

Informatika, informatizácia a informačná spoločnosť

- Informatika je veda o získavaní, zbere, prenose, triedení, ukladaní, uchovávaní (pamätaní), aktualizovaní, spracovaní, vyhodnocovaní a využívaní informácií na úrovni signálov, údajov, symbolov, správ, poznatkov a znalostí

Výpočtová technika

- skúma zákonitosti a princípy tvorby informačných procesov a spôsoby ich realizácie
- zaoberá sa vlastnosťami a zákonitosťami týchto procesov a ich algoritmickou realizáciou
- zahrňuje tvorbu, vývoj a využitie programových a technických prostriedkov výpočtovej techniky, ako nástroja na automatizované a automatické spracovanie informácií

Číslicový počítač (ČP)

- je zložitý univerzálny číslicový systém určený na samočinné vykonávanie postupnosti operácií nad údajmi zobrazenými číslicovým kódom, na základe vopred pripraveného a v pamäti uloženého programu

Systém a jeho vlastnosti

- je súhrn prvkov komponovaných do jedného celku za účelom dosiahnutia definovaného cieľa
- Kompozícia systému - vlastnosť sa nazýva zložitosť systému
- Opisanie systému sa uskutočňuje na základe určenia jeho funkcie a štruktúry
- Funkcia systému predstavuje pravidlo na dosiahnutie požadovaného cieľa
- Štruktúra systému vyjadruje kompozíciu jeho prvkov a ich vzájomných väzieb
- Organizácia systému je spôsob vytvorenia systému na báze kompozície prvkov z ktorých pozostáva za účelom dosiahnutia zadaných funkčných vlastností
- Počítačový systém (PS) - systém vytvorený na báze jedného alebo viacerých ČP, je charakterizovaný svojou:
 - funkčnou organizáciou
 - štruktúrnou organizáciou
- Funkčná organizácia PS je definovaná formou zobrazovania informácií, štruktúrou inštrukčných súborov, charakterom vnútorného a vonkajšieho riadenia operácií, spôsobom riadenia procesu spracovania informácií, prístupovým režimom používateľov k PS
- Štruktúrna organizácia PS je definovaná logickým a systémovým usporiadaním jeho komponentov, spôsobom ich vzájomnej komunikácie, konštrukčnou a technologickou realizáciou jeho komponentov
- Architektúra PS zahrňuje:
 - požiadavky programových prostriedkov - inštrukčno orientovaná architektúra
 - vnútornú organizáciu jeho technických prostriedkov - implementačno orientovaná architektúra
- Z hľadiska požiadaviek programovania inštrukčno orientovaná architektúra PS predstavuje abstrakciu prostredníctvom súboru inštrukcií - kódy operácií, adresovacie módy, špecifikáciu registrov, virtuálnu pamäť a pod
- Z hľadiska implementácie tech. prostriedkov implementačno orientovaná architektúra PS predstavuje abstrakciu na úrovni štruktúrnej organizácie - centrálnych procesorových jednotiek, komponentov pamäťového podsystemu, zberných systémov, mikrokódu, prúdových funkčných jednotiek

Hradlá

- Realizujú logické funkcie
 - Jedno-vstupové:
 - NOT gate, buffer

NOT



$$Y = \bar{A}$$

A	Y
0	1
1	0

BUF



$$Y = A$$

A	Y
0	0
1	1

- Dvoj-vstupové:

- AND, OR, XOR, NAND, NOR, XNOR

AND



$$Y = AB$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR



$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

XOR



$$Y = A \oplus B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

NAND



$$Y = \overline{AB}$$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR



$$Y = \overline{A + B}$$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

XNOR



$$Y = \overline{A \oplus B}$$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

- Viac-vstupové

Logické úrovne

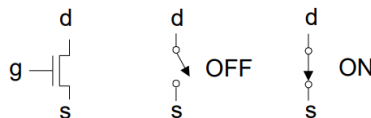
- Číslicové signály majú konečný počet diskretných hodnôt, väčšinou dve
- Rozdielne pásmo pre vstup a výstup

Tranzistory

- Základným stavebným prvkom hradíel
- Má 3-elektrody:
 - bipolárne: C, B, E; unipolárne: D, G, S, d a s sú spojené (ON) ak g je 1

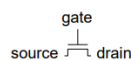
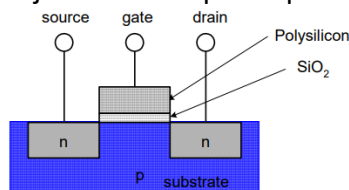
g = 0

g = 1



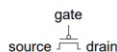
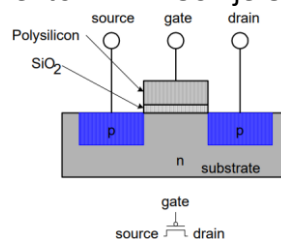
MOS Tranzistor (Metal Oxide Silicon)

- Najrozšírenejší druh tranzistorov riadených elektrickým poľom
- Vodivosť kanálu medzi s a d je riadená napätím privedeným na g



nMOS

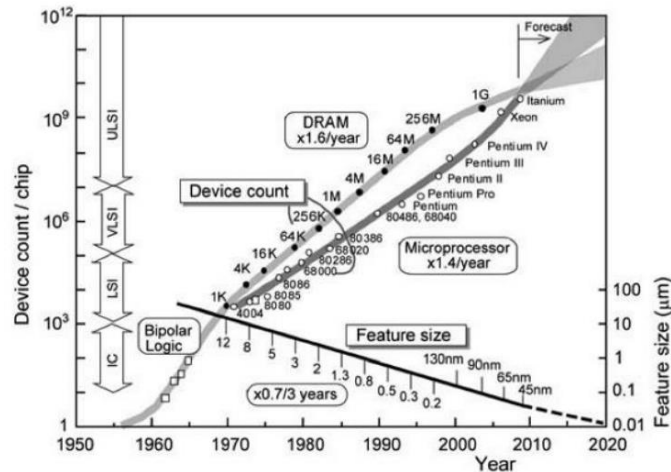
- pMOS tranzistor je komplementom k nMOS - je ON ak Gate = 0, OFF ak Gate = 1



- nMOS: dobre prenáša „nuly“, pMOS: dobre prenáša „jednotky“

Gordon Moore, 1929-

- Spoluzakladateľ firmy Intel
- Moore-ov zákon: Počet tranzistorov sa zdvojnásobí každý „rok“ (empirické pravidlo, 1965)



Logický systém

- je charakterizovaný
 - abstraktným opisom funkcie systému
 - štruktúra systému → logický obvod
- Logický obvod (LO) je definovaný pomocou:
 - svojich vstupov (X) a výstupov (Y), funkčnej špecifikácie a časových charakteristík

Logické obvody

- Štruktúru logického obvodu tvoria logické členy a vzájomné väzby medzi nimi
- Uzly - spojenie dvoch alebo viacerých kanálov
- Typy logických obvodov
 - Kombinačný logický obvod (KLO)
 - Výstup závisí len od okamžitej kombinácii vstupných hodnôt
 - Sekvenčný logický obvod (SLO)
 - výstup závisí od kombinácie vstupov a od vnútorných stavov obvodu z predchádzajúceho taktu

Kombinačný log. obvod

- Každý element obvodu je kombinačný obvod
- Každý vstup je spojený s výstupom len z jedného elementu obvodu
- Obvod neobsahuje slučky

Booleovské rovnice

- Definujú vzťah medzi vstupnými a výstupnými b. premennými

Definície

- Komplement: negovaná b. premenná $\bar{A}, \bar{B}, \bar{C}$
- Literál: b. premenná v priamej alebo v negovanej forme $A, \bar{A}, B, \bar{B}, C, \bar{C}$
- Implikant: súčin všetkých vstupných b. premenných v priamej alebo negovanej forme
- Implicant: súčet všetkých vstupných b. premenných v priamej alebo negovanej forme

Disjunktívna normálna forma

- je súčet implikantov, Implikant je súčin literálov
- Každý implikant definuje pravdivostnú hodnotu TRUE

A	B	Y	implikant	označenie implikantu
0	0	0	$\bar{A}\bar{B}$	m_0
0	1	1	$\bar{A}B$	m_1
1	0	0	$A\bar{B}$	m_2
1	1	1	AB	m_3

$$Y = F(A, B) = \bar{A}B + AB = \Sigma(1, 3)$$

Konjunktívna normálna forma

- je súčin implicitentov, Implicitent je súčet literálov
- Každý implicitent definuje pravdivostnú hodnotu FALSE

A	B	Y	implicitent	označenie implicitentu
0	0	0	$A + B$	M_0
0	1	1	$A + \bar{B}$	M_1
1	0	0	$\bar{A} + B$	M_2
1	1	1	$\bar{A} + \bar{B}$	M_3

$$Y = F(A, B) = (A + B) \cdot (\bar{A} + B) = \Pi(0, 2)$$

Algebraické vyjadrenia a LO

- Booleova algebra: Na báze: NOT, AND, OR
- Shefferova algebra: Na báze: NAND
- Peirceova algebra: Na báze: NOR
- Prevodov NAND na OR s neg. vstupmi:



- Prevod OR s neg. vstupmi na NAND:



Pravidlá kreslenia schém LO

- Vstupy sú naľavo, Výstupy sú napravo, Hradlá sú orientované zľava doprava, Každý kanál sa znázorňuje samostatnou čiarou
- Kríženie vodičov bez bodky: nie je vodivé spojenie medzi kanálmi
- Signál s hodnotou „X“:
 - X reprezentuje hodnotu na ktorej nezáleží, X reprezentuje konflikt
- Vysokoimpedančná hodnota „Z“:
 - Reprezentuje vysokoimpedančný stav zdroja signálu
 - výstupný signál môže mať hodnotu 0 ako aj 1
- Trojstavové zbernice: Obvody sú v stave vysokej impedancie, odpojené od zbernice

Karnaughové mapy (K-mapy)

- Princíp minimalizácie pomocou Karnaughových máp spočíva v aplikovaní pravidla spojovania na všetky dvojice susedných elementárnych súčinov
- Komplement: b. premenná v negovanej forme
- Literál: b. premenná v priamej forme alebo jej komplement
- Implikant: súčin literálov
- Prostý implikant: implikant definovaný s pravidelnou konfiguráciou s maximálnym možným stupňom k

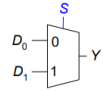
Kombinačné logické obvody

- Multiplexor, Dekodér, Demultiplexor, Komparátor, Prevodník kódu, Polovičná a úplná sčítačka

Multiplexor (MUX)

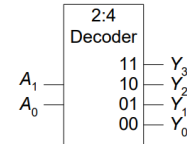
- Má M adresných (riadiacich), N dátových vstupov a jeden výstup, $M = \log_2 N$
- Implementácia: Pomocou hradiel, Trojstavové buffre

2:1 Mux



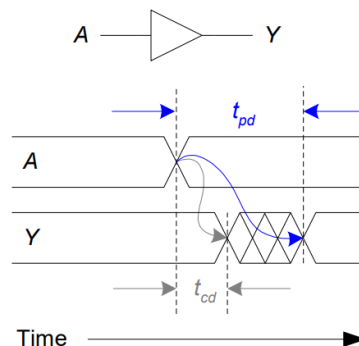
Dekodéry

- Má N vstupov, 2^N výstupov
- One-hot kódovanie: v danom čase je len jeden výstup HIGH



Časové charakteristiky

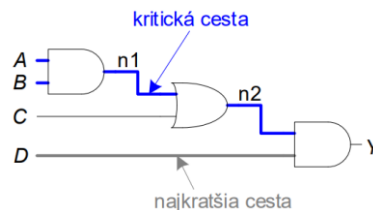
- Dynamické charakteristiky číslicových signálov
- Doba oneskorenia prechodu signálu
- Propagačné oneskorenie: $t_{pd} = \max$ oneskorenie signálu
- Kontaminačné oneskorenie: $t_{cd} = \min$ oneskorenie signálu



Propagačné a kontaminačné oneskorenie

- Oneskorenie je spôsobené: Vlastnosťami vodičov, Rýchlosť pohybu elektrónov

Kritická a najkratšia cesta



Falošné impulzy, hazardy

- Ak jedna zmena vstupného signálu vyvolá viacnásobnú zmenu výstupného signálu
- či vôbec vznikne rozhodujú časové charakteristiky vstupných signálov

Sekvenčné logické obvody (SLO)

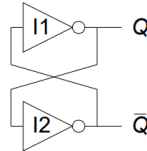
- výstup závisí od kombinácie vstupov a od vnútorných stavov obvodu z predchádzajúceho taktu, má pamäť
- Definície:
 - Stav: konfigurácia nezávislých a závislých veličín
 - Preklápacie obvody: uchováva jednobitovú informáciu
 - Sekvenčný obvod: pozostáva z KLO a z preklápacích obvodov
- výstupný vektor je závislý od sekvencie vektorov na vstupe v predchádzajúcich taktoch
- sekvenčný logický systém obsahuje podsystemy, ktoré sú schopné pamätať si informáciu z predchádzajúcich taktov

Obvody určené na pamätanie

- Na uchovávanie vnútorného stavu
- Bistabilné obvody
- Preklápacie obvody:
 - Hladinou riadené a hranou riadené (angl. flip-flop)
 - Synchronné a asynchronné
 - D, T, SR, JK

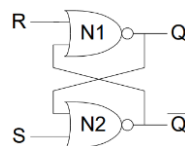
Bistabilný obvod

- Základ obvodov určených na pamätanie info., Má dva stabilné stavy: Q, \bar{Q}
- Uchováva 1 bit informácie pomocou stavovej premennej Q, Nemá žiadny vstup



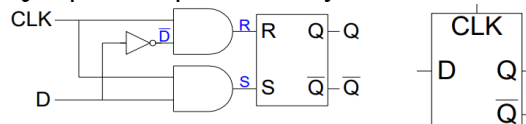
SR (Set/Reset) preklápací obvod

- Uchováva 1 bit informácie (Q) o stave systému
- Na kontrolu sa používajú vstupy S a R: Set: Nastaví výstup na 1, Reset: Nastaví výstup na 0, Je potrebné zabrániť kombinácií $S = R = 1$



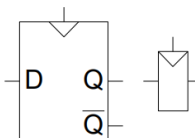
D PO

- Dva vstupy: CLK, D
 - CLK - definuje moment vzorkovania vstupu
 - D - na akú hodnotu má byť nastavený výstup
- Funkcia
 - ak CLK = 1 , D sa dostane na Q
 - ak CLK = 0 , Q si pamätá predchádzajúcu hodnotu



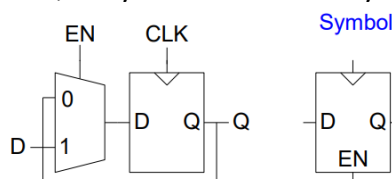
D Flip-Flop (D FF)

- Vstupy: CLK, D
- Funkcia:
 - Vzorkuje D pri prechode CLK z jednej úrovně na druhú
 - Q sa mení len pri zmene CLK
- Hranou riadený PO



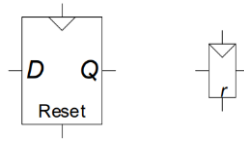
D FF so vstopom EN

- Vstupy: CLK, D, EN
- Vstup *EN*able riadi moment, kedy sa dostane dátový vstup *D* na výstup



D FF s možnosťou a. resetu

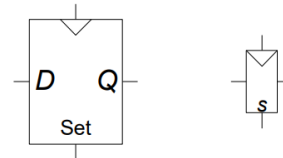
- Vstupy: CLK, D, Reset



- Dva typy:
 - Synchronne: len pri zmene CLK
 - Asynchronne: hneď čo sa nastaví $Reset = 1$

D FF s možnosťou nastavovania

- Vstupy: CLK, D, Set
- Funkcia
 - **Set = 1**: $Q = 1$
 - **Set = 0**: pracuje v „normálnom“ režime



Sekvenčný LO

- Bez vstupov 3 „vnútorné“ signály, Astabilný obvod → osciluje, Má to slučku

Synchronný sekvenčný LO

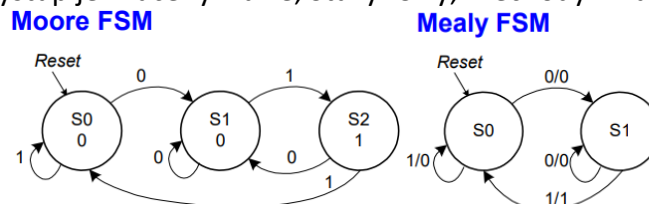
- vkladáme registre, uchovávajú info. o vnútornom stave systému, Stav sa mení pri zmene CLK: systém je synchronizovaný s hodinovým signálom

Konečno-stavové automaty (KSA)

- Obsahuje: Stavový register, Kombinačná logika
- Ďalší stav je definovaný aktuálny vnútorným stavom systému a momentálnym stavom na vstupe
- Dva typy:
 - Automat Moore: výstup je závislý len od vnútorného stavu
 - Automat Mealy: výstup je závislý tak od vnútorného stavu ako aj od vstupe

Graf prechodov

- KSA Moore: výstup je značený v uzle, Stavý: Uzly, Prechody: Hrany



Kódovanie stavov v KSA

- Binárne kódovanie
- One-hot kódovanie

Faktorizácia KSA

- Rozdelenie „veľkého“ KSA na menšie interagujúce KSA

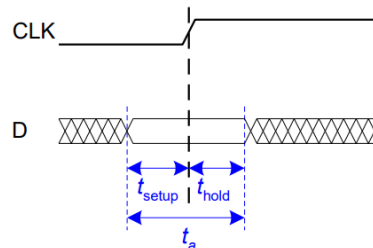
Časové charakteristiky SLO

- Hranou riadený PO
 - je nutné dodržať niektoré dynamické parametre vstupných signálov:
 - doba predstihu
 - doba presahu
 - minimálna strmosť hrany hodinového impulzu
 - minimálna doba jeho trvania
 - Nedodržanie môže vyvolať metastabilný stav
- Základné vlastnosti hranou riadeného PO:

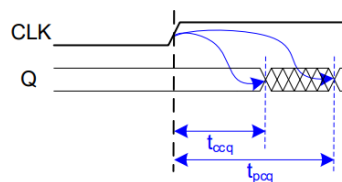
- Rozhodujúci je stav na vstupoch v tesnom okolí nábežnej hrany
- Metastabilný stav má za následok nespoľahlivú funkciu preklápacieho obvodu

Časovanie signálov v SLO

- Doba nastavenia: t_{setup} = časový úsek pred zmenou CLK, počas ktorého vstup by mal byť stabilný
- Doba presahu: t_{hold} = časový úsek po zmene CLK, počas ktorého vstup by mal byť stabilný
- Vzorkovací interval: t_a = v tomto časovom intervale sa nesmie zmeniť hodnota vstupu ($t_a = t_{setup} + t_{hold}$)



- Propagačné oneskorenie: t_{pcq} = časový úsek po uplynutí ktorého je garantované, že Q má stabilnú hodnotu
- Kontaminačné oneskorenie: t_{ccq} = časový úsek po uplynutí ktorého sa začína prejavovať zmena na Q



- Vstupy do SLO musia byť stabilné počas vzorkovania:
 - minimálne počas doby t_{setup} pred zmenou CLK
 - minimálne počas doby t_{hold} po zmene CLK
- Signál, kt. prechádza cez KLO medzi registrami je charakterizovaný kontaminačným a propagačným oneskorením

Doba nastavenia

- Závisí od propagačného oneskorenia signálu

Doba presahu

- Závisí od kontaminačného oneskorenia signálu

Odchýlka v časovaní

- Hodinový impulz prichádzajúci na vstupy rôznych registrov nie v rovnakom momente

Porušenie dynamickej disciplíny

- Asynchrónne vstupy môžu mať za následok nepredvídateľné správanie sa obvodu

Metastabilný stav

- Bistabilné zariadenia: dva stabilné stavy a jeden metastabilný stav
- Flip-flop: dva stabilné stavy (1 a 0) a jeden metastabilný stav

Paralelizmus

- Token: dátový vstup
- Doba odozvy: čas prechodu tokenu log. systémom
- Priepustnosť: počet tokenov spracovaných za jednotku času
- Paralelizmus zvyšuje priepustnosť

- Dva typy paralelizmu:
 - Priestorový paralelizmus - Zvýšenie počtu funkčných jednotiek
 - Temporálny paralelizmus - Úloha sa rozdelí na dielčie kroky, ktoré sa vykonávajú súbežne

Návrh LS

- Jazyk pre opis hardvéru (Hardware description language; HDL)
- HDL nie je programovací jazyk, je to jazyk, ktorý slúži na opis HW
- Dva hlavné HDL:
 - VHDL (VHDL 2008)
 - Verilog (SystemVerilog)
- Syntéza - Prevod HDL kódu do netlist reprezentácie
- Simulácia - Verifikácia činnosti obvodu

VHDL moduly

- Spôsoby opisu LS:
 - Behaviorálny: na čo slúži / z pohľadu chovania
 - Štruktúrálny: modulárny opis / z akých komponentov pozostáva
 - Tok dát (dataflow): tok dát medzi komponentmi
- VHDL Je „case insensitive“

Najvyššia	not	not
	*, /, mod, rem	mult, div, mod
	+, -	add, sub
	rol, ror	rotate
	srl, sll	shift logical
	<, <=, >, >=	relative comparison
	=, /=	equal, not equal
	and, nand	AND, NAND
	xor, xnor	XOR, XNOR
Najnižšia	or, nor	OR, NOR

Sekvenčná logika

- VHDL používa návrhové vzory na modelovanie správania sa preklápacích obvodov a stavových automatov
- Príkaz process - Príkazy vo vnútri bloku process sú vykonané len vtedy, ak je zaznamenaná zmena aspoň jedného signálu
- Sekvenčný príkaz je súčasťou príkazu/bloku process: if / else, case, case?, loop
- Na modelovanie kombinačných logických obvodov (KLO) je vhodnejšie použiť konkurenčné (paralelné) príkazy
- Príkaz case vedie na syntézu KLO len vtedy ak sme pokryli všetky možné vstupné kombinácie
- Príkaz case? na rozdiel od „obyčajného“ case dokáže pracovať aj s „don't care“ hodnotami
- SLO: Použi príkaz process(clk) a neblokujúce priradenie (<=) pre modelovanie SLO
- KLO: Použi konkurenčné (paralelné) príkazy na tvorbu KLO

Konečno-stavové automaty(FSMs)

- Tri bloky: Vstupný logický obvod, Stavový register, Výstupný logický obvod

Parametrizované moduly

- Kľúčové slovo generic - generická konštanta vo vnútri bloku entity
- Kľúčové slovo generate - Genereujúci príkaz

Simulácia / testbench súbory

- Na simuláciu modulov použijeme testbench súbor
- Testbench nie je určený na syntézu
- Typy:
 - Jednoduchá funkčná
 - S autokontrolou
 - S autokontrolou na báze testovacieho súboru s testovacími vektormi
- Hodinový signál:
 - Inicializuj vstupy (počas nábehovej hrany)
 - Porovnaj výstup s očakávanými výstupmi (počas dobehovej hrany)
- Hodinový signál je nevyhnutný v prípade synchronných sekvenčných obvodov

Komponenty číslicových systémov:

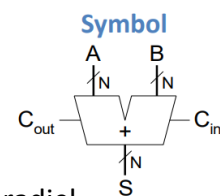
- hradlá, multiplexory, dekodéry, obvody na realizáciu aritmetických a logických operácií, počítadlá, registre, pamäte, ...

Pri návrhu a kompozícií sa uplatňujú princípy HMJ

- Rozdelenie systémov do modulov
- Moduly majú dobre zadefinované funkcie a rozhrania
- Možnosť znovupoužitia

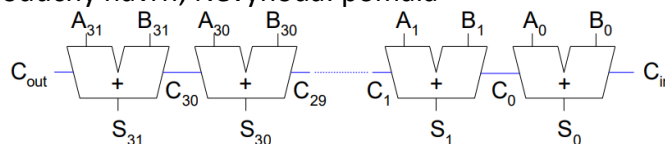
Sčítačky

- Sčítanie dvoch N-bitových čísel
 - Sčítačka so sériovým prenosom (pomalá)
 - Sčítačka so zrýchleným prenosom (rýchlejšia)
 - Paralelné prefixové sčítačky (najrýchlejšia)
- Sčítačky so ZP a prefixové sčítačky sú rýchle ale potrebujú viac hradíel



Sériová sčítačka

- Kaskáda 1b sčítačiek, Prenos sa šíri od najnižšieho až po najvyšší rád
- Výhoda: jednoduchý návrh, Nevýhoda: pomalá



Sčítačka so zrýchleným prenosom

- Výpočet prenosu (C_{out}) z k-bit veľkého bloku pomocou vlastného a tranzitívneho prenosu
- Prenos z i-teho rádu nastane vtedy ak je to dané vlastným prenosom (G_i) alebo v súčinnosti s indikovaným príznakom tranzitívneho prenosu (P_i)

32b:4b CLA

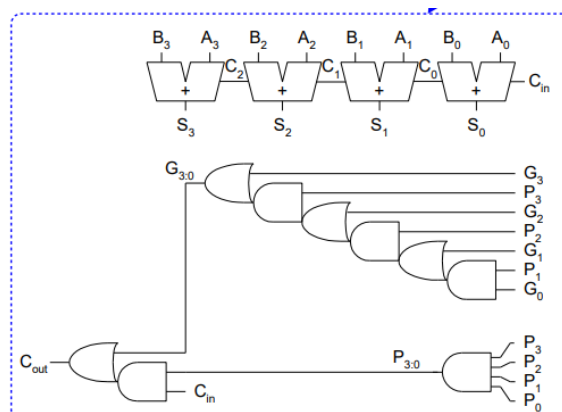
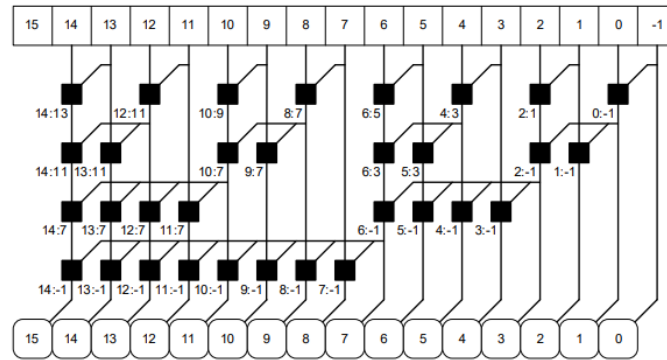
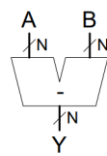


Schéma prefixovej sčítacky

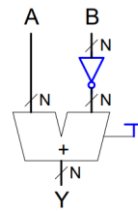


Odčítačka

Symbol

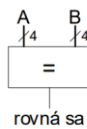


Implementácia

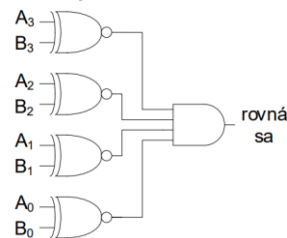


Komparátor : rovnosť

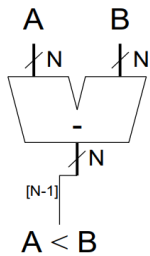
Symbol



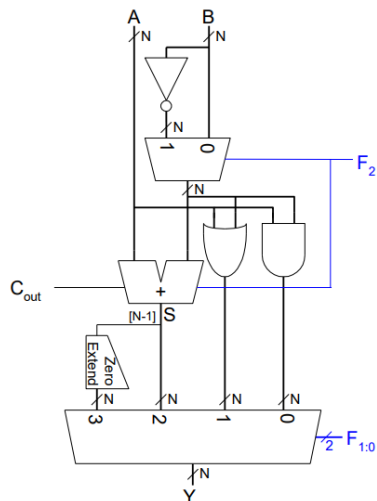
Implementácia



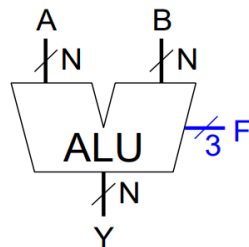
Komparátor : menší



Jednoduchá ALU



Aritmeticko-logická jednotka (ALU)



Posuvné (funkčné) jednotky (PFJ)

- Logický posun: posun doprava alebo doľava s pridaním 0

$11001 \gg 2 = 00110$

$11001 \ll 2 = 00100$

- Aritmetický posun: posun doľava je ako logický posun doľava; pri posune doprava sa kopíruje hodnota msb.

$11001 \ggg 2 = 11110$

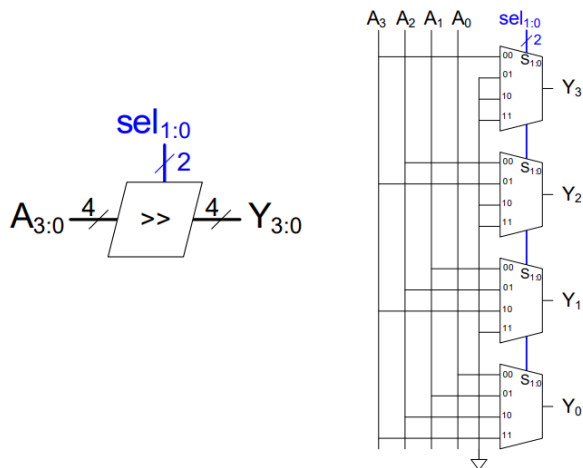
$11001 \lll 2 = 00100$

- Rotácia: bit ktorý sa pri rotácii dostane mimo rámec zobrazenia sa objaví v pozícií msb (doprava) alebo lsb (doľava)

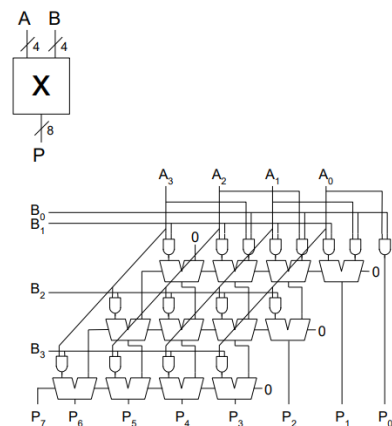
$11001 \text{ ROR } 2 = 01110$

$11001 \text{ ROL } 2 = 00111$

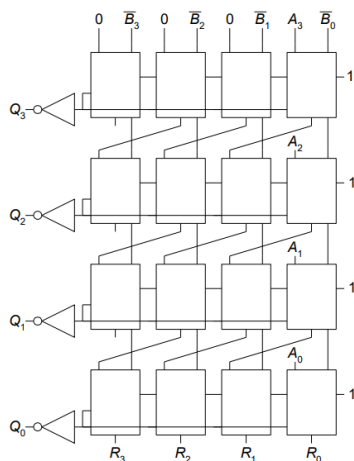
Návrh posuvnej jednotky (KLO)



Násobička 4 x 4



Delička 4 x 4



Reprezentácia čísel

- ČP používa binárnu č. sústavu
 - Prirodzené čísla
 - Beznamienkové binárne čísla
 - Celé čísla
 - Priamy kód
 - Doplňkový kód
 - Kód s posunutím

- Reálne čísla
 - Pevná rádová čiarka (PRČ) – Q notácia
 - Pohyblivá rádová čiarka (PHRČ)

Pevná rádová čiarka

- Vyjadri 6,75 v Q4.4 (= 4 bity pre celú časť vrátane znamienka a 4 pre desatinnú časť):

$$01101100$$

$$0110,1100$$

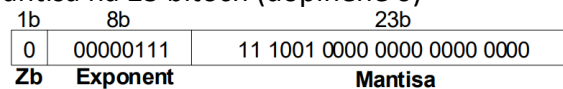
$$2^2 + 2^1 + 2^{-1} + 2^{-2} = 6.75$$
- Pozícia desatinnej čiarky je striktno určená
- Je potrebné sa dohodnúť na počte bitov vyhradených pre celú a pre desatinnú časť čísla

Pohyblivá rádová čiarka

- Čísla v pohyblivej rádovej čiarky sa vyjadrujú v tvare $\pm M \times b^E$
- M = mantisa, b = základ, E = exponent
- napr. : $273 = 2,73 \times 10^2$



- Príklad: Vyjadrite reálne číslo 228,010 v 32-bitovom formáte
 - Binárna reprezentácia: $228_{10} = 11100100_2$
 - Vedecký binárny zápis: $11100100_2 = 1,11001_2 \times 2^7$
 - 32-bitovú reprezentáciu čísla v pohyblivej rádovej čiarky
 - Znamienkový bit = 0 (kladné)
 - 8-bitový exponent = 7
 - Mantisa na 23 bitoch (doplnené 0)



IEEE-754 formáty

- Formát jednoduchšej presnosti:
 - 32 bitov
 - 1b znamienko, 8b charakteristika, 23b mantisa
 - bias = 127
- Formát dvojitej presnosti:
 - 64 bitov
 - 1b znamienko, 11b charakteristika, 52b mantisa
 - bias = 1023

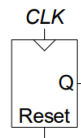
PHRČ: zaokrúhľovanie

- Pretečenie: číslo je príliš veľké pre zobrazenie v danom formáte
- Podtečenie: číslo je príliš malé pre zobrazenie v danom formáte
- Techniky zaokrúhľovania:
 - Nadol (down)
 - Nahor (up)
 - Smerom k nule (toward zero)
 - Na najbližšiu hodnotu (to nearest)

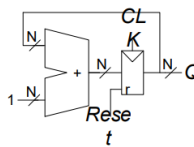
Počítadlo

- Hodnota sa inkrementuje/dekrementuje pri každom takte
- Prípady použitia:
 - Riadenia synchronizačných impulzov v digitálnych zobrazovačoch
 - Programové počítadlo

Symbol



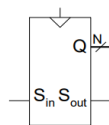
Implementácia



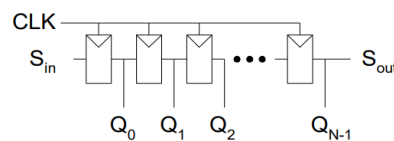
Posuvný register

- Pri každom takte vstupuje do a vystupuje z registra 1b
- Prevod sériového kódu na kód paralelný

Symbol:

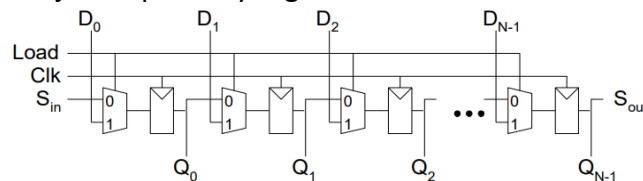


Implementácia



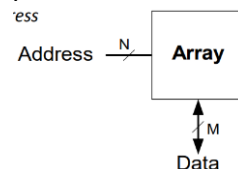
Posuvný register s paralelným nastavením

- Ak Load = 1, pracuje ako (bežný) N-bitový register
- Ak Load = 0, pracuje ako posuvný register

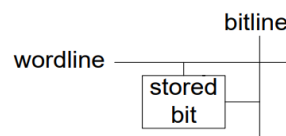


Pamäť

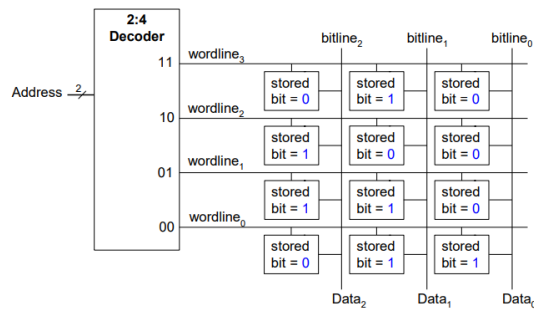
- Slúži na pamätanie údajov
- 3 základné typy:
 - Dynamic random access memory (DRAM)
 - Static random access memory (SRAM)
 - Read only memory (ROM)
- M-bit široký dátový výstup/vstup Data pre čítanie/zápis do pamäte na adresu definovanú N-bit širokým vstupom Address



- 2D pole pamäťových buniek, Každý element dokáž uchovať 1b údaj
- realizácia pamäti



- Adresná linka (wordline)
 - unikátna adresa slova, jeden riadok v pamäti pre čítanie/zápis



Typy pamäti

- Random access memory (RAM) - volatilná pamäť, energeticky závislá
- Read only memory (ROM) - nevolatilná pamäť, energeticky nezávislá

RAM: s náhodným prístupom

- Stráca obsah po odpojení elektrickej energie
- Čítanie a zápis sú relatívne rýchle operácie
- Hlavná pamäť počítača je typu RAM (DRAM)

ROM: určené len na čítanie

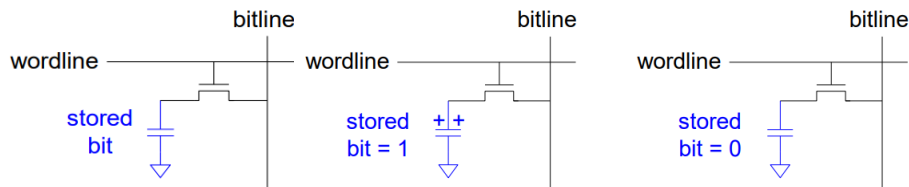
- NVRAM = Nestráca obsah po odpojení elektrickej energie
- Rýchle čítanie, zápis je pomalý, BIOS, PROM, EPROM, EAROM, EEPROM, Flash pamäť

Pamäte RAM

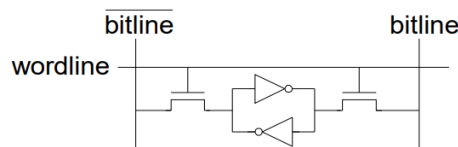
- DRAM (Dynamic random access memory)
- SRAM (Static random access memory)
- Rozdiel je v spôsobe ukladania dát:
 - DRAM používa kondenzátory
 - SRAM používa dvojicu invertorov v slučke

DRAM

- Konštrukčným základom bunky je kondenzátor

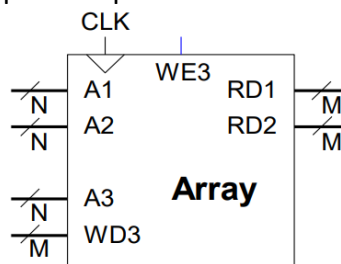


SRAM



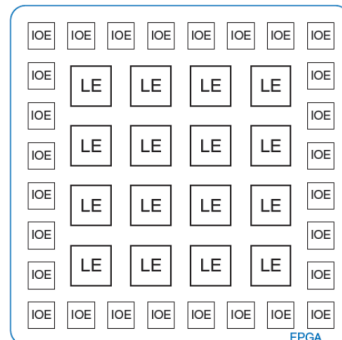
Viacportové pamäte

- Port: zabezpečuje vstup/výstup
- Súbor registrov: malá viacportová pamäť



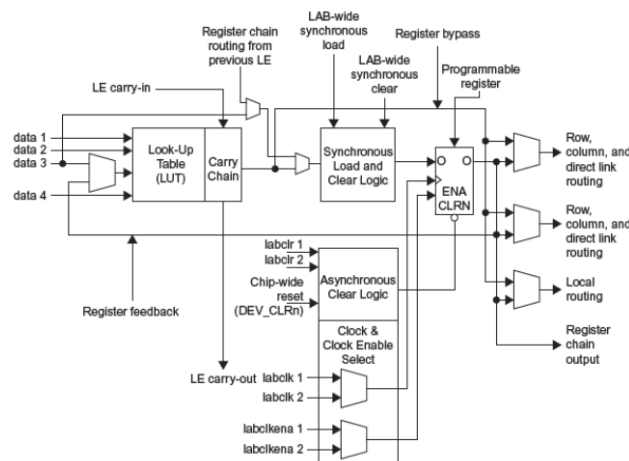
Logické polia

- PLAs (Programmable logic arrays)
 - Pole AND členov napojené na pole OR členov, Kombinačná logika
- FPGAs (Field programmable gate arrays)
 - Pole logických elementov, Kombinačná/sekvenčná logika, Reprogramovateľné
 - Pozostáva:
 - LE (Logic elements): realizácia logiky
 - IOE (Input/output elements): rozhranie pre komunikáciu s vonkajším svetom
 - Prepojovacia sieť: spája LE a IOE



- LE: Logický element pozostáva
 - LUT (lookup tables)
 - Preklápacie obvody (flip-flops)
 - Multiplexory

Altera Cyclone IV LE



ISA: inštrukčnoorientovaná architektúra

- abstrakcia prostredníctvom súboru inštrukcií

IA: implementačnoorientovaná architektúra

- abstrakcia na úrovni štruktúrnej organizácie

Asembler

- Asembler: jazyk symbolických inštrukcií
- Strojový jazyk/kód: inštrukcie sú v podobe binárneho reťazca
- MIPS architektúra
 - Vývoj 80-tych rokov na Stanfordskej univerzite pod vedením Johna Hennessyho
 - Použitá v zariadeniach od Silicon Graphics, Nintendo, a Cisco

Princípy návrhu počítačových architektúr

- Hennessy a Patterson navrhujú postupovať pri návrhu architektúry tak, aby boli dodržané nasledujúce odporúčania:
 - Jednoduchosť podporuje jednotnosť
 - Jadro musí byť rýchle
 - Zníženie počtu komponentov zvyšuje rýchlosť
 - Dobrý návrh vychádza z kompromisov

Inštrukcia: sčítavanie – add

Inštrukcia: odčítavanie – sub

Adresovanie

- Každé 32b slovo má svoju unikátnu adresu

Word Address	Data	
...
00000003	4 0 F 3 0 7 8 8	Word 3
00000002	0 1 E E 2 8 4 2	Word 2
00000001	F 2 F 1 A C 0 7	Word 1
00000000	A B C D E F 7 8	Word 0

Čítanie z pamäte – load

Zápis do pamäte – store

Organizácia zobrazovania objektov v pamäti

- Zarovnávanie viacbajtových objektov
 - Usporiadanie bajtov objektov v informačných slovách
- Ukladací endian - Malý endian, Veľký endian

Adresovanie slov po bajtoch

- Každý bajt v slove má svoju unikátnu adresu

Word Address	Data	
...
0000000C	4 0 F 3 0 7 8 8	Word 3
00000008	0 1 E E 2 8 4 2	Word 2
00000004	F 2 F 1 A C 0 7	Word 1
00000000	A B C D E F 7 8	Word 0

width = 4 bytes

- MIPS používa bajtovú organizáciu pamäte pri ukladaní 32b slov

Čítanie slova z pamäti

MIPS kód

lw \$s3, 4(\$0)

Word Address	Data	
...
0000000C	4 0 F 3 0 7 8 8	Word 3
00000008	0 1 E E 2 8 4 2	Word 2
00000004	F 2 F 1 A C 0 7	Word 1
00000000	A B C D E F 7 8	Word 0

width = 4 bytes

Zápis slova do pamäti

MIPS kód

sw \$t7, 44(\$0)

Word Address	Data	
...
0000000C	4 0 F 3 0 7 8 8	Word 3
00000008	0 1 E E 2 8 4 2	Word 2
00000004	F 2 F 1 A C 0 7	Word 1
00000000	A B C D E F 7 8	Word 0

width = 4 bytes

Ukladací endian

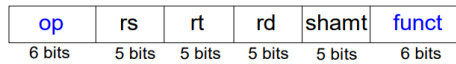
- Malý endian: v smere od najnevýznamnejších (LSB) po najvýznamnejšie bajty (MSB)
- Veľký endian: v smere od jeho najvýznamnejších bajtov po najnevýznamnejšie bajty

Big-Endian				Little-Endian			
Byte Address		Word Address		Byte Address		Word Address	
...		
C D E F		C		F E D C			
8 9 A B		8		B A 9 8			
4 5 6 7		4		7 6 5 4			
0 1 2 3		0		3 2 1 0			
MSB	LSB			MSB	LSB		

Strojový jazyk / kód

- Binárna reprezentácia inštrukcie
- 3 inštrukčné formáty (IF):
 - Typ R: operandy sú v registroch

R-Type



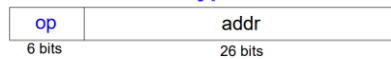
- Typ I: používa priamy operand

I-Type



- Typ J: vetvenia

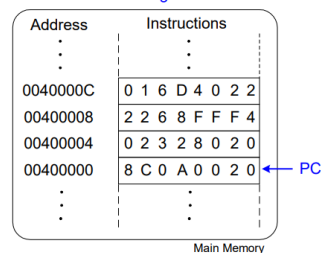
J-Type



Program v pamäti

- Inštrukcia a aj dáta sú v pamäti
- Rozdielna sekvencia inštrukcií = rozdielny program
- Spracovanie programu:
 - Procesor číta inštrukcie z pamäte počítača
 - Procesor vykonáva inštrukcie
- Programové počítadlo (PC): definuje poradie spracovania inštrukcií

Stored Program



Logické inštrukcie

- and, or, xor, nor
 - and: používa sa pri maskovaní bitov
 - or: používa sa pri rekombinácii bitov
 - nor: používa sa pri invertovaní bitov
- andi, ori, xori

Inštrukcie posuvu

- sll/sllv: logický posuv doľava
- srl/srlv: logický posuv doprava
- sra/srav: aritmetický posuv doprava

Vetvenie

- Ovplyvňuje poradie vykonania inštrukcií
- Typy vetvení:
 - Podmienené – beq, bne
 - Nepodmienené – j, jr, jal

Polia

- Kompozitná dátová štruktúra
- Index: určuje pozíciu prvku v poli
- Veľkosť: počet prvkov v poli

ASCII kód

- American Standard Code for Information Interchange
- Každý znak má svoj unikátny kód

Práca s funkciami

- Volajúci:
 - odovzdáva argumenty volanej funkcie
- Volaný:
 - Vykoná funkciu, Vráti výsledok, Zabezpečí návrat, Nesmie prepísať obsah registrov a pamäťových miest

Zásobník

- Pamäť pre dočasné premenné, Je typu LIFO (last-in-first-out)
- Zväčšuje sa dynamicky: Ak potrebujem ukladať viac dát
- Zmenšuje sa dynamicky: Ak už nepotrebujem tie dáta
- Rastie smerom nadol
- Smerník zásobníka: \$sp ukazuje na vrchol zásobníka

Address	Data		Address	Data
7FFFFFFC	12345678	← \$sp	7FFFFFFC	12345678
7FFFFFF8			7FFFFFF8	AABBCCDD
7FFFFFF4			7FFFFFF4	11223344
7FFFFFF0			7FFFFFF0	
⋮	⋮		⋮	⋮

Zásobník počas rekurzie

Address	Data		Address	Data		Address	Data	
FC		← \$sp	FC		← \$sp	FC		← \$sp \$v0 = 6
F8			F8	\$a0 (0x3)		F8	\$a0 (0x3)	
F4			F4	\$ra	← \$sp	F4	\$ra	← \$sp \$a0 = 3 \$v0 = 3 x 2
F0			F0	\$a0 (0x2)		F0	\$a0 (0x2)	
EC			EC	\$ra (0xBC)	← \$sp	EC	\$ra (0xBC)	← \$sp \$a0 = 2 \$v0 = 2 x 1
E8			E8	\$a0 (0x1)		E8	\$a0 (0x1)	
E4			E4	\$ra (0xBC)	← \$sp	E4	\$ra (0xBC)	← \$sp \$a0 = 1 \$v0 = 1 x 1
E0			E0			E0		
DC			DC			DC		

Adresovacie módy

- Registrový mód - Operand je uložený v registri
- Bezprostredný mód - 16b konštanta je súčasťou inštrukcie
- Posúvací mód - Adresa operandu sa počíta ako: Bázová adresa + posuv
- PC-Relatívny mód

Assembly Code

```
beq $t0, $0, else
(beq $t0, $0, 3)
```

Field Values

op	rs	rt	imm
4	8	0	3
6 bits	5 bits	5 bits	5 bits 6 bits

- Nepriamy indexovací mód

Field Values

op	imm
3	0x0100028
6 bits	26 bits

Machine Code

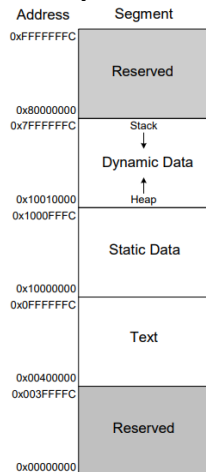
op	addr
000011	00 0001 0000 0000 0000 0010 1000
6 bits	26 bits

(0x0C100028)

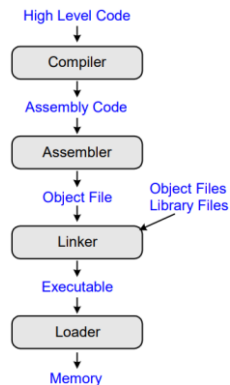
Čo je uložené v pamäti?

- inštrukcie, dáta

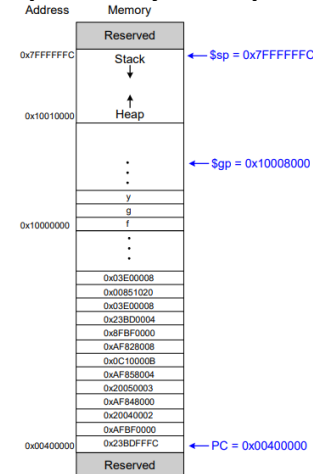
MIPS pamäť



Preklad & spúšťanie programu



Spustiteľný kód v pamäti



Mikroarchitektúra

- abstrakcia na úrovni štruktúrnej organizácie

Procesor

- Tok dát/operandov: spájanie funkčných jednotiek
- Tok riadenia: distribúcia riadiacich signálov

Mikroarchitektúra

- Rôzne IA pre tú istú ISA:
 - Jednocyklová: Každá inštrukcia sa vykoná za jeden strojový cyklus
 - Viaccyklová: Inštrukcie sa vykonávajú za jeden alebo viac cyklov
 - Prúdová: Vykonanie inštrukcie sa rozloží na dielčie kroky

Výkonnostné parametre

- Čas vykonania programu
- Definícia: CPI: počet SC procesora v priebehu ktorých sa vykoná jedna inštrukcia
- Výzvou je nájsť správnu rovnováhu medzi parametrami: cena – spotreba – výkon

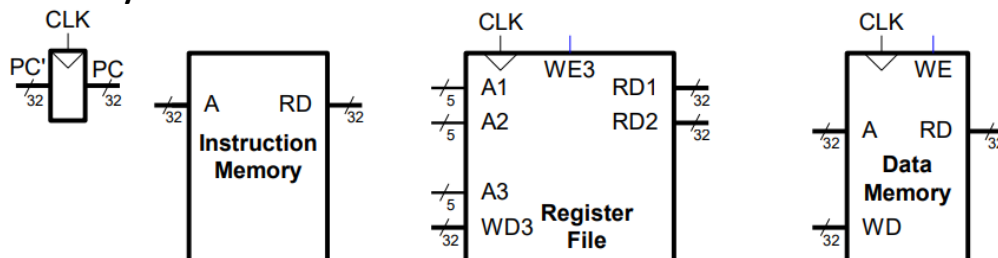
MIPS Procesor

- Nech je daná podmnožina MIPS inštrukčnej sady:
 - Inštrukcie typu R: and, or, add, sub, slt
 - Inštrukcie pre prácu s pamäťou: lw, sw
 - Inštrukcia vetvenia: beq

Vnútrotný stav architektúry

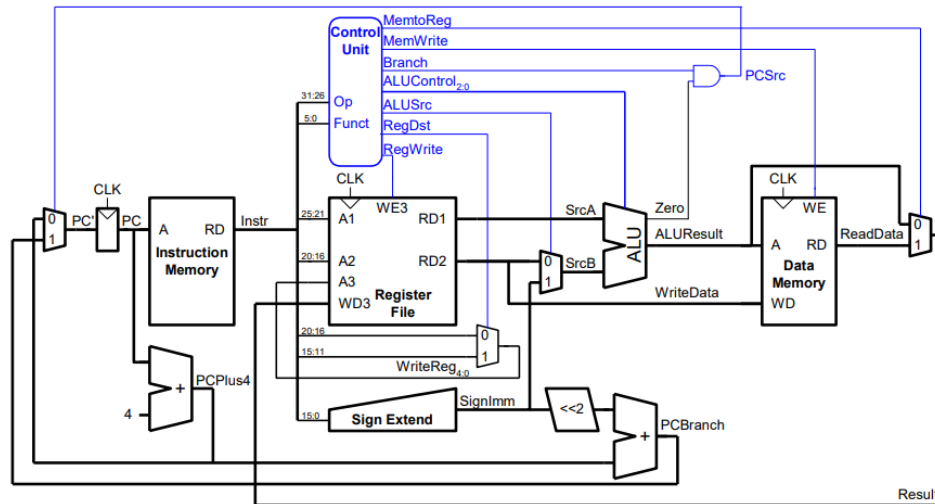
- V ktorom stave sa architektúra nachádza závisí od:
 - Hodnoty programového počítadla (PC)
 - Hodnôt uchovaných v registroch architektúry
 - Obsahu pamäte

Stavové elementy MIPS

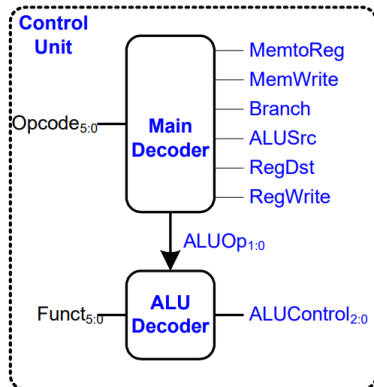


Jednocyklový MIPS procesor

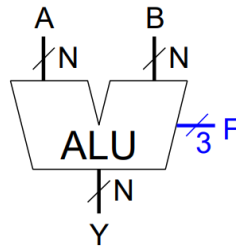
- Návrh procesora vychádza z návrhu jeho komponentov a prepojujacej sústavy, pričom sa zohľadňuje
 - tok dát/operandov medzi funkčnými jednotkami procesora
 - tok riadenia
- čítanie inštrukcie:



- riadenie

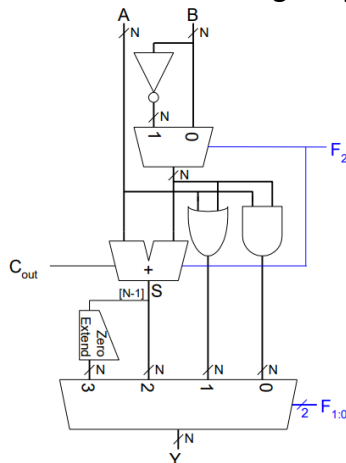


Aritmeticko-logická jednotka



F _{2:0}	Funkcia
000	A & B
001	A B
010	A + B
011	not used
100	A & ~B
101	A ~B
110	A - B
111	SLT

- Aritmeticko-logická jednotka



- Výkon jednocyklového procesora

- Kritická cesta:

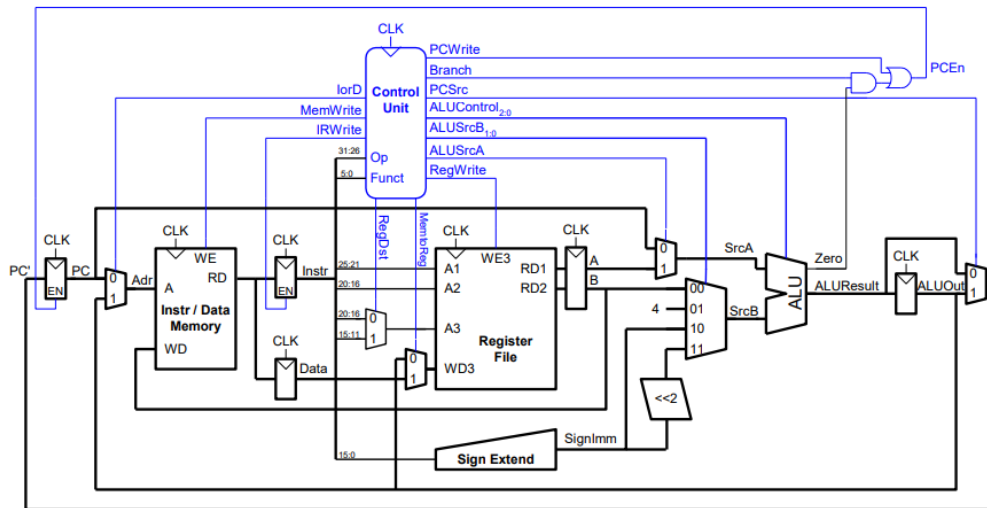
$$T_c = t_{pcq_PC} + t_{mem} + \max(t_{Rfread}, t_{sext} + t_{mux}) + t_{ALU} + t_{mem} + t_{mux} + t_{Rfsetup}$$

- o vo väčšine prípadov
pamäť, ALU, súbor registrov

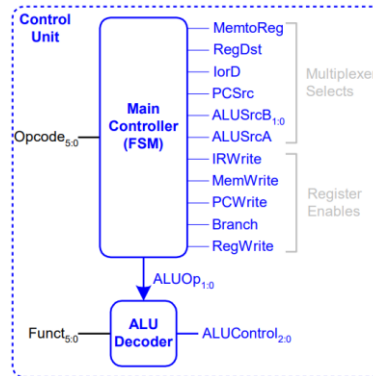
$$T_c = t_{pcq_PC} + 2t_{mem} + t_{Rfread} + t_{mux} + t_{ALU} + t_{Rfsetup}$$

Viaccyklový MIPS procesor

- + vyššia taktovacia frekvencia
- + jednoduchšie inštrukcie vyžadujú kratší čas
- + znovupoužiteľnosť niektorých funkčných jednotiek počas realizácie inštrukcie
- - zložitejšie riadenie
- Podobný postup: návrh smerovania toku dát/operandov & smerovania riadiacich signálov



- Riadiaca jednotka viaccyklového MIPS procesora



- Výkon viaccyklového procesora

- o Kritická cesta

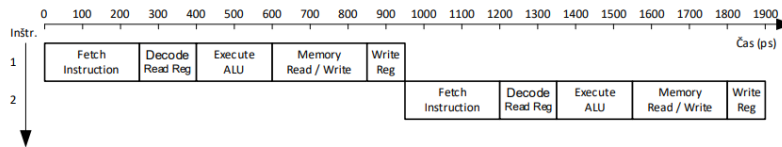
$$T_c = t_{pcq} + t_{mux} + \max(t_{ALU} + t_{mux}, t_{mem}) + t_{setup}$$

Prúdový MIPS procesor

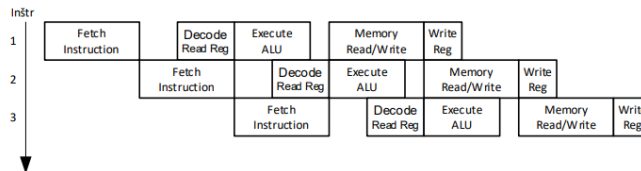
- Temporálny paralelizmus
- Rozdeľ vykonanie inštrukcie v jednocyklovom procesore na 5 fáz:
 - o Čítanie inštrukcie / Fetch
 - o Dekódovanie inštrukcie / Decode
 - o Vykonanie inštrukcie / Execute
 - o Pamätanie výsledku / Memory
 - o Zápis do pamäte / Writeback
- Pridaj záchytné registre medzi jednotlivé stupne zreťazenia

Jednocyklový vs. prúdový MIPS

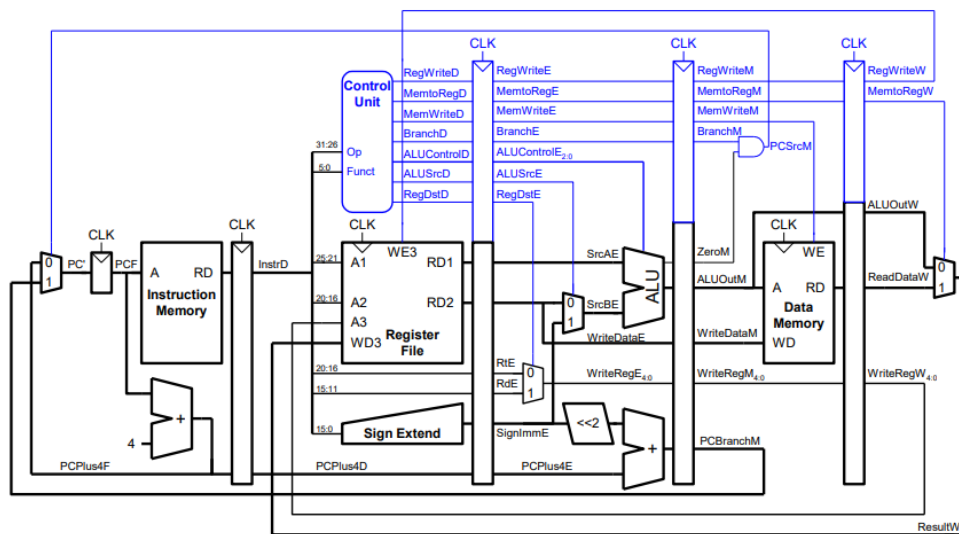
Jednocyklový MIPS



Zreťazený MIPS

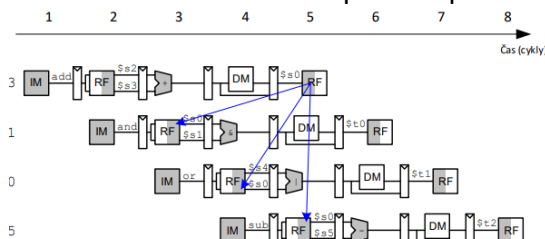


Riadenie prúdového MIPS

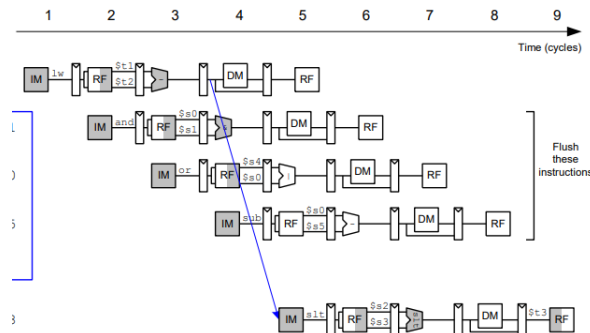


Hazardy prúdového spracovania

- Ak vykonanie inštrukcie je závislé od inštrukcie, ktorá ešte nebola dokončená
- Typy:
 - **Údajový hazard:** v dôsledku nedostupnosti operandov pri vykonaní inštrukcií



- eliminácia: sw prostriedky – vloženie NOP po inštrukcií, preusporiadanie inštrukcií, hw prostriedky – blokovanie prúdového spracovania, dopredné generovanie výsledku predchádzajúcej inštrukcie
- **Hazard riadenia:** v dôsledku vetvení
- Pokutovanie nesprávneho odhadu skoku
 - Vyraďovanie inštrukcií z procesu spracovania ak sa skok uskutoční
 - Predikovanie skoku



- **Zdrojový hazard:** nie sú dostupné zdroje na vykonanie inštrukcie

Moderné mikroarchitektúry

- „Hlboké“ prúdové funkčné jednotky: 10-20 stupňov
- Špekulatívne vetvenie: Ideálny prúdový procesor: CPI = 1, Zlá predikcia zvyšuje CPI
 - Statická predikcia vetvenia: Kontrola smeru skoku
 - Dynamická predikcia vetvenia: Na základe histórie skokov
- Superskalárne procesory : Viac inštrukcií je spracovaných naraz, Hazardy môžu byť problém
- Vykonávanie mimo poradia: Analýza niekoľkých inštrukcií, Priprav čo najviac inštrukcií na spracovanie, Spracuj inštrukcie aj mimo poradia
 - závislosti:
 - RAW (read after write): údajová závislosť
 - WAR (write after read): údajová antizávislosť
 - WAW (write after write): výstupná závislosť
 - Plánovanie spracovania inštrukcií
 - Instruction level parallelism (ILP)
 - Prehľadávací obvod (Scoreboard)
- Premenovávanie registrov
- SIMD : Single Instruction Multiple Data
 - Jeden prúd inštrukcií spracováva viacnásobný tok dát
- Multivláknové procesory: mikro, nano, ...
- Multiprocesory: jeden čip ale viac procesorov

Proces vs vlákno

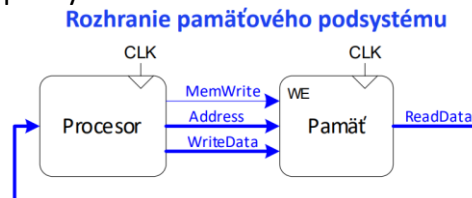
- Proces: program vykonávaný na počítači
 - Je možné spúšťať niekoľko procesov súčasne
- Vlákno: časť programu
 - Proces môže pozostávať z niekoľkých vlákien
- Vlákna v konvenčných procesoroch
 - V danom čase „beží“ len jedno vlákno
 - Ak sa pozastaví činnosť vlákna
 - Uloží sa stav vlákna
 - Načíta a spustí sa druhé (čakajúce) vlákno
 - Volá sa to zmena kontextu
- Multithreading
 - Aktivuje sa niekoľko vlákien súčasne
 - Nezvyšuje mieru paralelizmu na úrovni inštrukcií (ILP) v rámci vlákna, ale zvyšuje priepustnosť

Multiprocessory

- Multiprocessorové systémy s definovanou komunikačnou topológiou
 - Tesne viazané multiprocessory
 - Voľne viazané multiprocessory

Výkonnostné parametre

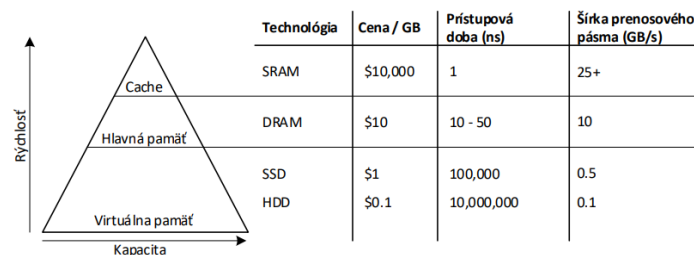
- Výkon PS závisí od
 - Výkonu procesora
 - Pamäťového podsystemu



Návrh pamäťových modulov

- Cieľom je navrhnuť pamäťový podsystem tak rýchly ako je sám procesor
 - Aplikuje sa hierarchická stavba PaP
 - Vlastnosti ideálnej pamäte:
 - rýchla
 - lacná
 - veľká kapacita
- dve sú splniteľné

Hierarchia PaP



Lokalita

- Temporálna lokalita:
 - Časová charakteristika prístupu
- Priestorová lokalita:
 - Priestorová charakteristika prístupu

Výkonnostné parametre

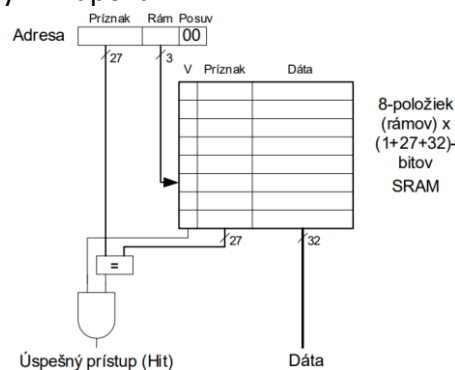
- Úspešné sprístupnenie (hit): údaj sa našiel na danej úrovni PaP
- Neúspešné sprístupnenie (miss): údaj sa nenašiel
- Koeficient úspešnosti (Hit Rate; HR)
 - $HR = \# \text{ úspešné sprístupnenie} / \# \text{ prístupov do pamäte} = 1 - MR$
- Koeficient neúspešnosti (Miss Rate; MR)
 - $MR = \# \text{ neúspešné sprístupnenie} / \# \text{ prístupov do pamäte} = 1 - HR$
- Priemerná doba prístupu do pamäte (AMAT):
 - $AMAT = HR_{cache} t_{cache} + MR_{cache} [HR_{MM} t_{MM} + MR_{MM} (HR_{VM} t_{VM})]$

Gene Amdahl, 1922-

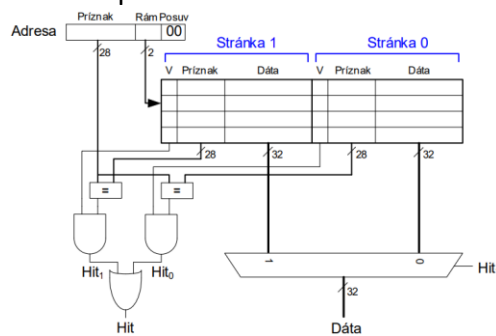
- Amdahlov „zákon“: úsilie vynaložené na zvýšenie výkonu subsystemu minie účinkom, pokiaľ subsystem neovplyvní veľké percento celkového výkonu

Vyrovňavacia pamäť typu „cache“

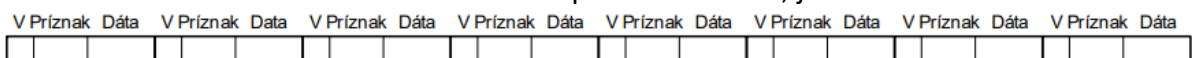
- Pamäť určená na vyrovňovanie rýchlosti prenosu údajov medzi procesorom a hlavnou pamäťou
- Je to rýchla asociatívna pamäť
- Parametre cache
 - Kapacita (C)
 - Veľkosť bloku / skupiny slov v pamäti (b)
 - Počet blokov v pamäti ($B = C/b$)
 - Stupeň asociativity (N)
 - Rám bloku cache ($S = B/N$)
- Ideálne, často používané dáta sú uložené v cache
- Cache sa delí podľa stupňa asociativity:
 - Cache s priamym mapovaním



- N-cestná asociatívna pamäť



- Plne asociatívna cache – zníži počet konfliktov, je to drahé riešenie



- Kapacita pamäte cache
 - Pamäť cache má obmedzenú kapacitu
- Metódy vyradovania
 - Náhodný výber
 - Výber podľa príznaku aktivity
 - Stratégia LRU (Least recently used)
- Viacúrovňová organizácia cache
 - Veľká cache má nižší koeficient neúspešnosti a prístupová doba je väčšia v porovnaní s menšou cache

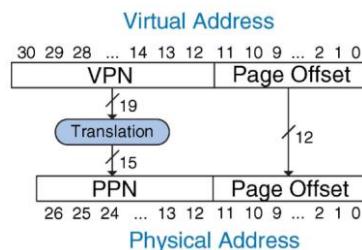
Virtuálna pamäť

- Virtuálnou pamäťou sa získa väčší pamäťový priestor než čo nám ponúka hlavná pamäť
- Virtuálna adresa
 - V programe sa používa virtuálna adresa na určenie miesta kde sa údaj nachádza
- Ochrana pamäte
 - Každý program má vlastnú schému prekladu virtuálnych adres na fyzické
- Cache vs virtuálna pamäť

Cache	Virtuálna pamäť
Blok	Stránka
Neúspešný prístup	Výpadok stránky
Príznak (kľúč) asociatívneho výberu = kópia bázeovej adresy bloku v cache	Virtuálna adresa stránky

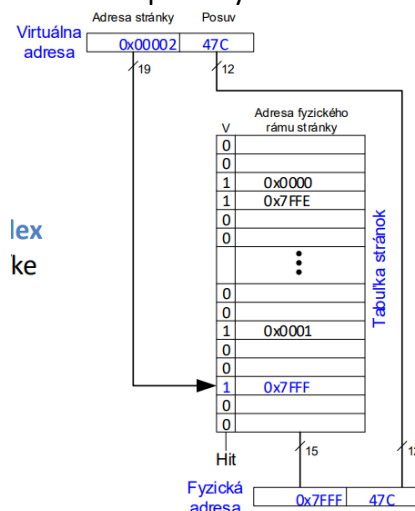
- transformácia virtuálnych adres na fyzické adresy sa nazýva mapovanie adres
- Virtuálna adresa je adresa, ktorou sa identifikuje slovo (blok) programu v logickom adresovom priestore
- Fyzická adresa predstavuje adresu kópie tohto slova (bloku) lokalizovaného v HP
- Existujú dve skupiny VP
 - s pevnou dĺžkou blokov, ktoré sa nazývajú stránky
 - s premenlivou dĺžkou blokov, ktoré sa nazývajú segmenty

Mapovanie pamäte



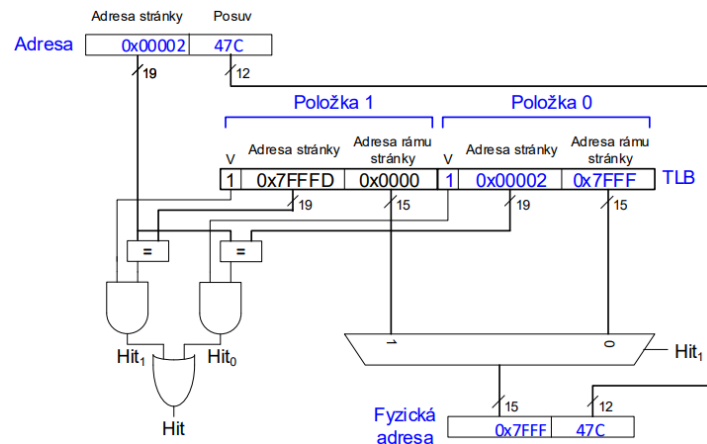
Tabuľka stránok

- Priebežný stav pamäte ČP v procese mapovania je registrovaný pomocou tabuľky stránok
- Adresa stránky funguje ako index položky v tabuľke stránok



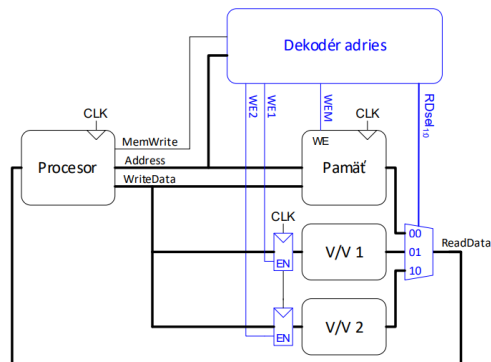
Pamäť preložených adries (TLB) - Translation Lookaside Buffer

- mapovanie prostredníctvom pamäte preložených adries
- sú uložené adresy najčastejšie používaných stránok
- Ukážka TLB



Vstupno-výstupný podsystem

- Každé V/V zariadenie má svoju adresu
- Čítanie/zápis z/do V/V zariadenia vyžaduje adresu V/V zariadenia
- Rozhranie V/V podsystemu



- Komponenty V/V podsystemu
 - Dekodér adries - Zisťuje adresu zariadenia
 - V/V registre - Obsahujú dáta určené pre V/V zariadenia
 - Multiplexor ReadData - Zabezpečuje smerovanie medzi V/V zariadením