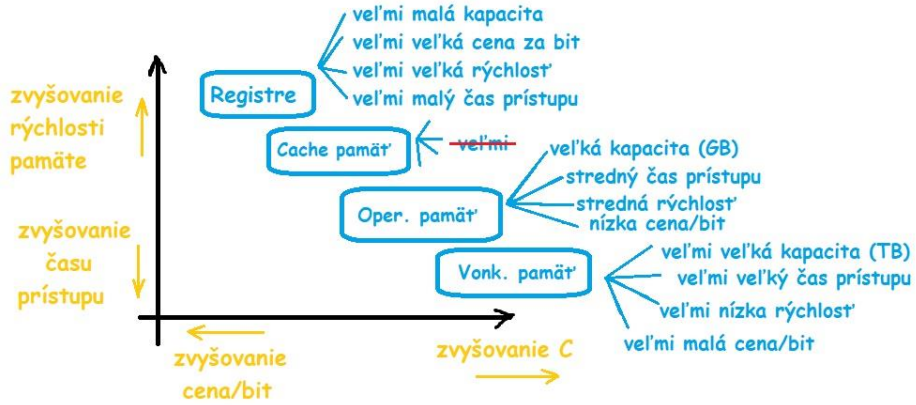
15. Vnútorné pamäte

Popíšte hierarchiu pamätí v PC:



Rozdeľte vnútorné pamäte podľa spôsobu výberu z pamäťového priestoru

Pamäte sa používajú na **ukladanie informácií** (údaje, programy, adresy, textové súbory)

Informácie vieme **rozdeliť** na – bity, bajty, slová, bloky, segmenty, stránky, štruktúry a ukladajú

Pamäťové zariadenia

Prístup k bunkám riadený pomocou adries | **Prístup k bunkám riadený obsahom pamäte**

Prístup k bunkám riadený pomocou adries: → RAM

a.) s NÁHODNÝM prístupom – doba prístupu je rovnaká pre všetky bunky v PAMATI,
nezáleží na poradí bunky v pamäti,
polovodičové pamäte RAM a ROM

b.) so SEKVENČNÝM – informácie sa ukladajú napr. na magnetickú pásku, alebo povrch optického disku (špirálová dráha),
doba prístupu nie je rovnaká pre všetky bunky v PAMATI,
ZÁLEŽÍ NA PORADÍ

c.) s CYKLICKÝM – informácie sú na nosiči, ten predstavuje SLUČKU/MNOŽINU SLUČIEK
umožňuje aby na nejakom mieste nosiča došlo k náhlej ZMENE ADRESY z najväčšej na najmenšiu → HARDDISK

Prístup k bunkám riadený obsahom pamäte:

Vyberáme pomocou tzv. **výberového kľúča**. Tie sa používajú vo vyrovnávacích pamätiach → **CACHE pamäť**

Popíšte pamäte ROM, ich štruktúru a delenie

Po vypnutí zdroja uchovávajú svoj OBSAH !!! Sú energeticky nezávislé; KLO

Ich štruktúra: DEKÓDER (AND hradlá) + programovateľné OR hradlá 2^n*y, n – počet vstupov

y - počet výstupov

Klasifikácia ROM = Pamäte programovateľné –

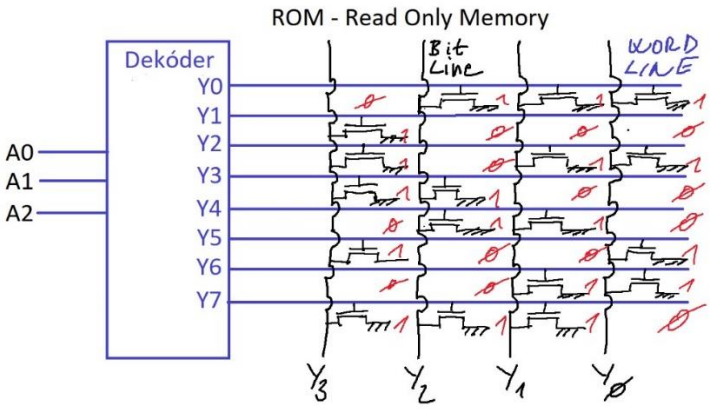
VÝROBCOM – obsah pamäte je zavedený počas výroby IO za pomoci technologickej masky a tá určuje ktoré kontakty sa PREPOJA a ktoré budú NEPREPOJENÉ → V TOVÁRNI Maskovanie je finančne náročné

U ZÁKAZNÍKA – vyrábané ako IO, je možné ich programovať špec. Zar. pripojenými k PC

PROM – 1 krát naprogramovateľná pamäť, dáta nemôžu byť vymazané

EPROM – informácia je uložená v podobe el. náboja, novú INFO vieme do pamäti uložiť pomocou programátora

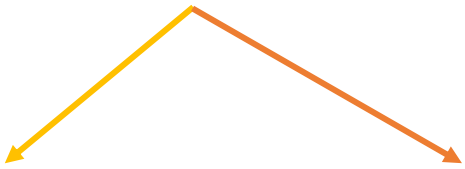
EEPROM – dáta sa mažú el. prúdom bit po bite, prepis možný len niekoľkokrát, FLASH pamäť



Popíšte pamäte RWM, ich delenie, fyzickú realizáciu (čím je tvorený 1 bit):

Do tejto pamäte vie CPU zapísať a vie z nej aj čítať

Po vypnutí zdroja strácajú svoj obsah !!! , Je energeticky závislá



STATICKÉ SRAM

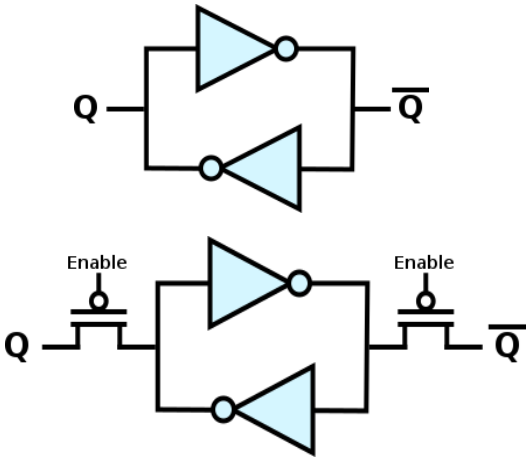
Základný pamäťový element tvorí RS preklápací obvod, po Zápise informácie zostane v stabilnom stave, ten sa zmení po zápise novej hodnoty; Vstup jedného invertora je výstupom Druhého invertora.

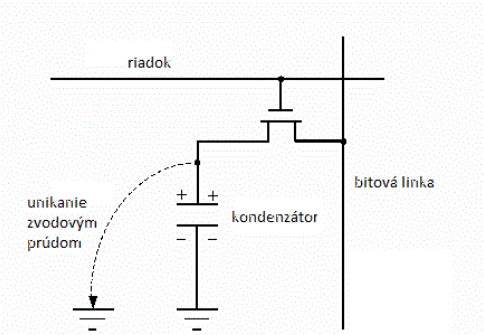
1 BIT REALIZUJEME VIACERÝMI TRANZISTORMI

DYNAMICKÉ DRAM

Základný pamäťový element je realizovaný pomocou parazitnej C. Tá sa pri zápise nabije. Vplyvom zvodového prúdu sa vybíja (STRATY) STRATY INFORMÁCIÍ → Preto je nutné periodicky obnovovať náboje C (REFRESH)

1 BIT REALIZUJEME JEDNÝM TRANZISTOROM





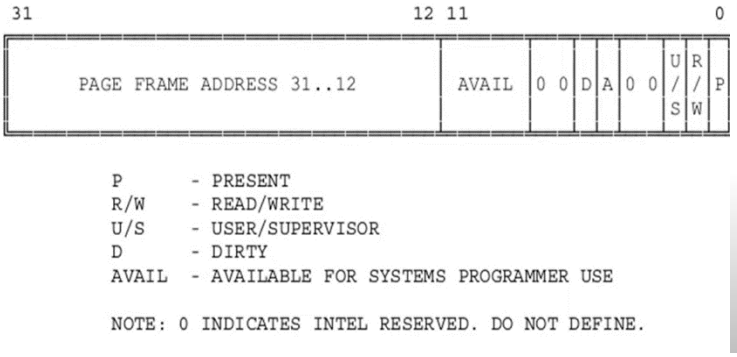
Načrtnite princíp virtuálnej pamäte:

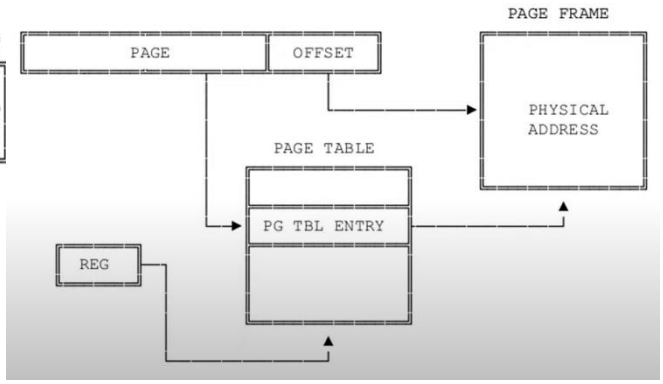
→ správa pamäte v multitaskingových operačných systémoch
→ preklad lineárnej adresy pamäte na fyzickú pamäť
→ rieši problém obmedzenej veľkosti operačnej pamäte
→ umožňuje vykonávanie procesov, ktoré sa nenachádzajú v operačnej pamäti celé
→ rôzne druhy nesusediacej pamäte sú prezentované aplikácii ako súvislá pamäť
  - virtuálny adresný priestor

→ Realizuje sa 2 spôsobmi
  - **STRÁNKOVANÍM** → rozdeľuje pamäť na stránky, ktoré majú hardvérom pevne danú (rovnakú) veľkosť – najčastejšie 4 kB
    - procesy sú umiestnené na disku
    - každy proces ma svoj vlastny adresný priestor
    - proces môže čítať a zapisovať iba v rámci svojho adresného priestoru
    - vytvárame zdanie vlastnej pamäte a tá pamať môže byť väčšia ako RAM
    - poskytuje nepriamu adresaciu
    - máme jednu vrstvu medzi procesom a fyzickou pamäťou to je virtuálna pamäť, ktorú spravuje jednotka správy pamäte – Memory Management Unit
    - program pracuje iba s virtuálnymi adresami
    - jadro OS ma za úlohu riadiť mapovanie kazdej virtuálnej adresy na fyzickú adresu
    - MMU → Memory Management Unit
      - používa na preklad tabuľku, v ktorej
      - virtuálna adresa je indexom do tabuľky
      - fyzická adresa je hodnotou na indexe
      - tabuľka sa volá tabuľka stránok
      - jeden proces ma jednu tabuľku, jeden adresný priestor
      - MMU dokáže obmedziť, ktoré virtuálne adresy sú prístupne v užívateľskom režime procesora
      - procesor obsahuje špeciálny register, v ktorom sa nachádza fyzická adresa RAM-ky, kde sa nachádza tabuľka stránok
      - meniť hodnotu tohto registra nie je možne v užívateľskom režime, iba v privilegovanom režime
    - Umožňuje:
      - rozlíšenie oprávnenia typu prístupu → čítanie/zápis
      - určí, či mapovanie existuje
      - rozlíšenie oprávnenia prístupu→ privilegovaný režim/ užívateľský režim
      - zistiť, či prístup k určitej adrese vyvolal zmenu hodnôt v pamäti
      - zistiť, či bolo vôbec niekedy pristúpene k nejakej adrese
    - **Rámec** (page frame) vs **Stránka** (page)

- Virtuálna adresa
  - formát virtuálnej adresy
    - bity 31 - 12 → číslo stránky
    - bity 11 - 0 → offset → posunutie vzhľadom na začiatok stránky

- Tabuľka stránok
- Položka tabuľky stránok
- preklad virtuálnej adresy na fyzickú adresu





Tabuľka stránok obsahuje položky

Preklad virtuálnej adresy na fyzickú adresu

- **SEGMENTOVANÍM** → rozdeľuje pamäť na segmenty, ktoré majú rôznu veľkosť

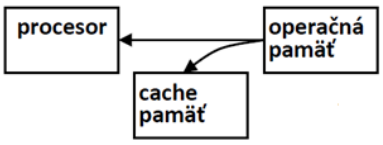
Načrtnite princíp Cache pamäte.

Cache

- Procesor sa obracia v 80 -95% prípadov na rovnakú oblasť v pamäti
- Zapíšeme ju do Cache pamäti ušetríme cca 90% prístupov k OP (operačnej pamäti)
- Cache pamäť kladie vysoké nároky na pamäťové bunky:
  - rýchlosť (používajú sa statické pamäte)
  - schopnosť komparovať
- vysoké nároky na riadiaci systém pamäti cache

PRINCÍP ČINNOSTI

- 1. ak sa procesor obracia na údaje, ktoré nie sú v pamäti cache, načítajú sa údaje z hlavnej pamäti nielen do procesora, ale aj do pamäti cache

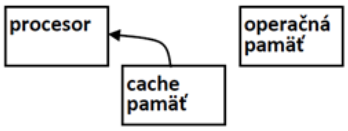


- 2. keď procesor nezaťažuje zbernicu, nahrá sa z hlavnej pamäte do cache pamäti celá oblasť údajov – stránky
  - ak sú potrebné údaje v cache pamäti, tak sa prečítajú údaje len z pamäti cache
  - ak sa procesor bude obracať na údaje, ktoré nie sú v cache pamäti, tak sa zopakuje krok číslo 1

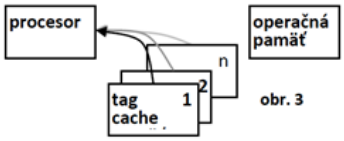
- v prípade, že jedna oblasť údajov v pamäti cache je málo, musíme mať v pamäti cache zapísaných viacej rôznych oblastí
- jednotlivé oblasti sú označované Tagom

Podľa počtu oblastí s údajmi delíme cache pamäte na:

- Jednocestné pamäte cache



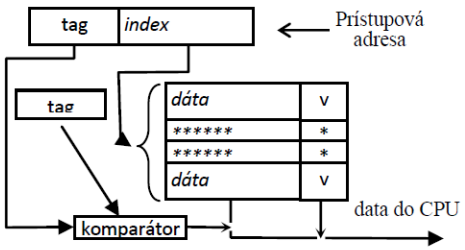
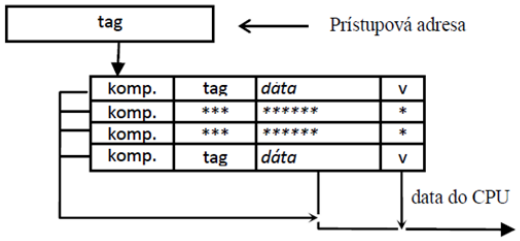
Viaccestné pamäte cache



výmena údajov medzi hlavnou (operačnou) a vyrovnávacou (cache) pamäťou
ak sú zaplnené všetky stránky cache pamäti, treba sa rozhodnúť, ktorá stránka v Cache pamäti bude prepísaná
LRU (Lest Recently Used) – najdlhšie nepoužívaný →každá oblasť údajov má čítač, ktorý sa vynuluje, ak je oblasť údajov čítaná, alebo inkrementuje, ak oblasť údajov nie je čítaná →v prípade potreby sa prepíše oblasť údajov, ktorá má v čítači najvyššiu hodnotu.
FIFO - najdlhšie zapísaný →inkrementovaním čítača je vyhodnotená stránka, ktorá je v pamäti cache najdlhšie

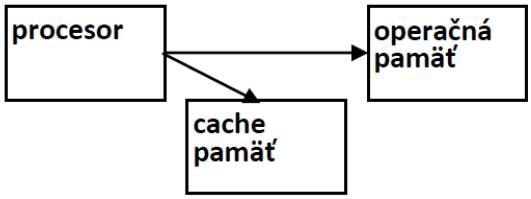
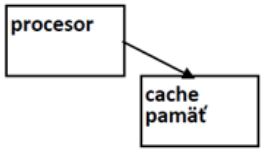
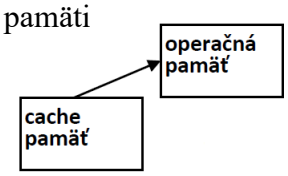
ČÍTANIE ÚDAJOV Z PAMATE CACHE

- Keď procesor číta z pamäti, treba určiť, či údaj je v cache pamäti, alebo nie
- Spôsob, akým to bude realizované, sa určuje podľa organizácie pamäti cache
  - Priamo mapovaná pamäť cache
  - Plne asociovaná pamäť cache

Priamo mapovaná pamäť cache	Plne asociovaná pamäť cache
do pamäti cache sa zapisuje s údajmi aj TAG	Celá adresa je určená TAG -om
Tag je časťou adresy	
radič pamäti cache porovnáva či požadovaný TAG je v pamäti cache →ak je, tak dáta na výstup z cache pamäti sa určia na základe INDEXU, →dáta sú ešte potvrdzované tzv. príznakom validity	V pamäti cache sú zapísané okrem údajov aj adresy na bunkách, ktoré sú *schopné komparácie*
	

ZÁPIS ÚDAJOV DO CACHE A OP

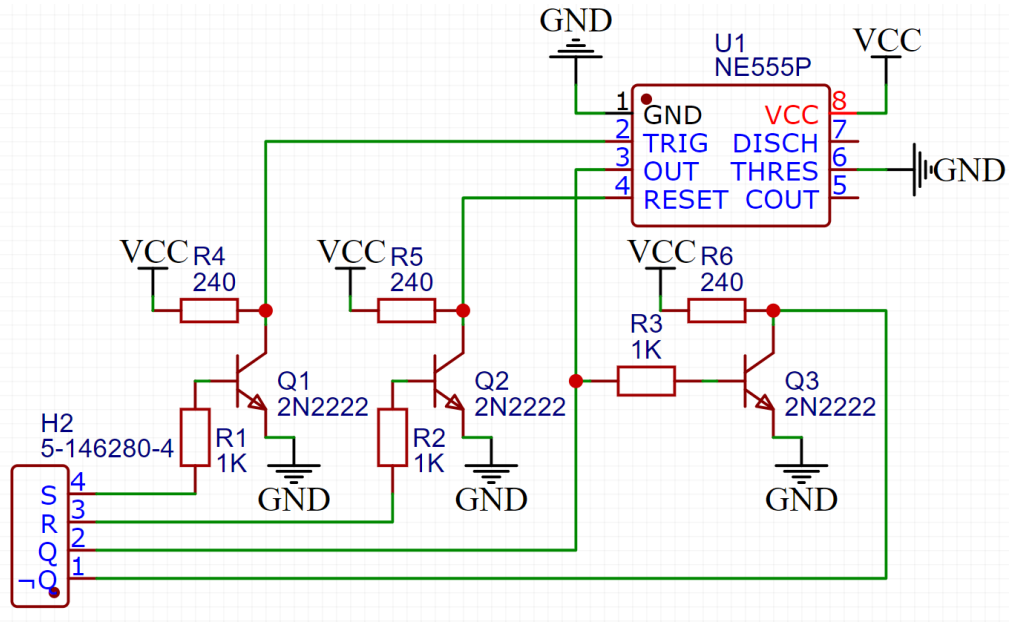
- Podľa spôsobu zápisu delíme pamäte cache
  - Zápis ponad cache alebo súčasný zápis
  - Oneskorený zápis

Zápis ponad cache alebo súčasný zápis	Oneskorený zápis
	Zápis na 2 razy: →1.najskôr len do pamäti cache 
→ činnosť systému sa spomaľuje, pretože sa -údaje zapisujú do vyrovnávacej pamäte -súčasne sa zapisujú aj do pomalej operačnej pamäti	→ 2. pri výmene stránky z pamäti cache do operačnej pamäti 

Vysvetlite činnosť Bistabilného preklápacieho obvodu s vyžitím časovacieho obvodu 555:

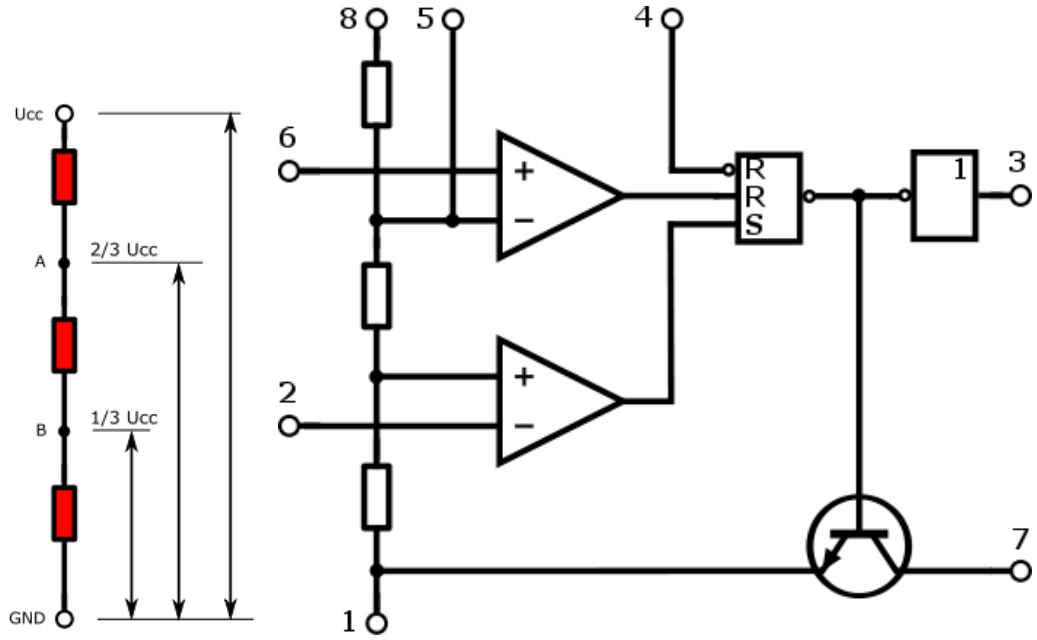
Realizácia BKO pomocou NE555:

Zapojenie aj s komplementárnymi komponentami na dosiahnutie plnej funkčnosti bloku obvodu **RS**$_S$



Obvod NE555P pri tomto zapojení ako RSS obvod si vyžaduje na vstupe negovaný Set a negovaný Reset. Ďalej má len jeden nenegovaný výstup. Tým pádom potrebujeme 3 hradlá NOT na to aby plne spĺňal funkciu obvodu RSS. Za to sú na schéme 3 tranzistory (v zapojení NOT). Samotný IO NE555P má vnútorný

negovaný RS preklápací obvod, ktorý má 3 vstupy (Set a negovaný a nenegovaný Reset). Negovaný reset sme zapojili na výstup Reset NOTu a nenegovaný tým pádom znefunkčníme (zemníme pin 6 a pin 5 je NC). Na to, aby sa na vstupe S vnútorného RS obvodu objavila logická jednotka potrebujeme na pin 2 dodať napätie nižšie ako 1/3 napájacieho – je to kvôli zapojeniu komparátora – preto na pin 2 pripájame negovaný set. Na pine 3 máme nenegovaný výstup, ktorý ešte negujeme NOTom a dostávame Q negované. Pin 7 je NC, pin 8 je napájanie a pin 1 je zem.



Uveďte použitie pamätí v digitálnych osciloskopoch:

Pamäte nájdeme najmä v DO!

Ukladanie firmvéru DO na EPROM

Ukladanie nameraného priebehu do dlhodobej pamäti

Akvizičná pamäť *(typu FIFO)* – pamäť v ktorej sa ukladajú zdigitalizované vzorky aj spolu s časom odobratia vzorky v reálnom čase, na ňu máme najvyššie požiadavky s hľadiska rýchlosti veľkosti či odozvy – od jej kvality veľmi závisí celková kvalita osciloskopu a výsledného merania (závisí od nej vzorkovacia frekvencia, rozlíšenie vzoriek kvalita ZOOMu, čas zrekonštruovania meraného signálu a podobne)