/wiki/Zbernica) počítača k pripojeniu [periférnych zariadení](https://sk.wikipedia.org/wiki/Perif%C3%A9rne_zariadenie) k [matičnej doske](https://sk.wikipedia.org/wiki/Mati%C4%8Dn%C3%A1_doska). Tieto zariadenia môžu byť: integrované obvody, prídavné karty

ATA –

* zbernica navrhnutá na prenos dát z a na pevný disk.(Advanced Technology Attachment)

**???**

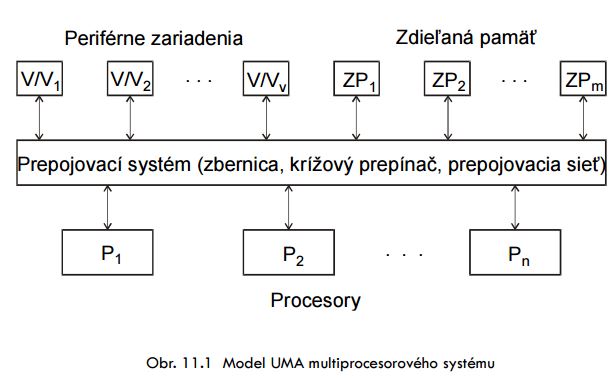
**11.1. Princípy a základné koncepcie paralelných počítačových architektúr. Charakteristika architektonických riešenie súčasných a perspektívnych mikroprocesorov. Multiprocesorové systémy : UMA (+obr.). Multiprocesorové systémy : NUMA (+obr.). Teoretické modely počítačových architektúr (str. 19).**

**Charakteristika architektonických riešenie súčasných a perspektívnych mikroprocesorov** ⁄

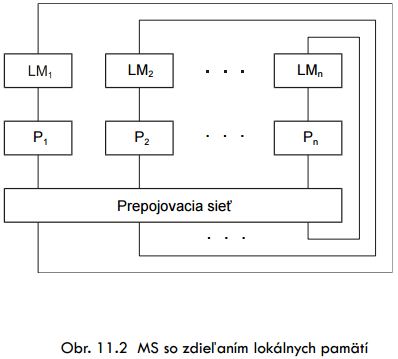
* Zvýrazňovanie a odhaľovanie paralelizmu v architektúrach von Neumannových mikroprocesorov (paralelizmus na úrovni inštrukcií). ⁄ Zavádza sa prúdový inštrukčný paralelizmus implementovaný vo viacerých inštrukčných prúdoch (vláknach). Tento paralelizmus označovaný ako vláknový paralelizmus umožňuje paralelne spracúvať inštrukcie v menších vláknach, čím sa vytvára modulárne prostredie na paralelné vykonávanie inštrukcií programu.
* Komprimovanie logických prvkov na ploche čipu umožňuje zrýchliť medzičipové komunikácie. ⁄ Vysoká integrácia prvkov na ploche čipu vytvára predpoklady na integrovanie viac ako jedného mikroprocesora na čipe.
* Koncepcia rozmiestnenia komponentov na ploche čipu zameraných na: ν modulárny návrh štruktúrnej organizácie funkčných prvkov univerzálneho mikroprocesora na čipe, ktorý spracúva jednoduché inštrukčné prúdy so zložitým inštrukčným súborom, ν návrh jednoduchých mikroprocesorov integrovaných do multiprocesorového čipu, ν integrovanie jednoduchého mikroprocesora do pamäťového čipu za účelom vytvorenia tesnejšej väzby pre komunikácie typu procesor-pamäť.

**Multiprocesorové systémy** ⁄ Skupina vzájomne prepojených počítačov, ktoré riešia jednu zložitú úlohu alebo sa nezávisle podieľajú na spracovaní viacerých programov. ν MS so zdieľanou (centralizovanou) pamäťou - multiprocesory, ν MS s distribuovanou (lokálnou) pamäťou - multipočítače.

**Modely UMA** Fyzická pamäť je rovnomerne využívaná všetkými procesormi. ⁄ Prístupový čas k jednotlivým slovám pamäti je rovnaký (jednotlivé procesory môžu mať k dispozícii privátne vyrovnávacie pamäte typu cache). ⁄ Využíva sa na zrýchlenie vykonania jediného rozsiahleho programu pre časovo náročné aplikácie (aplikácie reálneho času) alebo na multiprogramový režim práce.



Na koordináciu paralelného spracovania úloh (synchronizácia a komunikácia medzi procesormi) sa využívajú zdieľané premenné v spoločnej pamäti. ⁄ V závislosti od využívania pamäte všetkými procesormi alebo iba ich časťou sa rozlišujú: ν symetrické modely UMA, ν nesymetrické modely UMA.



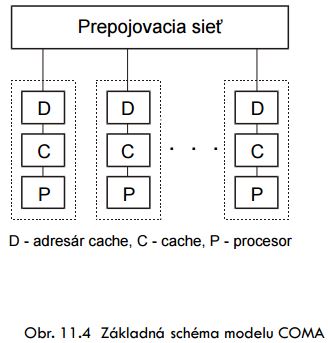
**Modely NUMA** ⁄ Zdieľaná pamäť je fyzicky distribuovaná procesorom prostredníctvom lokálnych pamäti (LM). ⁄ Prístupový čas do lokálnych pamätí je rôzny. ⁄ Množina lokálnych pamätí tvorí globálny adresový priestor jednotlivých procesorov. ⁄ Delíme ich na ν MS so zdieľaním lokálnych pamätí, vyznačujúcich sa rýchlym prístupom do vlastnej lokálnej pamäte, ν Hierarchicky štruktúrované MS so skupinovou („klastrovou“) organizáciou, ktoré vytvárajú základ pre tvorbu kombinovaných paralelných štruktúr.

**Teoretické modely** ⁄ Na základe modelov UMA, NUMA a COMA konkrétnych architektúr sa vytvárajú teoretické modely paralelných počítačových systémov, ktoré sa používajú pri návrhu algoritmov a obvodov VLSI: ν modely PRAM (Parallel Random-Access Machine) na získanie teoretickej hodnoty ohraničenia výkonnosti, ν modely VLSI (Very Large-Scale Integration) na ocenenie obvodovej zložitosti na ploche čipu. ♦ Nové trendy ⁄ Biopočítače ⁄ Kvantové počítače

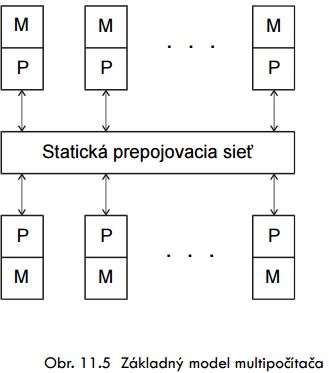
**11.2. Princípy a základné koncepcie paralelných počítačových architektúr. Charakteristika architektonických riešenie súčasných a perspektívnych mikroprocesorov. Multiprocesorové systémy : COMA (+obr.). Multipočítače (+obr.). Teoretické modely počítačových architektúr.**

**Charakteristika architektonických riešenie súčasných a perspektívnych mikroprocesorov** ⁄

* Zvýrazňovanie a odhaľovanie paralelizmu v architektúrach von Neumannových mikroprocesorov (paralelizmus na úrovni inštrukcií). ⁄ Zavádza sa prúdový inštrukčný paralelizmus implementovaný vo viacerých inštrukčných prúdoch (vláknach). Tento paralelizmus označovaný ako vláknový paralelizmus umožňuje paralelne spracúvať inštrukcie v menších vláknach, čím sa vytvára modulárne prostredie na paralelné vykonávanie inštrukcií programu.
* Komprimovanie logických prvkov na ploche čipu umožňuje zrýchliť medzičipové komunikácie. ⁄ Vysoká integrácia prvkov na ploche čipu vytvára predpoklady na integrovanie viac ako jedného mikroprocesora na čipe.
* Koncepcia rozmiestnenia komponentov na ploche čipu zameraných na: ν modulárny návrh štruktúrnej organizácie funkčných prvkov univerzálneho mikroprocesora na čipe, ktorý spracúva jednoduché inštrukčné prúdy so zložitým inštrukčným súborom, ν návrh jednoduchých mikroprocesorov integrovaných do multiprocesorového čipu, ν integrovanie jednoduchého mikroprocesora do pamäťového čipu za účelom vytvorenia tesnejšej väzby pre komunikácie typu procesor-pamäť.



**Modely COMA** ⁄ Predstavuje špeciálny prípad modelu NUMA, v ktorom distribuovaná hlavná pamäť je konvertovaná do modulov vyrovnávacích pamätí typu "cache„. ν Moduly cache reprezentujú globálny adresový priestor; ν Prístup do vzdialených cache sa uskutočňuje prostredníctvom adresárov cache; ⁄ Prostredníctvom vhodného typu prepojovacej siete sa sprístupňujú údajové bloky príslušnému procesoru. ⁄ Prostredníctvom smerovačov je možné vytvárať hierarchické štruktúry ν Má význam kvôli škálovateľnosti (scalability).



**Multipočítače** Disponujú s distribuovanou pamäťou. ⁄ Sú označované aj ako voľne viazané multiprocesory. ⁄ Pozostávajú z uzlových počítačov prepojených pomocou statickej prepojovacej siete prenosu správ, ⁄ Lokálne pamäte sú privátne, t.j. prístupné iba lokálnym procesorom. ⁄ Medziprocesorové komunikácie sa uskutočňujú prenosom správ.

**Teoretické modely** ⁄ Na základe modelov UMA, NUMA a COMA konkrétnych architektúr sa vytvárajú teoretické modely paralelných počítačových systémov, ktoré sa používajú pri návrhu algoritmov a obvodov VLSI: ν modely PRAM (Parallel Random-Access Machine) na získanie teoretickej hodnoty ohraničenia výkonnosti, ν modely VLSI (Very Large-Scale Integration) na ocenenie obvodovej zložitosti na ploche čipu. ♦ Nové trendy ⁄ Biopočítače ⁄ Kvantové počítače