

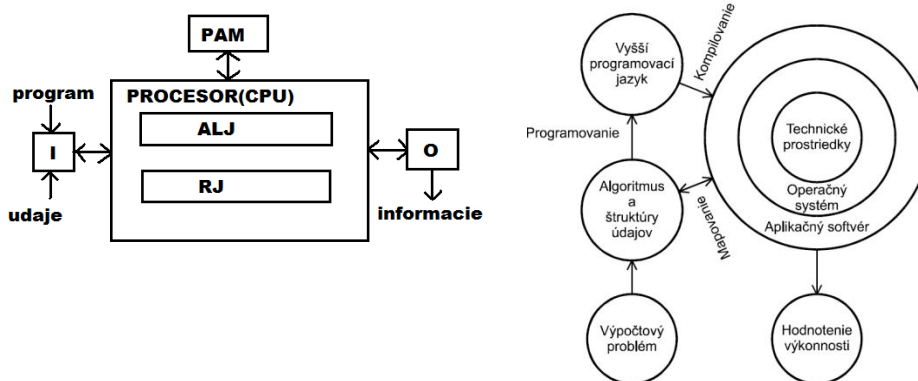
1.1. Princípy a základné koncepcie počítačových systémov. Definícia číslicového počítača. Architektúra PS (ISA, IA). Koncepcia programového riadenia číslicového počítača: von Neumannova architektúra číslicového počítača.

Číslicový počítač 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).

Von Neumanova architektúra - VN počítač pracuje na princípe programového riadenia.

Vlastnosti: 2-ková číslicová sústava, algoritmus, nezávislosť programu (spôsob kódovania hodnôt), nezávislosť štruktúrnej organizácie počítača (pamäť, aritmetická jednotka, riadiaca jednotka, I/O jednotka) od riešenia

Hlavné časti: pamäť, riadiaca jednotka, aritmeticko-logická jednotka, vstupná a výstupná jednotka



Počítačový systém je systém vytvorený na báze 1 alebo viacerých ČP.

Architektúra PS ISA (inštrukčno orientovaná arch.)

požiadavky programových prostriedkov vrátane techniky programovania úloh
kódy operácií, adresovacie módy, špecifikácia registrov, virtuálna pamäť

Architektúra PS IA (implementačno orientovaná arch.)

vnútorná organizácia jeho technických prostriedkov

CPU, fyzická pamäť, zbernicové systémy, mikrokód, prúdové funkčné jednotky, jednotky na paralelné spracovanie

1.2. Princípy a základné koncepcie počítačových systémov. Definícia číslicového počítača. Vývoj a klasifikácia číslicových počítačov: charakteristické črty rozvoja výpočtovej techniky. Generácie vývoja počítačov.

Prvá generácia (1945 - 1950):

- prvková základňa: elektrónky, pamäť: magnetická bubnová (nerozlišuje sa vnútorná a vonkajšia pamäť),
- jediná centrálna procesorová jednotka - procesor (CPU - Central Processor Unit) v dekomponovanom tvare na riadiacu jednotku a aritmetickú jednotku),
- aritmetika v pevnej rádovej čiarky, využívajúca programové počítadlo, inštrukcie vetvenia a akumulátor,
- účasť CPU na všetkých pamäťových a V/V operáciách,
- programovanie v strojovom jazyku alebo v asembléri,
- individuálny prístup používateľa k počítaču.

Druhá generácia (50-te roky):

- prvková základňa: tranzistory, operačná pamäť: feritová
- dvojúrovňová pamäť: vnútorná (operačná) pamäť a vonkajšia pamäť
- aritmetika v pohyblivej rádovej čiarky, multiplexovanie pamäte, indexové registre
- V/V operácie vykonáva V/V jednotka (V/V procesor)
- programovanie v jazykoch vyššej úrovne (Fortran, Algol, Cobol)
- jazyk medziregistrových prenosov na navrhovanie technických prostriedkov ČP
- vznik jednoduchších operačných systémov, začiatok uplatňovania filozofie pridelovania času a jednoduchých prerušovacích systémov, dávkový spôsob prístupu používateľa k počítaču.

Tretia generácia (od polovice 60-tych rokov do polovice 70-tych rokov):

- prvková základňa: tranzistorové mikromoduly a neskôr integrované obvody SSI a MSI)
- hlavná pamäť: feritová, polovodičová
- operačná rýchlosť: rádovo 10⁶ op/s, kapacita operačnej pamäte: rádovo 0,1 až 10 MB
- existencia rodín počítačov, kompatibilných smerom od jednoduchších k zložitejším modelom
- mikroprogramové riadenia CPU
- prúdové spracovanie informácií (pipelining)
- vyrovnávacie pamäte typu cache na vyrovnávanie rýchlosti medzi hlavnou pamäťou a CPU
- ďalší rozvoj vyšších programovacích jazykov, ich rozšírenie o jazyky simulačné
- multiprogramovanie, podporujúce viacpoužívateľský prístup prostred. prekrývania činnosti CPU a V/V jednotiek
- operačný systém na podporu virtuálneho pamäťového priestoru so zdieľaním zdrojov
- aplikácie v oblasti informačných a riadiacich systémov, pracujúcich v reálnom čase

Štvrtá generácia (1975 - 1990)

- prvková základňa: integrované obvody LSI, VLSI, mikroprocesory
- hlavná pamäť: polovodičová (dynamická), uplatňovanie nových fyzikálnych princípov (holografia, laserová technika a pod.)
- operačná rýchlosť: dosahuje rádovo 1 až 100 MFLOPS, kapacita hlavnej pamäte: 10 MB až 100 GB
- architektúry výkonných paralelných počítačových systémov (multiprocessorových a multipočítačových) so zdieľanou a distribuovanou pamäťou, rozvoj superpočítačov, výkonných personálnych počítačov, pracovných staníc a počítačových sietí, technická podpora riešenia vektorových operácií (vektorové procesory), operačné systémy, jazyky a kompilátory na podporu paralelného spracovania procesov (multiprocessing)
- vysokošpecializované a konverzačné jazyky, maximálne zjednodušujúce styk používateľa s počítačom
- podpora riešenia systémových programov technickými prostriedkami
- viacprocesorové koncepcie počítačových systémov, umožňujúcich paralelný prístup veľkého počtu používateľov (distribuované počítačové systémy)
- rozvoj lokálnych a regionálnych sieťových prostredí na prenos informácií

Piata generácia (1990 - doteraz)

- prvková základňa: integrované obvody s hustotou rádovo 10⁶ diskretných prvkov na čipe v dôsledku využívania nových technológií (technológia GaAs, medené, molekulové a optické technológie)
- ciele nových architektúr: dosiahnutie operačnej rýchlosti rádovo 10 - 100 TFLOPS v počítačových systémoch v triede personálnych počítačov kapacita hlavnej pamäte prekračuje 128 MB, kapacita pevných diskov - 10 GB a taktovacia frekvencia - 1 GHz.
- architektúry nových počítačových systémov v triede paralelných počítačov s extrémnym počtom procesorov, resp. procesorových elementov (masívne paralelné počítače), počítačov na logické programovanie, databázových a znalostných počítačov, personálnych počítačov a počítačov sieťových prostredí

- uplatňovanie princípov skalability pri návrhu architektúr počítačových systémov s dôrazom na dosiahnutie vysokých hodnôt spoľahlivostných parametrov a parametrov výkonnosť/cena
- heterogénne počítačové systémy na riešenie rozsiahlych problémov (heterogeneous processing)
- inteligentný medzistyk (porozumenie reči, obrazu a prirodzeného jazyka) a dôraz na riešenie úloh umelej inteligencie
- počítačové systémy integrovaných komunikácií so zdôraznením telekomunikačnej a multimediálnej techniky

Klasifikácia počítačov:

1. Flynnova klasifikácia:

SISD-Single Instruction Single Data

- jedna riadiaca a vykonávacia jednotka, čítanie a zápis do pamäte jednej hodnoty, Von Neumann

MISD - Multiple Instruction Single Data

- rôzne operácie nad rovnakými dátami, Data Flow architektúry

SIMD - Single Instruction Multiple Data

- jedna riadiaca jednotka, jedna cache pamäť, rovnaké operácie nad rôznymi dátami, Von Neumann

MIMD - Multiple Instruction Multiple Data

- asynchrónne pracovanie procesov, zdieľaná pamäť na výmenu dát

2. Používateľsko - aplikačná klasifikácia

- Osobné počítače (PC): personálne univerzálne výpočtové prostriedky informačnej spoločnosti.
 - Pracovné stanice (WS): problémovo orientované výpočtové prostriedky na úrovni výkonnejších PC
 - Špecializované počítače: určené spravidla na vykonávanie jednoúčelových funkcií s vysokou operačnou rýchlosťou,
 - Sálkové počítače (mainframe): výkonnejšie prostriedky (prv kategória „minipočítače“) odlišné hlavne aplikačným určením v rámci ktorého prostredníctvom terminálových a rozsiahlych sietí LAN, WAN vytvárajú prostredie pre hromadný používateľský prístup (servery).
 - Superminipočítače: skupina výkonných počítačov, viacprocesorové koncepcie
 - Minisuperpočítače a superpočítače: skupina počítačov z triedy SIMD a MIMD, hlavne kategórií s vysokou výkonnosťou, určené na riešenie úloh extrémnej zložitosti a nárokov na procesory a pamäť, zložitých numerických úloh, úloh v reálnom čase a špecializovaných úloh UI.
- ##### 3. Klasifikácia podľa použitého výpočtového modelu
- Programové riadenie(CF- Control flow): riadenie podľa inštrukcií
 - Riadenie tokom dát(DF- Data flow): riadenie prostredníctvom prúdu operandov
 - Podľa požiadavky(DD- Data demand):riadenie na základe požiadaviek inštrukcií programu na vyslanie operandov

1.3. Princípy a základné koncepcie počítačových systémov. Definícia číslicového počítača. Klasifikácia číslicových počítačov (všeobecne). Výpočtové modely (CF,DF,DD). Flynnova klasifikácia (+ obr.).

Výpočtový model

1. CF (control flow) – programové riadenie

Riadenie výpočtového procesu sa uskutočňuje prostredníctvom interpretácie sériového prúdu inštrukcií programu.

2. DF (data flow) – riadenie tokom dát

Riadenie výpočtového procesu uskutočňuje prostredníctvom prúdu operandov(údajov).

3. DD (demand driven) – podľa požiadavky

Redukčné modely. Riadenie výpočtového procesu uskutočňuje na základe požiadaviek inštrukcií programu na vyslanie operandov.

Flynnova klasifikácia

Architektúra SISD (Single Instruction Single Data Stream)

Architektúry von Neumannovho typu, v ktorých inštrukcie jediného prúdu postupne vykonávajú operácie s údajmi jediného údajového prúdu.

Architektúra MISD (Multiple Instruction Single Data Stream)

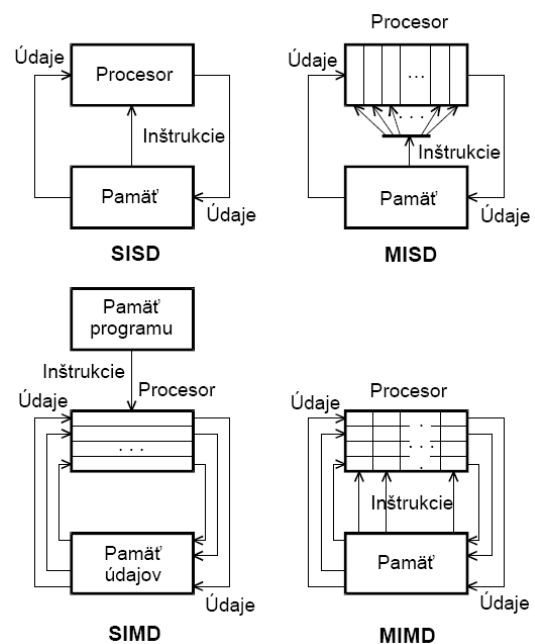
Reprezentuje tzv. *zreťazené* (prúdové) počítačové systémy, v ktorých jediný údajový prúd sa spracúva viacnásobným inštrukčným prúdom.

Architektúra SIMD (Single Instruction Multiple Data Stream)

Je architektúra, v ktorej jediný prúd inštrukcií vykonáva operácie nad mnohonásobným prúdom údajov, čo je charakteristické pre paralelné počítačové systémy.

Architektúra MIMD (Multiple Instruction Multiple Data Stream)

Je architektúra, v ktorej viacnásobný prúd inštrukcií viacerých programov alebo úloh spracúva mnohonásobný údajový prúd. Reprezentuje rôzne typy multiprocesorových a multipočítačových architektúr.



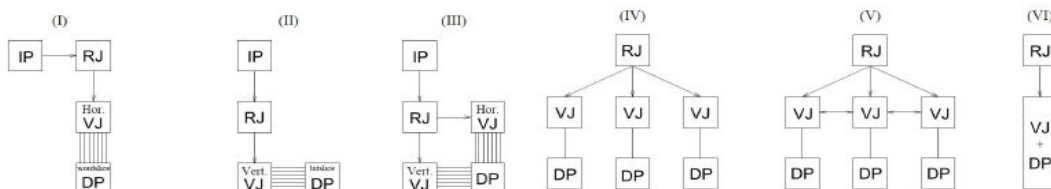
1.4. Princípy a základné koncepcie počítačových systémov. Definícia číslicového počítača. Klasifikácia číslicových počítačov (všeobecne). Kuck-ova klasifikácia. Shore-ova klasifikácia (+ obr.). Fengova klasifikácia. Erlangenova klasifikácia (+obr.).

Kuck-ova klasifikácia

SISSES (Single Instruction Scalar, Single Execution Scalar)
 SISSEA (Single Instruction Scalar, Single Execution Array)
 SIASEA (Single Instruction Array, Single Execution Array)
 MISMES (Multiple Instruction Scalar, Multiple Execution Scalar)
 MISMEA (Multiple Instruction Scalar, Multiple Execution Array)

Shore-ova klasifikácia

Machine I
 Machine II
 Machine III
 Machine IV
 Machine V
 Machine VI



Fengova klasifikácia

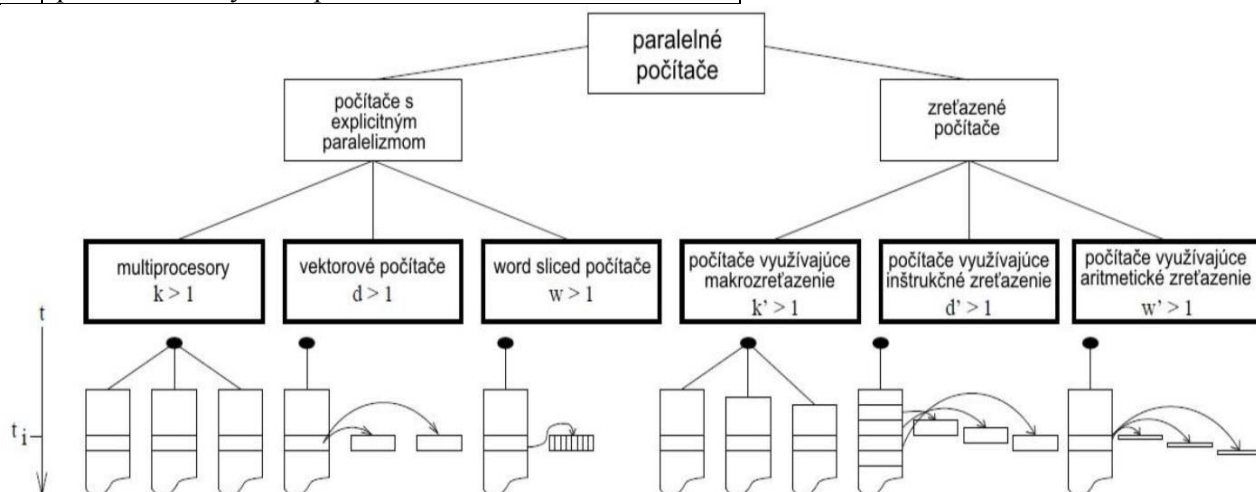
Systémy popisuje pomocou dvojice čísel. Prvé číslo reprezentuje veľkosť slova a druhé počet slov, ktoré je možné spracovať v jednom čase.

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 ($k \cdot k'$, $d \cdot d'$, $w \cdot w'$), 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 zretazených 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)

Proces návrhu PS sa z hľadiska svojej komplexnosti uskutočňuje v troch oblastiach jeho opisu (špecifikácie) a návrhu, ktorými sú:

- Správanie sa (funkcia) – inštrukčno orientovaná architektúra)
- Štruktúra – implementačno orientovaná architektúra
- Realizácia (technológia)

Špecifikácia inštrukčno orientovanej architektúry (oblasť správania sa- ISA)

- 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- IA)

- 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 je definovaný sedmicou

$\check{C}S = (U, V, P, F, G, f, g)$

kde U, V, F, G sú premenné definované vo vonkajších kanáloch ČS, určených na jeho styk s okolím, ktoré môžu byť:

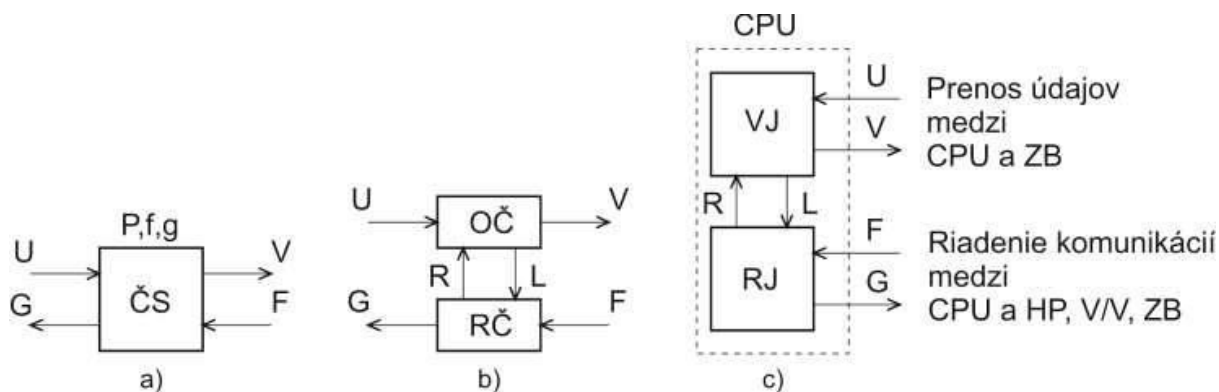
- údajové (zobrazujú spracúvané informácie),
- komunikačné (zobrazujú informácie o spracúvaných údajoch), prezentované ako riadiace, informačné a stavové premenné,

P – premenná, vyjadrujúca stav ČS,

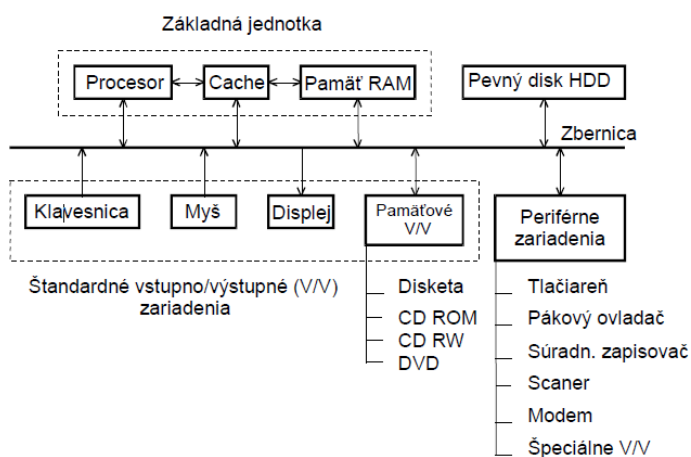
f, g – funkčné vzťahy, ktoré vyjadrujú reakciu ČS na podnety definované hodnotami nasledujúcich premenných, pôsobiacich v príslušných kanáloch:

$V = f(U, F), G = g(P)$

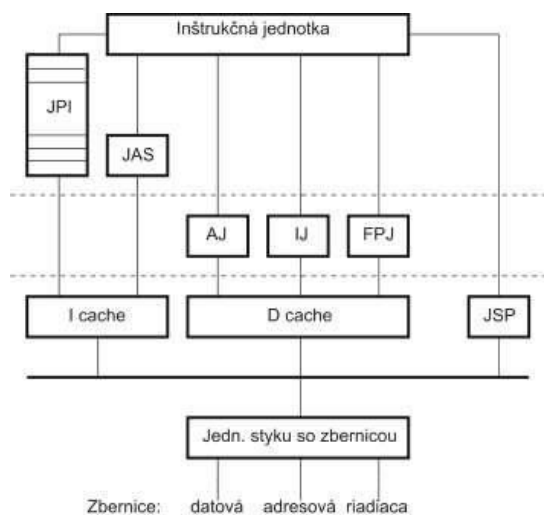
Každý ČS chápaný chápeme, ako formálny model definovaný nad reálnym číslicovým zariadením, ktorý je možné z hľadiska jeho funkcie dekomponovať na operačnú časť (OČ) a riadiacu časť (RČ).



Číslicový systém (a) a jeho dekompozícia na operačnú a riadiacu časť (b) a príklad (c) dekompozície procesora (CPU) na vykonávaciu (VJ) a riadiacu (RJ) jednotku.



Obr. 4.2 Príklad zobrazenia implementačno orientovanej architektúry na systémovej úrovni



JPI - jednotka predvýberu inštrukcií
 JAS - jednotka adresies skokov
 AJ - adresovacia jednotka
 IJ - jednotka celočíselných operácií
 FP - jednotka operácií v pohyblivej rádovej čiark
 JSP - jednotka správy pamäti

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).

George Boole

Boolova algebra – algebrická štruktúra s logickými operáciami

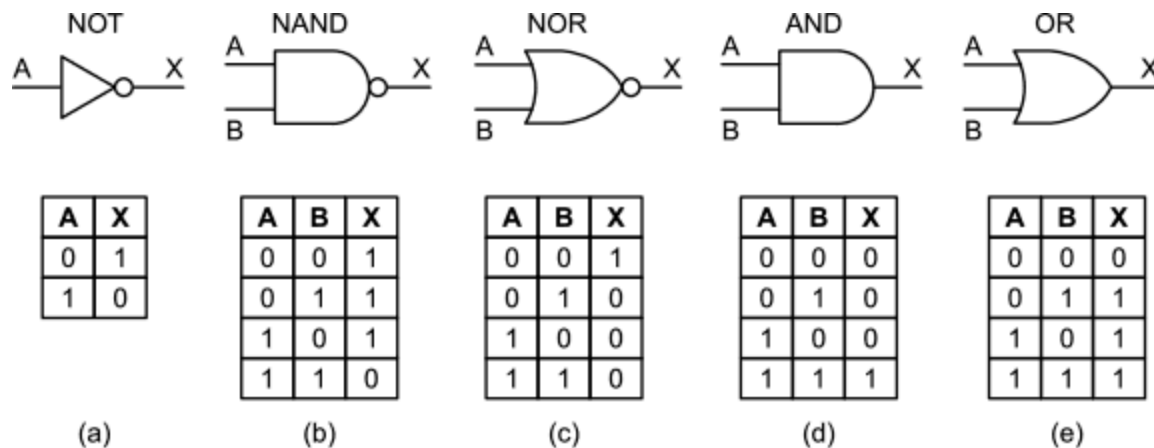
Claude Shannon – implementácia – elektromechanické relé - obvody

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

Prenos informácií: elektrický, optický, rádiový a pod.

Prenos informácií je riadený proces

- Obvody na pamätanie údajov

sú určené na ukladanie (pamätanie) údajov v tvare jedno a viacbitových slov

- preklápacie obvody, registre, pamäte

- Obvody na realizáciu predikátov

- komparátor, generátor parity, generátor priority

- Obvody na realizáciu operácií

Obvody na vykonávanie operácií nad obsahmi registrov, zberníc a iných funkčných prvkov sú najčastejšie komponentmi operačnej časti procesora ČP

Vo všeobecnosti sa tieto operácie rozdeľujú do troch skupín

- celočíselné aritmetické operácie (sčítanie, odčítanie, násobenie a delenie);
- logické operácie (logický súčet a súčin, operácie XOR, maskovanie, logické posuvy a pod.);
- operácie špeciálnej aritmetiky (desiatková aritmetika, aritmetika s pohyblivou rádovou čiarkou, špeciálne funkcie a pod.).

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

Hradlovacie obvody a budiče

Hradlo je základný logický člen mikroarchitektúry ČP, ktorého priechodnosť medzi vstupným (X) a výstupným (Y) kanálom označeným rovnako pomenovanou vstupnou a výstupnou premennou X a Y .

$$Y = \begin{cases} X \Leftarrow R = f(R_1, R_2, \dots, R_k) = 1 & X - \text{je vstupná premenná v kanále } X, \\ 0 \Leftarrow R = f(R_1, R_2, \dots, R_k) = 0 & Y - \text{výstupná premenná v kanále } Y, \end{cases}$$

R – riadiaci signál definovaný ako funkcia f riadiacich premenných R_1, R_2, \dots, R_k , t.j. $Y = X \wedge R$, resp. $Y = X \wedge 1 = X$.

Významnou aplikáciou hradla pri riadení komunikačných operácií je riadenie prenosu údajov z viacerých zdrojov do jedného cieľového zdroja, ktorým je zbernica. V súčasnosti sa pri obvodom riešení používajú trojstavové hradlá. Ak 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.

Trojrvstvové hradlá sú definované vzťahom:

$$Z = \begin{cases} DI \Leftarrow G = 0 \\ Z \text{ je v } 3.\text{stave} \Leftarrow G = 1 \end{cases} \quad \begin{array}{l} DI - \text{je údajový vstup,} \\ Z - \text{výstup (zbernica),} \\ G - \text{riadiaci trojstavový vstup.} \end{array}$$

Prevodníky kódov

Prevodník (PK) binárnej informácie vo všeobecnosti je funkčný prvok ČP, v ktorom sa uskutočňuje transformácia:

$$f: X \rightarrow Y$$

kde

f - je transformačný vzťah, definujúci typ prevodu binárnej informácie,

X, Y - vstupy (vstupné slová) a výstupy (výstupné slová) prevodníka, ktoré sú definované ako množiny n -tíc hodnôt vstupných premenných x_1, x_2, \dots, x_n a množiny m -tíc hodnôt výstupných premenných y_1, y_2, \dots, y_m .

Kodéry a dekodéry

Funkčný prvok ČP určený na prevod zložiek kódu „1 z n “ na paralelné m -rádové zložky $n = 2^m$ sa nazýva kodér (KD).

Schéma zapojenia kódera je daná nasledovne:

$$y_j = \bigvee_{x_i=1} x_i, i = 1, 2, \dots, n; j = 1, 2, \dots, m, m \leq \log_2 n \quad \begin{array}{l} y_j - \text{je výstupná premenná, ktorej hodnoty zobrazujú} \\ \text{j-ty rád kódovej zložky} \end{array}$$

Pri $n=2^n$ platí že x_i - vstupná premenná, ktorej hodnoty zobrazujú i -ty rád kódovej zložky.

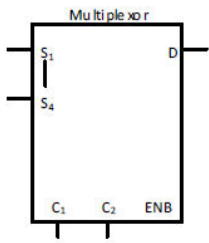
Dekodér s n vstupnými a m výstupnými premennými ($DK(n \times m)$) definovaný vyššie uvedenými vzťahmi sa nazýva úplný dekodér.

Dekodér, v ktorom medzi počtom n vstupných a m výstupných premenných platí $m < 2n - 1$, sa označuje ako dekodér neúplný.

Multiplexor

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

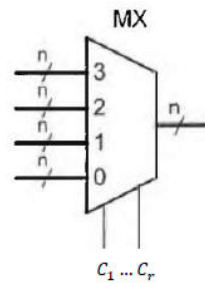
S4 na jeden údajový výstup (D).



$$D = S_i$$

$$i = \sum_{j=1}^N C_j 2^{j-1}$$

$$C_j = \{0,1\}$$



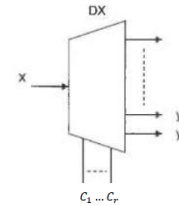
Demultiplexor

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

$$y_i = \begin{cases} X & \Leftrightarrow i = \sum_{j=1}^r C_j 2^{j-1} \\ 0 & \Leftrightarrow i \neq \sum_{j=1}^r C_j 2^{j-1} \end{cases}$$

$$i = \sum_{j=1}^r C_j 2^{j-1}$$

$$C_j \in \{0,1\}$$



riadacích vstupov C_1, C_2, \dots, C_r , kde $r = \log_2 n$.

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).

Preklápacie obvody (PO)

- sú určené na pamätanie jednobitových slov
- jeden alebo viac vstupov I_1, I_2, \dots, I_n (vstupné kanály)
- dva navzájom komplementárne výstupy Q a \bar{Q} (výstupné kanály).
- pamäťové správanie rôznych typov PO je interpretované spravidla Mooreovým automatom.
- astabilné, monostabilné, bistabilné
- synchronné vs. asynchrónne

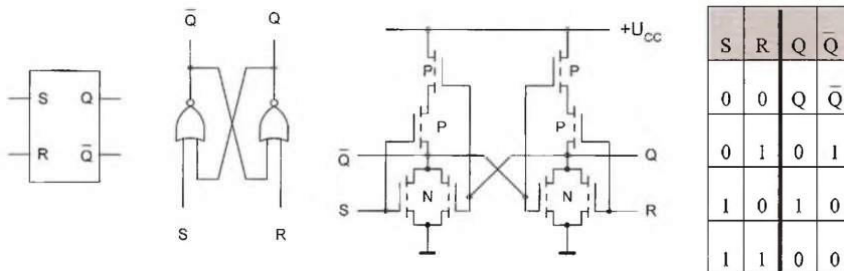
Podľa typu synchronizácie

- Synchronizácia úrovňou hodinového signálu (úrovňová alebo hladinová synchronizácia) (Level Triggered Latch)
- Synchronizácia nábežnou hranou hodinového signálu (Positive edge triggered flip-flop)
- Synchronizácia zostupnou hranou hodinového signálu (Negative edge triggered flip-flop)

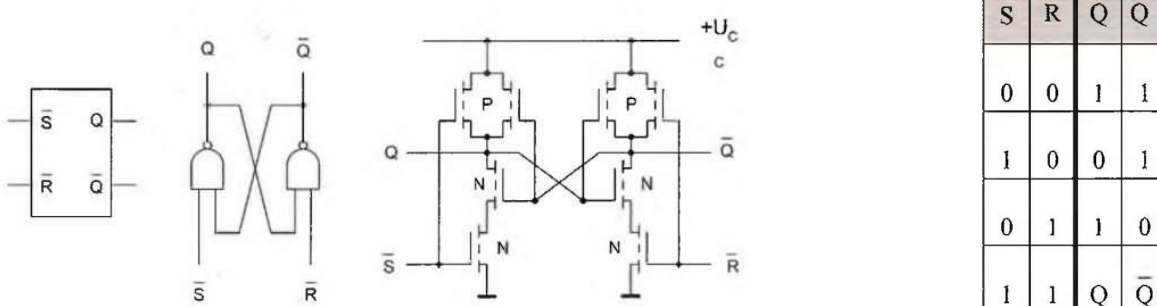
Asynchrónne preklápacie obvody

- Najjednoduchšie sekvenčné súčiastky
- 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.

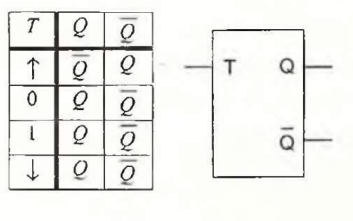
Obvod z NOR:



Obvod z NAND:



Obvod T:



3.2 Stavebné prvky číslicových počítačov. Preklápacie obvody: typy, charakteristika. Základné vlastnosti hranou riadeného PO. Synchronný preklápací obvod D (+pravd. tabuľka).

Preklápacie obvody (PO)

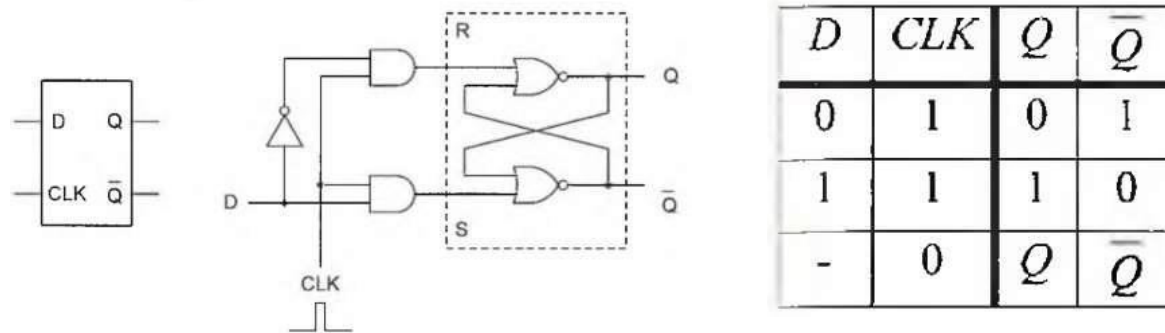
- sú určené na pamätanie jednobitových slov
- jeden alebo viac vstupov I_1, I_2, \dots, I_n (vstupné kanály)
- dva navzájom komplementárne výstupy Q a \bar{Q} (výstupné kanály).
- pamäťové správanie rôznych typov PO je interpretované spravidla Mooreovým automatom.
- astabilné, monostabilné, bistabilné
- synchronné vs. asynchronné

Podľa typu synchronizácie

- Synchronizácia úrovňou hodinového signálu (úrovňová alebo hladinová synchronizácia) (Level Triggered Latch)
- Synchronizácia nábežnou hranou hodinového signálu (Positive edge triggered flip-flop)
- Synchronizácia zostupnou hranou hodinového signálu (Negative edge triggered flip-flop)

Obvod D

- Mení stav na základe zmeny stavu na vstupe (D) a synchronizačného, resp. hodinového impulzu (CLK).



Základné vlastnosti hranou riadeného PO

- Rozhodujúci je stav na vstupoch v tesnom okolí nábežnej hrany hodinového impulzu.
- V iných okamžikoch sú zmeny na vstupoch bezvýznamné.
- Nevhodné časovanie hodinových impulzov a vstupných signálov môže vyvolať metastabilný stav.
- 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.

3.3 Stavebné prvky číslicových počítačov. Preklápacie obvody: typy, charakteristika. Nulovanie a nastavenie PO. Blokovanie preklápacieho obvodu. Synchronny preklápací obvody T a JK (+pravd. tabuľky).

Preklápacie obvody (PO)

- sú určené na pamätanie jednobitových slov
- jeden alebo viac vstupov I_1, I_2, \dots, I_n (vstupné kanály)
- dva navzájom komplementárne výstupy Q a \bar{Q} (výstupné kanály).
- pamäťové správanie rôznych typov PO je interpretované spravidla Mooreovým automatom.
- astabilné, monostabilné, bistabilné
- synchronne vs. asynchronne

Podľa typu synchronizácie

- Synchronizácia úrovňou hodinového signálu (úrovňová alebo hladinová synchronizácia) (Level Triggered Latch)
- Synchronizácia nábežnou hranou hodinového signálu (Positive edge triggered flip-flop)
- Synchronizácia zostupnou hranou hodinového signálu (Negative edge triggered flip-flop)

Nulovanie a nastavenie PO

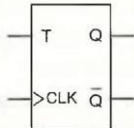
- Niektoré PO D, T a JK majú okrem vstupov CLK, D, T, J a K aj asynchronne vstupy nulovania a nastavenia výstupu.
- Sú účinné vždy bez ohľadu na ostatné vstupy.
- Vstup nulovania, ktorým je vynútený stav $Q = 0$, je označovaný ako R (reset) alebo CLK (clear) aktívny v stave 0.
- Vstup nastavenia, ktorým 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.

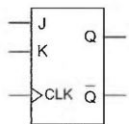
Obvod T:

T	CLK	Q	\bar{Q}
0	\uparrow	Q	\bar{Q}
1	\uparrow	\bar{Q}	Q
-	0	Q	\bar{Q}
-	1	Q	\bar{Q}
-	\downarrow	Q	\bar{Q}



Obvod JK:

J	K	CLK	Q	\bar{Q}
0	0	\uparrow	Q	\bar{Q}
1	0	\uparrow	1	0
0	1	\uparrow	0	1
1	1	\uparrow	\bar{Q}	Q
-	-	0	Q	\bar{Q}
-	-	\downarrow	Q	\bar{Q}
-	-	1	Q	\bar{Q}



3.4 Stavebné prvky číslicových počítačov. Preklápacie obvody: typy, charakteristika. Registre: charakteristika a základné operácie. Pamäte, funkcia pamäte v riadiacej časti ČS.

Preklápacie obvody (PO)

- sú určené na pamätanie jednobitových slov
- jeden alebo viac vstupov I_1, I_2, \dots, I_n (vstupné kanály)
- dva navzájom komplementárne výstupy Q a \overline{Q} (výstupné kanály).
- pamäťové správanie rôznych typov PO je interpretované spravidla Mooreovým automatom.
- astabilné, monostabilné, bistabilné
- synchronné vs. asynchronné

Podľa typu synchronizácie

- Synchronizácia úrovňou hodinového signálu (úrovňová alebo hladinová synchronizácia) (Level Triggered Latch)
- Synchronizácia nábežnou hranou hodinového signálu (Positive edge triggered flip-flop)
- Synchronizácia zostupnou hranou hodinového signálu (Negative edge triggered flip-flop)

Register

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

Základné operácie:	ZS) $RGX := X$	zápis slova X
	NL) $\overline{RGX} := 0$	nulovanie registra RGX
	NS) $\overline{RGX} := 5$	nastavenie registra RGX na dekadickú hodnotu slova $X = (5)_{10} = (101)_2$
	CT) $DO := RGX$	čítanie obsahu RG do výstupného kanálu DO v priamom kóde
	NCT) $DO := \overline{RGX}$	čítanie obsahu RG do výstupného kanálu DO v obrátenom (inverznom) kóde (resp. NCT) $DO := \neg RGX$)

Pamäť

Slúži na zapamätanie polí binárnych reťazcov

Funkcia pamäte v operačnej časti ČS

- statická pamäť typu RAM (Random Access Memory)
- dynamická pamäť typu DRAM (Dynamic RAM)

Funkcia pamäte v riadiacej časti ČS

- statická pamäť typu ROM (Read Only Memory)
- jednorázovo programovateľná pamäť typu PROM (Programmable ROM),
- preprogramovateľná pamäť typu EPROM (Erasable PROM),
- RWM (Read Write Memory)

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

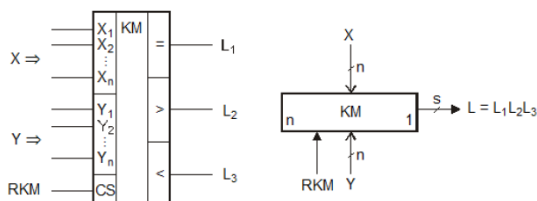
Obvody na generovanie predikátov

- Komparátor
- Generátor parity
- Generátor priority

Komparátor

- je funkčný prvok ČP určený na vyhodnocovanie relácií „=, ≠, <, >, ≤, ≥“ medzi dvoma hodnotami binárnych slov $X = x_n x_{n-1} \dots x_1 x_0$ a $Y = y_n y_{n-1} \dots y_1 y_0$.

$$\begin{aligned} L_1 &| X = Y \\ L_2 &| X > Y \\ L_3 &| X < Y \end{aligned}$$



Generátor parity

- je funkčný prvok komunikačných obvodov ČP určený na zabezpečenie korektného prenosu informácií 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 vysielateľom a prijímačom.
- Na základe uvedeného generátor parity generuje predikát „dané n-bitové slovo obsahuje párný, resp. nepárny počet jedničiek“.

$$L = \begin{cases} 1 & \Leftrightarrow \sum_{i=1}^n x_i \pmod{2} = 0 \\ 0 & \Leftrightarrow \sum_{i=1}^n x_i \pmod{2} = 1 \end{cases}$$

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.
- V danom prípade gen. priority realizuje predikát „i-ty rád daného slova X obsahuje jedničku s najvyššou váhou“, čo sa formálne vyjadrí v tvare: $L_i | GPR(X)$

$$L_i = \begin{cases} x_i & \Leftrightarrow \bigvee_{j=i+1}^n x_j = 0 \\ 0 & \Leftrightarrow \bigvee_{j=i+1}^n x_j = 1 \end{cases}$$

3.6 Stavebné prvky číslicových počítačov. Typy sčítačiek (vymenovať). Polovičná sčítačka (+obr.), úplná sčítačka (+obr.), sčítačka so sériovým prenosom (+obr.).

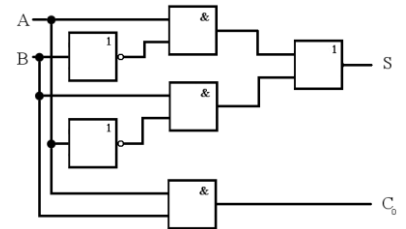
Typy sčítačiek

- 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

Polovičná sčítačka

realizuje sčítanie dvoch jednomiestnych binárnych čísel

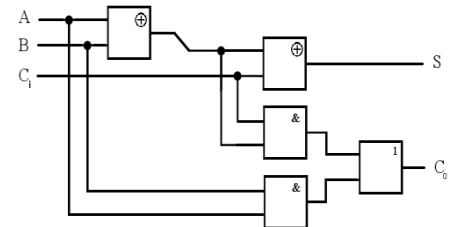
- vstup: dva jednobitové sčítance A, B
 - výstup: jednobitový súčet a jednobitový príznak prenosu do vyššieho rádu (Carry flag)
 - odovzdáva ďalej príznak prenosu, ale sama ho nedokáže spracovať
- $$S = A + B; C = AB$$



Úplná sčítačka

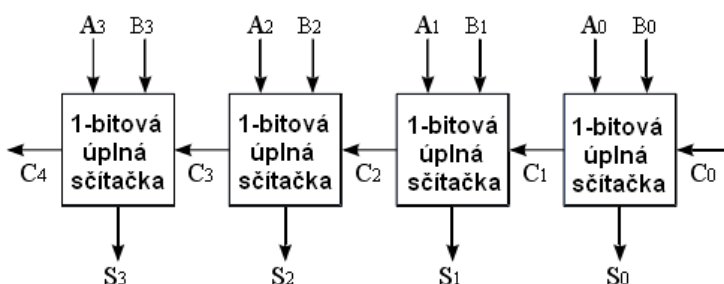
realizuje sčítanie dvoch jednomiestnych binárnych čísel

- pripočítava prenos z predchádzajúceho rádu
- vstup: tri jednobitové sčítance A_i , B_i , C_i (Carry-in)
- výstup: jednobitový súčet a jednobitový príznak prenosu do vyššieho rádu C_{i+1} (Carry-out)



Sčítačka so sériovým prenosom

- vznikne zreťazením N úplných jednobitových sčítačiek prepojením výstupu C_i úplnej sčítačky n-tého bitu so vstupom C_{i+1} sčítačky (n+1) Bitu
- vstup: dve N-bitové čísla plus prenos z predchádzajúceho rádu
- výstup: súčet majúci N+1 bitov (N bitov + prenos)
- príznak prenosu sa odovzdáva z jedného bitu do druhého
- teoreticky rozšíriteľná na ľubovoľný počet bitov
- s rastúcou šírkou (počtom zreťazených sčítačiek) narastá celkové oneskorenie obvodu



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

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

Organizácia zobrazovania objektov v pamäti

- Usporiadanie bajtov objektov ~ ukladací endián
- Zarovnávanie viacbajtových objektov

Základné pojmy

- Ľubovoľnú diskretnú informáciu v číslicovom počítači zobrazujeme prostredníctvom sústavy symbolov, ktorým môžeme priradiť rôznečíselné hodnoty.

- Reprezentácia číselných hodnôt vopred definovanou sústavou čísl sa nazýva číselná sústava

Číselné sústavy

- Pozitívne - Nepozitívne - Symetrické - Nesymetrické

□ Priamy kód

$$(N^P)_B = N_{sign}^P \underbrace{N_{n-1}^P \dots N_0^P}_{|(N^P)_B|} = N_{sign}^P (N_{mod}^P)_B$$

kde

$$N_{sign}^P = \begin{cases} 0 & \Leftarrow [N] \geq [0] \\ 1 & \Leftarrow [N] < [0] \end{cases}$$

□ Inverzný kód

$$(N^I)_B = N_{sign}^I N_{n-1}^I N_{n-2}^I \dots N_0^I$$

kde

$$N_{sign}^I = \begin{cases} 0 & \Leftarrow [N] \geq [0] \\ 1 & \Leftarrow [N] < [0] \end{cases}$$

$$[(N^I)_B] = \begin{cases} |N| & \Leftarrow [N] \geq [0] \\ 2^n - 1 - |N| & \Leftarrow [N] < [0] \end{cases}$$

□ Doplnkový kód

$$(N^D)_B = N_{sign}^D N_{n-1}^D N_{n-2}^D \dots N_0^D$$

kde

$$N_{sign}^D = \begin{cases} 0 & \Leftarrow [N] \geq [0] \\ 1 & \Leftarrow [N] < [0] \end{cases}$$

$$[(N^D)_B] = \begin{cases} |N| & \Leftarrow [N] \geq [0] \\ 2^n - |N| & \Leftarrow [N] < [0] \end{cases}$$

□ Kód s posunutou nulou / Modifikovaný kód

$$(N^M)_B = N_{sign}^M N_{n-1}^D N_{n-2}^D \dots N_0^D$$

kde

$$N_{sign}^M = \begin{cases} 1 & \Leftarrow [N] \geq [0] \\ 0 & \Leftarrow [N] < [0] \end{cases}$$

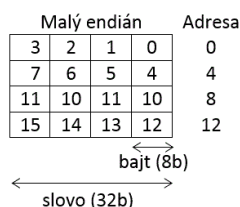
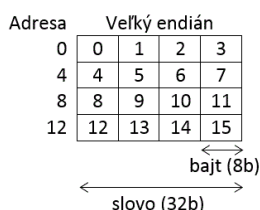
$$[(N^M)_B] = \begin{cases} 2^n + |N| & \Leftarrow [N] \geq [0] \\ 2^n - |N| & \Leftarrow [N] < [0] \end{cases}$$

Veľký bajtový endián

- najvýznamnejší bajt (Most Significant Byte) na najnižšiu adresu

Malý bajtový endián

- najmenejvýznamnejší bajt (Least Significant Byte) na najnižšiu adresu



Sprístupňovanie slov

Zarovnané sprístupňovanie bajtov

- Jednoduchosť návrhu HW
- Jednoduché sprístupňovanie
- Využitie v RISC

Nezarovnaný výber bajtov

- Zložitý návrh HW
- Zložitý sprístupňovanie
- Využitie v CISC

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

Reprezentácia údajov

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

Organizácia zobrazovania objektov v pamäti

- Usporiadanie bajtov objektov ~ ukladačí endián
- Zarovnávanie viacbajtových objektov

Operácie s celými číslami

- Sčítanie, odčítanie, násobenie, delenie

Operácia násobenia v PRČ

- Nech $A[n-1:0]$, $B[n-1:0]$, $Q[n-1:0]$, $C[1]$ a PC sú registre. Nech PC slúži ako počítadlo cyklov. Zavedme **C#Q#A** na označenie zreťazenia registrov. Potom

Algoritmus celočíselného násobenia

```
BEGIN
  C#Q := 0 || PC := n
  WHILE (PC > 0)
    IF A(0) = 1 THEN
      C#Q := Q + B
    ENDIF
    C#Q#A := 0 # SR1(C#Q#A) ||
    PC := PC - 1
  END
END
```

Algoritmus celočíselného delenia

```
BEGIN
  C#Q := 0 || PC := n
  WHILE (PC > 0)
    C#Q#A := SL1(C#Q#A)#0
    IF (Q ≥ B) THEN
      C#Q := Q - B || A(0) := 1
    ENDIF
    PC := PC - 1
  END
END
```

Booth-ov algoritmus

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

Pohyblivá rádová čiarka

$$N = \pm M \times 2^{\pm E}$$

kde M je mantisa čísla; ak $0.5 \leq M < 1$ potom mantisa je v normalizovanom tvare

E je exponent čísla v rozsahu $-(2^p - 1) \leq E \leq 2^p - 1$; kde p je počet bitov v exponente - 1

Normalizované binárne číslo podľa štandardu IEEE-75

$$N = \pm 1.M \times 2^X = 1.\overbrace{bb|bbb\dots b}^X \times 2^X$$

kde X je charakteristika čísla ($X = E + 2p - 1$)

b je binárna číslica (0 alebo 1)

Znamienko sa nachádza v pozícií najvýznamnejšieho bitu (najľavejší bit) v zobrazovacom formáte

Najvýznamnejší bit mantisy v zobrazovacom formáte nevystupuje

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 D_i je množina všetkých hrán vychádzajúcich z vrcholu (i).

Násobenie v PHRČ

$$\sum_{(i,j) \in D_i} p_{i,j} = 1$$

Operácia násobenia násobenca s násobiteľom sa skladá z niekoľkých častí:

- násobenie mantís podľa algoritmu celočíselného násobenia bezznamienkových, resp. znamienkových čísel (Boothov algoritmus)
- násobenie sa uskutočňuje na základe algoritmu celočíselného bezznamienkového násobenia
- zaokrúhlenie výsledku, pretože výsledkom násobenia dvoch n-rádových čísel je 2n-rádový, resp. (2n-1)-rádový výsledok, ktorého n najnižších rádov sa použije v procese zaokrúhľovania výsledku
- výpočet exponenta výsledku, ktorý zahŕňa odčítanie bázy posunutia od súčtu exponentov s posunutou nulou.

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

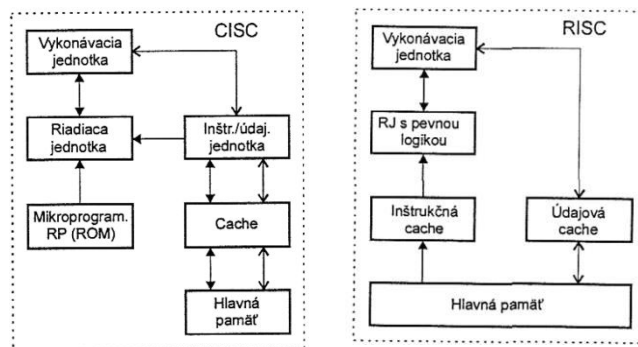
Inštrukčno orientovaná architektúra (ISA)

CISC (Complex Instruction Set Computer)

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

RISC (Reduced Instruction Set Computer)

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



Inštrukcie a ich funkcia

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

Súbor inštrukcií

- 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 inštrukcií

<FI>::=<KO><PKO>{<PAO>[,AO]} -- informácie potrebné k vykonaniu inštrukcie

KO	Kód operácie
PKO	Príznak kódu operácie
PAO	Príznak adresy operandov
AO	Adresa operandov

Formát inštrukcii závisí od:

- **vnútorných registrov** (počet – dĺžka FI, operačná rýchlosť, typ – univerzálne, špeciálne)
- **adresovacích módov** (zložitosť a počet, použitie – dynamická/statická alokácia)

3-adresové inštrukcie: **<FI>::=<KO><CO><ZO1><ZO2>**

2-adresové inštrukcie: **<FI>::=<KO><CO/ZO><ZO>**

1-adresové inštrukcie: **<FI>::=<KO><ZO>**

0-adresové inštrukcie: **<FI>::=<KO>**

Adresovacie módy

Registrový	(RG1<-RG2)
Bezprostredný	(RG1<-konst)
Priamy	(RG1<-M[konst])
Nepriamy registrový .	(RG1<-M[RG2])
Indexovací	(RG1<-M[(RG2)*krok])
Relatívny	(PC<-PC+krok+posuv <= podm="ano")
	(PC<-PC+krok <= podm="nie")

5.2. Východiská architektonických koncepcií číslicových počítačov. Základné charakteristiky ČP: výkonnosť, systémové vlastnosti. Hodnotenie výkonnosti (TCPU, ACPI). Jednotky výkonnosti.

Základné charakteristiky ČP:

Výkonnosť ČP – kvantitatívne hodnotenie

- čas odozvy (Response Time),
- priepustnosť (Throughput),
- koeficient vyťažiteľnosti (Efficiency).

Systémové vlastnosti – kvalitatívne hodnotenie

- súbor inštrukcií,
- kompilačná technológia,
- implementácia a riadenie CPU,
- technológia a hierarchia pamäťového podsystemu.

Hodnotenie výkonnosti

Čas vykonania programu: $T = T(\text{CPU}) + T(\text{W})$

pozn. dolný index v zátvorke

$T(\text{W})$ - čas čakania na vykonanie V/V operácie

$T(\text{CPU})$ - čas v priebehu ktorého CPU vykoná program $\rightarrow T(\text{CPU}) = T(\text{CPU}_p) + T(\text{CPU}_s)$

$T(\text{CPU}_p)$ - používateľský čas

$T(\text{CPU}_s)$ - systémový čas

Priemerný počet SC inštrukcie v danom programe: $\text{ACPI} = N / NI$

N - celkový počet SC potrebných na vykonanie programu

NI - počet inštrukcií daného programu

Jednotky výkonnosti

MIPS - 10⁶ inštrukcií za sekundu (Milion Instruction Per Second, odvodené jednotky: GIPS)

FLOPS - počet operácií za sekundu vykonaných v pohyblivej rádovej čiarke (odvodené jednotky: MFLOPS, GFLOPS, TFLOPS),

KLIPS - 10³ logických vývodov za sekundu (Logic Inference Per Second, určené na definovanie výkonnosti počítačového prostredia na riešenie úloh umelej inteligencie (UI), jeden vývod zahŕňa približne 1000 inštrukcií)

Dhrystone - 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),

Whetstone - 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 transcendentálne funkcie.

LINPACK - určené na testovanie aplikačných programov

SPEC - zastaralý benchmark

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

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)

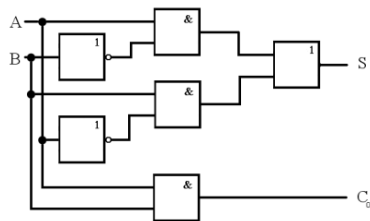
$F(x,y): D \times D \rightarrow D$ f =operácia x,y =vstupné premenné D =operandy operačných jednotiek

Polovičná sčítacia

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

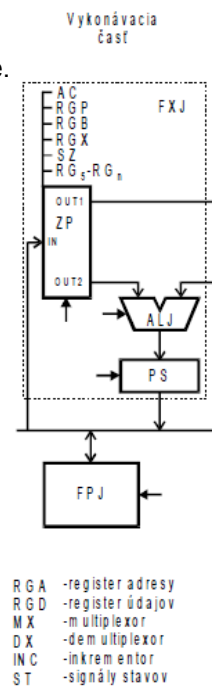
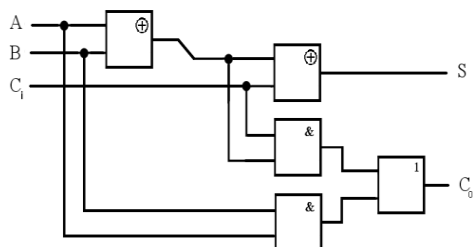
$$S = A + B; C = AB$$



Úplná sčítacka

realizuje sčítanie dvoch jednomiestnych binárnych čísel

- pripočítava prenos z predchádzajúceho rádu
- vstup: tri jednobitové sčítance A_i , B_i , C_i (Carry-in)
- výstup: jednobitový súčet a jednobitový príznak prenosu do vyššieho rádu C_{i+1} (Carry-out)



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 číslíkového 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 postupností ich vykonávania v súčinnosti s programovým počítadlom (PC- Program Counter), jednotkou vetvenia (BU- Branch Unit), registrom inštrukcií (RI) a dekóderom inštrukcií (DKI)
- synchronizačnej (**SRJ**) a funkčnej (**FRJ**) riadiacej jednotky, určenej na generovanie interných a externých signálov IT, ET, IR, ER

Komponenty RJ

Jednotka predvýberu inštrukcií(JPI)

- výber,
- dekódovanie

Inštrukčná jednotka(IJ)

- rozklad na mikrooperácie,
- vykonanie mikrooperácie

Jednotka adres skokov(JAS)

- riadenie a plánovanie vykonávania mikrooperácie

KLASIFIKÁCIA RIADIACEJ JEDNOTKY (RJ)

RJ s pevnou logikou

- spontánne riadenie

RJ s programovateľnou logikou

- algoritmické riadenie

□ Inštrukčný cyklus

- je postupnosť elementárnych krokov (strojových cyklov - SC), v priebehu ktorých procesor vykoná operáciu definovanú inštrukciou programu.
- Čas T_{SC} vykonania SC je

$$T_{SC} = n \times T_C$$

kde

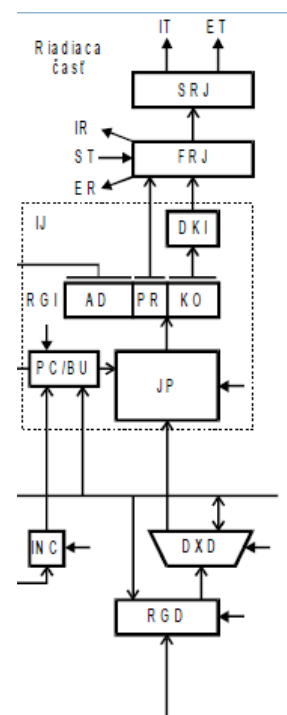
- T_C –perióda taktovacej (hodinovej) frekvencie procesora,
- n – počet periód T_C .

□ Inštrukčný cyklus

- Inštrukčný cyklus CPI pozostáva z m fáz F_m z ktorých každá trvá n_i SC, kde $i = 1, 2, \dots, m$.

$$CPI = F_1 + F_2 + \dots + F_m$$

- 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 RČ z obr. 6.1. Komponenty stykovej časti a ich funkcia.

Procesor je multifunkčná jednotka číslcového 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 podsystemu.

- Zabezpečuje styk s ostatnými jednotkami ČP pri prenose informácií
- Zabezpečuje prenos informácií medzi procesorom a hlavnou pamäťou resp. V/V podsystemom

Komponenty stykovej časti

1. Jednotka správy pamäte

- adresova pamäť (ADJ), určená na výpočet virtuálnej adresy(VA)
- jednotka riadenia pamäte(MMU- Memory Management Unit), určená na prevod virtuálnej adresy (VA) na fyzickú adresu (FA) pamäťového priestoru v hlavnej pamäti.

Prevod logických adries (LA) na fyzické adresy (FA)

- Logický adresový priestor
- stránky
- adresa slova = базовá adresa stránky + posuv

Fyzický adresový priestor

- stránkové rámy
- adresa slova = базовá stránkového rámu + posuv
- kapacita logického priestoru >> kapacita fyzického adresného priestoru

2. Vyrovnávací pamäť typu cache, určená na rýchlu komunikáciu procesora s pamäťovým podsystemom

- obsahuje adresy najviac používaných stránok vo vopred definovanom časovom intervale práve prebiehajúcej časti programu

3. Zbernicová jednotka

- Zabezpečuje rozhranie medzi procesorom a ostatnými komponentmi ČP pri vykonávaní V/V a pamäťových inštrukcií
- organizuje prenos informácií medzi procesorom V/V, resp. pamäťovým podsystemom ČP.
- Synchronizuje prenos informácií

7.1. Teória prúdového spracovania výpočtového procesu. Definícia zret'azenia. Podmienky zret'azenia, sekvenčné a prúdové spracovanie. Stupne systému zret'azenia.. Časový diagram spracovania prúdu inštrukcií v skalárnych (obr.), superskalárnych (obr.) a superprúdových procesoroch (obr.).

Zret'azenie

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) zret'azenia**, 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) zret'azenia.

Spracovanie n-úloh, z ktorých každá je rozložiteľná na k podúloh (krokov), môže byť realizované

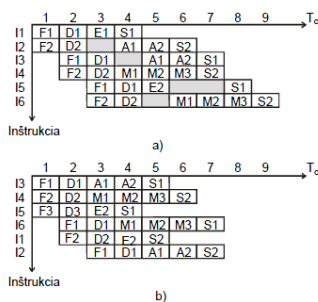
- štandardnou funkčnou jednotkou (FJ), kompletizujúcou výsledky riešenia jednotlivých úloh vždy až po uplynutí intervalu $T_i = kT_k$ potrebného na ich vykonávanie (T_k – doba vykonania k-tej podúlohy),
- zret'azenou (prúdovou) funkčnou jednotkou (PFJ), kompletizujúcou výsledky riešenia jednotlivých úloh už po uplynutí intervalu T_k , z čoho vyplýva, že výpočet nasledujúcej úlohy začína ešte pred ukončením úlohy predchádzajúcej.

Stupne systému zret'azenia

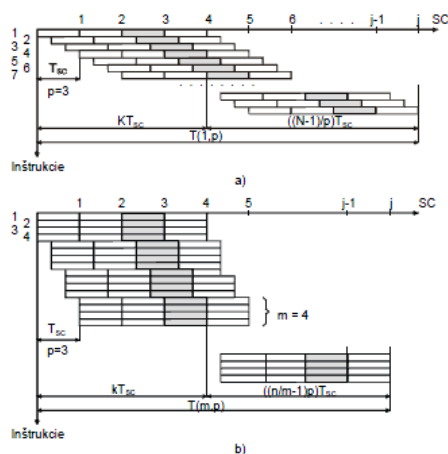
1. Stupeň F(etch)(načítava inštrukcie z pamäte)
2. Stupeň D(encode)(dekóduje inštrukcie)
3. Stupeň E(xecute)(vykonáva operáciu definovanú
4. Stupeň S(tore), resp. WB (WriteBack)(zapisuje operácie do registra/pamäte)

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
1. inštrukcia	F ₁	D ₁	E ₁	WB ₁			
2. inštrukcia		F ₂	D ₂	E ₂	WB ₂		
3. inštrukcia			F ₃	D ₃	E ₃	WB ₃	
4. inštrukcia				F ₄	D ₄	E ₄	WB ₄
5. inštrukcia					F ₅	D ₅	E ₅
6. inštrukcia						F ₆	D ₆
7. inštrukcia							F ₇

v IF KO)
výsledok



Obr. 8.2 Superskalárne spracovanie (a - bez dopredného prehľadávania, b - s dopredným prehľadáváním a usporiadaným vstupným prúdom)



Obr. 8.4 Časové diagramy spracovania prúdu inštrukcií v (a - superprúdový procesore, b - superprúdovo superskalárnom procesore)

7.2. Teória prúdového spracovania výpočtového procesu. Doba vykonania prúdu n inštrukcií.
Efektívnosť prúdového spracovania. Základné úlohy syntézy prostriedkov na zret'azené spracovanie
úloh. Základné typy zret'azených systémov. Základné pojmy:
inicializácia zret'azenia, rezervačná tabuľka, latentnosť, stavový graf inicializácií.

Doba vykonania prúdu n inštrukcií

$$T(m,p) = \left(k + \left(\frac{n}{m} - 1 \right) \frac{1}{p} \right) T_{sc} = \left(k + \frac{n-m}{mp} \right) T_{sc}$$

- T_{sc} – 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.

V prípade skalárneho zret'azenia: $m = p = 1$

V prípade superskalárneho zret'azenia: $m > 1$ a $p = 1$

Efektívnosť prúdového spracovania je daná 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

$$\frac{T_s}{T_p} = \lim_{n \rightarrow \infty} \frac{n}{1 + (n - 1)/k} = k$$

Základné úlohy syntézy prostriedkov na zret'azené spracovanie úloh

- Analýza procesu zret'azenia - Stratégia riadenia procesu zret'azenia - Návrh technických prostriedkov

Základné typy zret'azených systémov

Lineárny systém zret'azenia - vstupný prúd údajov postupne prechádza všetkými jeho k stupňami počas n synchronizačných intervalov (krokov) T_i ($i = 1, 2, \dots, n$) zret'azenia.

Nelineárny systém zret'azenia - sa charakterizuje spätnoväzobným prepojením jeho jednotlivých stupňov; - proces zret'azenia nie je definovaný jednoznačne, t.j. môže byť zobrazený viacerými RTB;
- takéto zret'azené systémy sa nazývajú dynamické

Inicializácia zret'azenia

- Začiatok procesu zret'azeného spracovania vstupného prúdu údajov zobrazeného v RTB.

Latentnosť (L)

- Počet synchronizačných intervalov (krokov) medzi začiatkami dvoch za sebou nasledujúcich inicializácií zret'azenia.
- Cyklicky opakovaná sekvencia prípustných latentností sa nazýva cyklus latentností ($L(j)$).
- Priemerná latentnosť ($L[s](i)$) – aritmetický priemer
- Minimálna latentnosť ($L[\min]$) – funkcia min

Stavový graf inicializácií

- Vyjadruje prípustné stavy prechodov medzi postupnosťou následných inicializácií.
- Umožňuje vypočítať $L[\min]$
- vrcholy grafu zobrazujú v danom časovom okamihu všetky možné nasledujúce inicializácie

Rezervačná tabuľka

- postupnosť aktivácie jednotlivých stupňov v priebehu zret'azeného spracovania danej funkcie v PFJ

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

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 zret'azenia (prúdového spracovania), ktoré môžu byť : údajové, štruktúrne a riadenia

Hazardy zret'azenia(prúdového spracovania)

Údajové

- 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:

1. **údajová závislosť**, spôsobuje **hazard** typu **RAW(ReadAfterWrite)**,

I1: load RG1,M[A] => RG1:=M[A]

I2: store M[B], RG1 => M[B]:=RG1

2. **údajová antizávislosť**, spôsobuje **hazard** typu **WAR(WriteAfterRead)**,

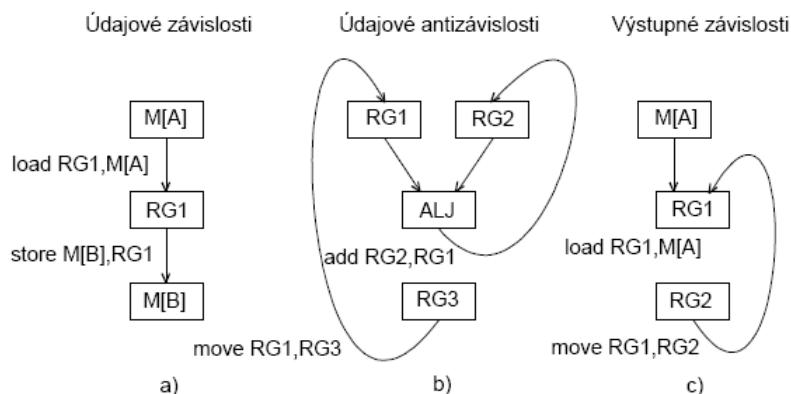
I1: add RG2,RG1 => RG2:=RG1+RG2

I2: move RG1,RG3 => RG1:=RG3

3. **výstupná závislosť**, spôsobuje **hazard** typu **WAW(WriteAfterWrite)**.

I1: load RG1,M[A] => RG1:=M[A]

I2: move RG1, RG2 => RG1:=RG2



Obr. 7.2 Príklady vzniku údajových hazardov typu
(a - RAW, b - WAR, c - WAW)

Podstata vzniku RAW

Pri jednoduchom prúdovom spracovaní inštrukcií môže vzniknúť iba hazard RAW, ktorý spôsobuje využívanie spoločného registra v dvoch rôznych fázach nasledujúcich inštrukcií. Hazardy WAR a WAW vznikajú iba

v osobitných prípadoch prúdového spracovania inštrukcií.

Hazard RAW pri prúdovom spracovaní inštrukcií sa môže odstrániť:

Softvérovými prostriedkami

- vloženie prázdnej inštrukcie po inštrukcií, ktorá môže spôsobovať hazard,
- preusporiadanie vykonávania inštrukcií

Hardvérovými prostriedkami

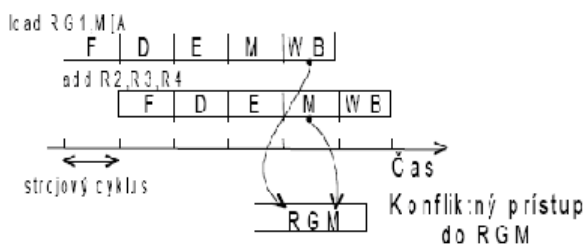
- blokovanie prúdového spracovania, v dôsledku ktorého sa generujú prázdne cykly v prúdovom spracovaní inštrukcií;
- dopredné generovanie výsledku predchádzajúcej inštrukcie pre jeho použitie ako vstupného operandu nasledujúcej inštrukcie.

Štrukturálne hazardy a ich odstránenie

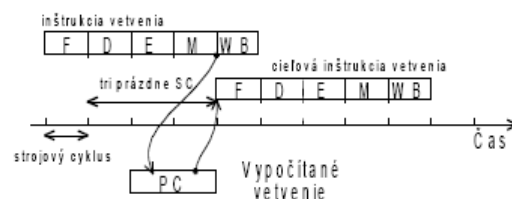
Štrukturálne hazardy vznikajú, ak dve za sebou nasledujúce inštrukcie využívajú tie isté zdroje technických prostriedkov zreťazného systému.

Hazard sa môže riešiť:

- **vložením prázdneho strojového cyklu** (oneskorenie vykonania druhej inštrukcie)
- **použitím viacnásobných zdrojov** (rôznych registrov, ktoré časovo oddeľujú prístup do registrovej pamäte -RGM).



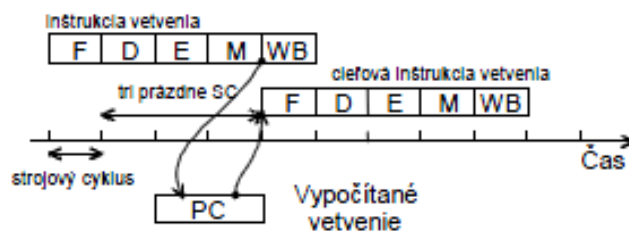
Obr. 7.4 Konflikt zdrojov v prípade štrukturálneho hazardu



Obr. 7.5 Odstránenie hazardu riadenia v prípade inštrukcie vetvenia

Hazardy Riadenia a jeho odstránenie

- Hazardy riadenia vznikajú pri prúdovom vykonaní inštrukcií **skokov (jump)** a **vetvenia (branch)**.



Obr. 7.5 Odstránenie hazardu riadenia v prípade inštrukcie vetvenia

9.4. Pamäťový podsystem. Základné mechanizmy riadenia logického adresného priestoru. Mapovanie virtuálnej pamäte. Mapovanie prostredníctvom tabuľky stránok (+obr.). Mapovanie prostredníctvom pamäti preložených adres (+obr.). Segmentová organizácia pamäte (+obr.). Metódy vyrad'ovania stránok. Stratégia FIFO vyrad'ovania stránkových rámov.

Základné mechanizmy riadenia logického adresného priestoru

mapovanie pamäte

- prevod virtuálnych adres na fyzické adresy
- premiestňovanie stránok medzi FP a VP

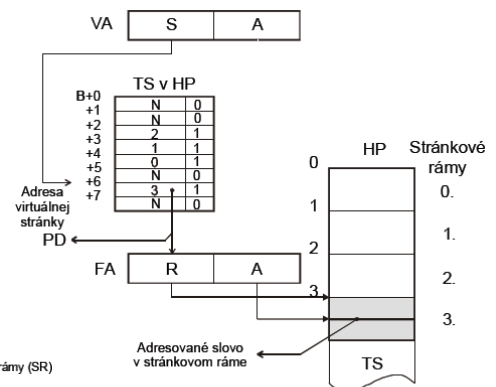
Mapovanie virtuálnej pamäte

- mapovanie prostredníctvom tabuľky stránok TS (PT- Page Tables)
- mapovanie prostredníctvom pamäte preložených stránok PPS (TLB- Translation Lookaside Buffer)
- mapovanie prostredníctvom registrovej pamäte RP

Mapovanie prostredníctvom tabuľky stránok

Ak **B** je pevne definovaná bázoá adresa TS uloženej v HP, tak výsledná adresa (R) stránkového rámu, zodpovedajúceho stránke s virtuálnou adresou (S) je $R = TS[(B) + S]$

Nedostatok- pomalé mapovanie v dôsledku dvojnásobného prístupu do HP pri každom výpočte FA.



Obr. 9.12 Mapovanie VP prostredníctvom TS

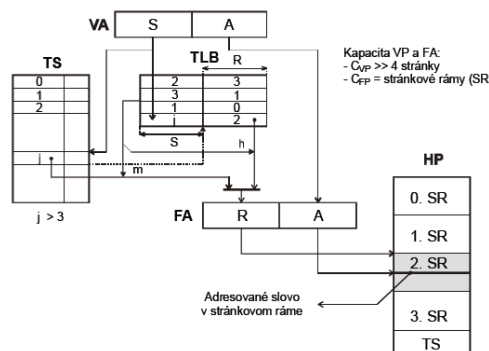
Mapovanie prostredníctvom pamäti preložených adres

Sú v nej uložené adresy najčastejšie používaných stránok v práve prebiehajúcom intervale spracovania programu.

Položka v TLB (asociatívna pamäť) má formát $TLB = \langle S \rangle \langle R \rangle$

Pri asociatívnom prehľadávaní TLB sú definované nasledujúce situácie

- prítomnosť stránky v TLB (h -hit), výpadok stránky v TLB (m -miss).



Obr. 9.13 Mapovanie VP prostredníctvom TLB

Segmentová organizácia pamäte

Segmentová organizácia pamäte

vyplýva z požiadavky vytvoriť v pamäťovom podsysteme priestor premenlivej dĺžky.

-Môže byť uložený od ľubovoľnej adresy virtuálneho priestoru.

Spravidla sa používa v kombinácii so stránkovou organizáciou pamäte.

Segmentovo-stránková (viacúrovňová) organizácia virtuálnej pamäte.

Metódy vyrad'ovania stránok

- Ak vo FP (v HP) nie je umiestnená stránka z adresovaného priestoru VP

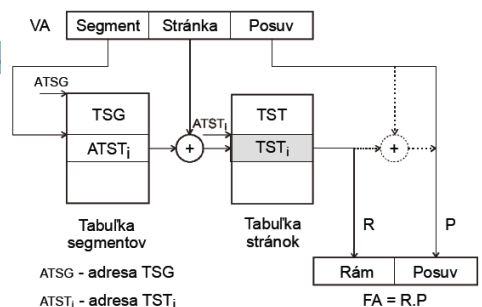
$VP[TLB[\gamma].S] := HP[TLB[\gamma].R]$ vyradenie stránkového rámu z HP $HP[TLB[\gamma].R] := VP[VA]$

.....zápis kópie stránky do HP

$TLB[\gamma].S := VA$ úprava TLB

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

$$h \mid S \in TLB, m \mid S \notin TLB$$



Obr. 9.14 Mapovanie adresového priestoru v segmentovo-stránkovo organizovanom virtuálnom systéme

Stratégia FIFO

- ▣ Vyradí sa stránkový rám, ktorý bol najdlhšie umiestnený v HP (resp. v cache)

$$PC_i := \begin{cases} PC_i & \Leftarrow P_i = 0 \\ PC_i + 1 & \Leftarrow P_i = 1 \end{cases}$$

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