

### 3. Úvod

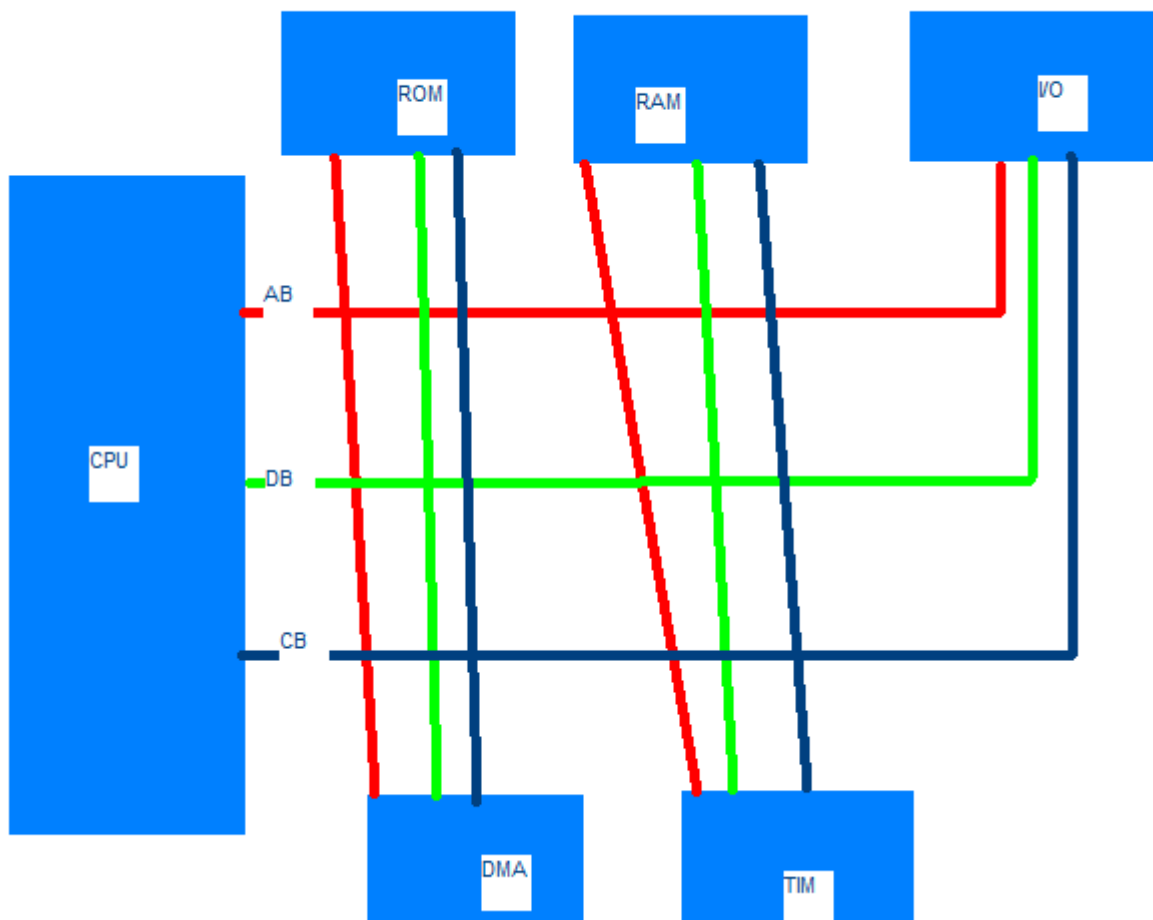
1951....zymax (1. sériově vyráběné PC)

1971....mikroprocesor 4004 (4-bit)

INTEL: 4004, 8008, 8080A (IBM PC-XT), 80286, 80386, 80486, PENTIUM

MOTOROLA: 6800, 68000, 68020, 68040, 68060

## Mikro počítače a mikroprocesor



Základní struktura mikro počítače

Obvody mikropočítače můžeme rozložit do pěti částí, jak je zobrazeno na obrázku. Data jsou mikropočítačem zpracovávána po slovech. V daném časovém okamžiku (takt procesoru) mikropočítač pracuje s jedním slovem. Typická délka slova je 8, 16, 32 nebo 64 bitů. Nejrozšířenější jsou mikropočítače s délkou slova 8 a 16 bitů. Osmibitové slovo nazýváme bajt (byte).

### **Generátor taktů (TIM)**

Generuje hodinový (taktovací, synchronizační) signál, který synchronizuje činnost samotného procesoru a také jeho spolupráci s ostatními částmi mikropočítače. U současných typů mikroprocesoru bývá již tento generátor jejich součástí.

### **Mikroprocesor (CPU)**

Je základním prvkem mikropočítače. Řídí jeho celou činnost. Zajišťuje provádění instrukcí uložených v paměti, řídí toky dat ze vstupních částí mikropočítače, tyto data zpracovává a následně řídí tok dat směrem k výstupním portům.

### **Paměť ROM**

Obsahuje ve většině případů instrukce, které zajišťují realizaci daného algoritmu řízení pro přizpůsobení mikropočítače určité aplikaci. Dále paměť může obsahovat konstanty a neměnné tabulky používané v programu. Z této paměti lze pouze číst, programuje se při výrobě. Určitými variantami paměti ROM jsou paměti PROM, EPROM.

### **Paměť RWM**

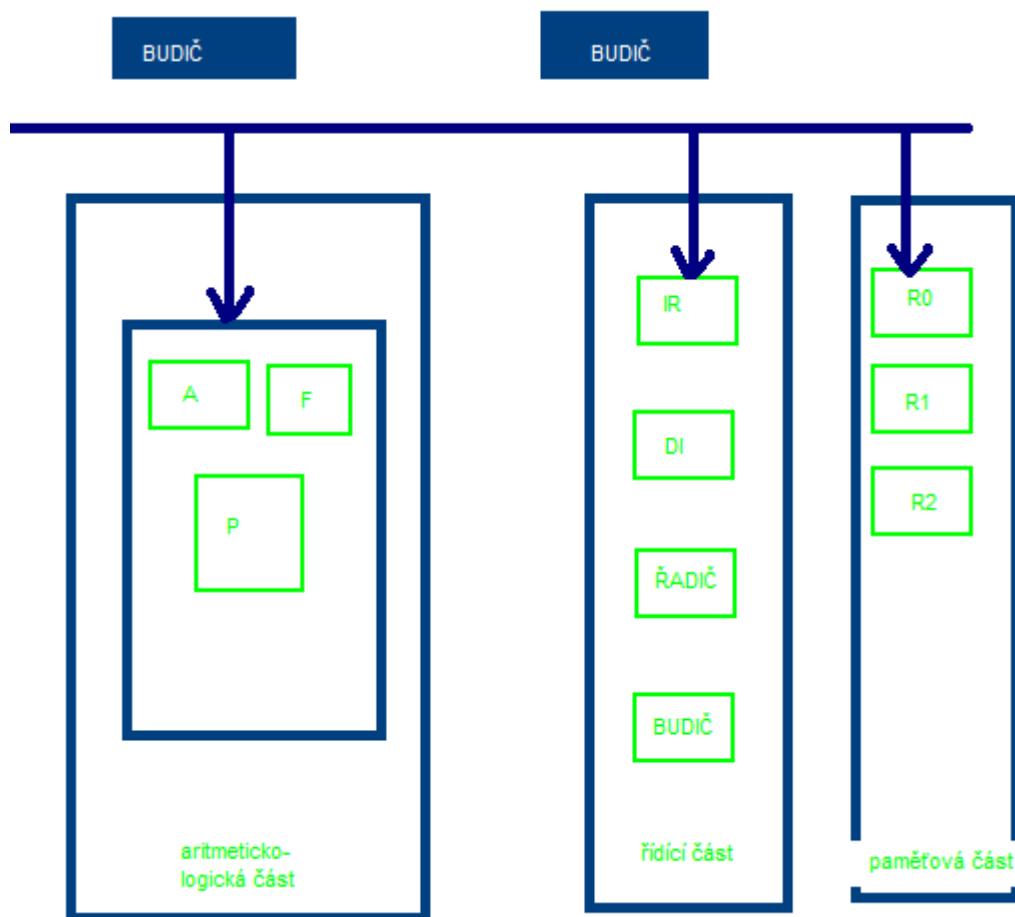
Označována také jako RAM, zjišťuje dočasné uložení dat zpracovávaných mikroprocesorem. Data do paměti může mikroprocesor uložit a opět zpětně vyzvednout. Do této paměti lze i zapisovat.

### **Vstupní a výstupní porty**

Umožňují spojení mikropočítače s okolním prostředím (klávesnice, display, výkonové akční členy atd.). Tyto však někdy nemusí být součástí každého mikropočítače.

## **Struktura mikroprocesoru**

Termín mikroprocesor označuje ve většině případů integrovaný obvod vyrobený technologií vysoké hustoty integrace. U starších typů mikroprocesoru nebyly vždy součástí mikroprocesoru všechny jeho nezbytné části jako jsou např. generátor hodinového signálu, obvody pro řízení sběrnice a jiné. Výrobci k těmto mikroprocesorům dodávali specializované obvody, které doplňovaly mikroprocesor o tyto části. Tyto obvody pak tvořily spolu s mikroprocesorem tzv. skupinu mikroprocesoru. Současné typy vyráběných mikroprocesorů jsou již těmito potřebnými částmi vybaveny přímo na čipu.



## Řadič

Řídí a koordinuje činnost všech ostatních částí mikroprocesoru. Podílí se i na vytváření vnějších signálů sběrnice celého mikropočítače. Jeho hlavní činností je načítat instrukci po instrukci z paměťových míst a po dekódování ji zpracovat.

Adresa, odkud má danou instrukci načíst, se při každém instrukčním cyklu nachází v programovém čítači (Program Counter [PC], Instruction Pointer [IP]), jehož obsah se po provedení instrukci nejčastěji modifikuje prostou inkrementací. To je v případě, že jsou instrukce v paměti uloženy postupně za sebou. Někdy je obsah programového čítače modifikován naplněním jinou hodnotou, než která by vznikla inkrementací. Je to v případech, kdy je třeba provést jinou instrukci, která přímo nenásleduje za právě provedenou. Takový případ nastává po instrukcích skoku (větvení), volání podprogramu, přerušování, atd.

Adresu pak musí řadič zpřístupnit paměťovému místu, kde je uložen program – tedy instrukce – nebo data. V některých případech instrukce zabírá více než jedno paměťové místo. Pak jsou její jednotlivé části uloženy v paměti postupně za sebou. Současně jsou řadičem generovány v potřebné časové posloupnosti i signály potřebné k řízení paměti. Ta po té vystaví potřebnou instrukci nebo její část na datovou sběrnici, odkud si jí řadič vyzvedne do registru instrukcí (IR). V tomto registru zůstává instrukce po celou dobu jejího výkonu.

## Soubor registrů

Tvoří tzv. paměťovou část mikroprocesoru. Jsou to v podstatě samostatné izolované paměťové buňky, které jsou vytvořeny přímo ve struktuře mikroprocesoru a jsou adresované přímým zapsáním dat. Tento způsob adresace zajišťuje velkou rychlost přístupu dat bez použití adresových a řídicích sběrnic. Tuto paměťovou část mikroprocesor využívá k dočasnému uchování informací – dat v průběhu vykonávání operací nebo je využívána programem, tedy vlastně uživatelem. Různé typy mikroprocesorů se liší také počtem registrů. Někdy bývá uspořádání takové, že existuje jakási základní sada registrů, tzv. banka a další registry jsou tvořeny stejnými názvy přepnutím další banky. Jinými slovy je v každém okamžiku (strojovém cyklu) přístupná pouze jedna sada registrů. Při požadavku přístupu do jiné banky je nutno danou banku nastavit jako aktuální. Toto je jeden z mnoha možných způsobů uspořádání paměťové části mikroprocesoru nikoliv mikropočítače.

## Aritmeticko-logická část

Je další důležitou částí mikroprocesorů. Tvoří jí aritmeticko-logická jednotka a podílí se na vykonávání všech aritmetických a logických operací mikroprocesoru. U jednoduchých procesorů umožňuje sčítat a odečítat, u výkonějších typů pak může provádět násobení, dělení a posuvy slov ve dvojnásobné přesnosti. Oproti kalkulačkám mají tedy mnohem menší matematickou vybavenost a složitějších nebo přesnějších matematických operací v krátkých časech, je vhodné doplnit dalšími technickými prostředky (např. koprocesory). Některé složitější mikroprocesory mohou již tyto technické prostředky obsahovat ve své struktuře. Představiteli těchto mikroprocesorů jsou např. signálové procesory. Obsahují např. na čipu implementovanou hardwarovou násobičku, která umožňuje vynásobit dvě čísla v jednom strojovém cyklu.

Některé aritmeticko-logické jednotky ke své činnosti vyžadují tzv. střadač (akumulátor), tj. registr, ve kterém je ve většině případů uložen jeden z operandů vstupujících do matematické operace. Další operand je poté k dispozici buď přímo z operačního kódu instrukce anebo se může nacházet v nějakém registru či paměti mikropočítače. Je-li přítomen střadač, pak u takových mikroprocesorů se vždy účastní matematických operací. Výsledek je poté uložen zpět do střadače. Tento způsob užívání střadače může některým uživatelům poněkud komplikovat práci nezbytnými přesuny při přísunu dat, která se stávají operandy jednotlivých operací a při úklidu výsledků, který je nutný, abychom si je nepřepsali.

Některé mikroprocesory však mohou provádět matematické operace bez účasti střadače (u takových pak střadač vůbec neexistuje) a to tak, že pro tyto operace využívá buď soubor registrů na čipu nebo přímo paměťové pozice. Takové mikroprocesory se vyznačují většími možnostmi z hlediska uživatele pro vývoj programu. Menší nevýhodou při využívání paměti pro matematické operace je doba přístupu do paměti. Ta může mít za následek zmenšení výpočetní rychlosti. Pak je v tomto případě vhodné využívat paměti (např. umístěných přímo na čipu, pokud jsou ve struktuře mikroprocesoru obsaženy), které nezatěžují mikroprocesor svou přístupovou dobou.

Součástí aritmeticko-logické části je tzv. příznakový registr (Flags). Jednotlivé jeho bity nám dokumentují prováděné matematické operace, informují nás o významných vlastnostech výsledků.

Na základě těchto bitů můžeme provádět modifikaci daného algoritmu např. prostřednictvím podmínek v programu.

Nyní si ještě popíšeme základní parametry, podle kterých je možné mikroprocesory dále dělit:

- **Šířka vnitřní sběrnice** – jinými slovy *kolikabitový je*. V některých případech platí, že šířka vnitřní sběrnice mikroprocesoru (jeho vnitřní sběrnice dat) je shodná se šířkou datové sběrnice mikropočítačové struktury s tímto procesorem. V jiných je typické, že mikroprocesor obsahuje bloky, které jsou kvůli zvýšení výkonnosti uvnitř vícebitové. Šířka vnější sběrnice je v takovém případě redukována. Obvykle se tedy šířka vnitřní sběrnice mikroprocesorů posuzuje podle šířky slova registrů eventuálně střadače.
- **Rychlost mikroprocesoru** – je posuzována podle frekvence krystalu (oscilátoru), lépe však podle výkonu mikroprocesoru, daným počtem vykonaných instrukcí za sekundu. Tento údaj se udává v tzv. MIPSech, tj. *million of instructions per second*. Vzhledem k tomu, že instrukce trvají různý počet strojových cyklů, je tento údaj určitým průměrem. U mikroprocesorů dnešní doby trvá provedení většiny instrukcí pouze jeden strojový takt.

## Základní činnost mikroprocesoru

### Časování mikroprocesoru

U mikroprocesorů je časování odvozeno z hodinových signálů, které mikroprocesor synchronizují. Ve většině případů je to signál vzniklý dělením frekvence krystalového oscilátoru. Základní časovou jednotkou je tedy perioda tohoto signálu zvaná takt.

Každá instrukce se potom vykonává určitý počet taktů (dob), celý tento čas nazýváme instrukční cyklus. Délka instrukčního cyklu se může lišit podle složitosti instrukce. Některé instrukce, ve většině případech instrukce s podmínkou, mají proměnnou délku instrukčního cyklu. Podle délky instrukčních cyklů jednotlivých instrukcí jsme však schopni určit délku trvání jednotlivého programu. Toto např. vyžadujeme u výpočtu programového zpoždění. S růstem výkonnosti mikroprocesorů se však výrobci snaží o efektivnější činnost mikroprocesoru, což vede k tomu, že některé bloky pracují vlastně paralelně a jednotlivé instrukční cykly se překrývají – tzv. pipelining. Pak je výpočet doby programu velmi problematický, ne-li nemožný.

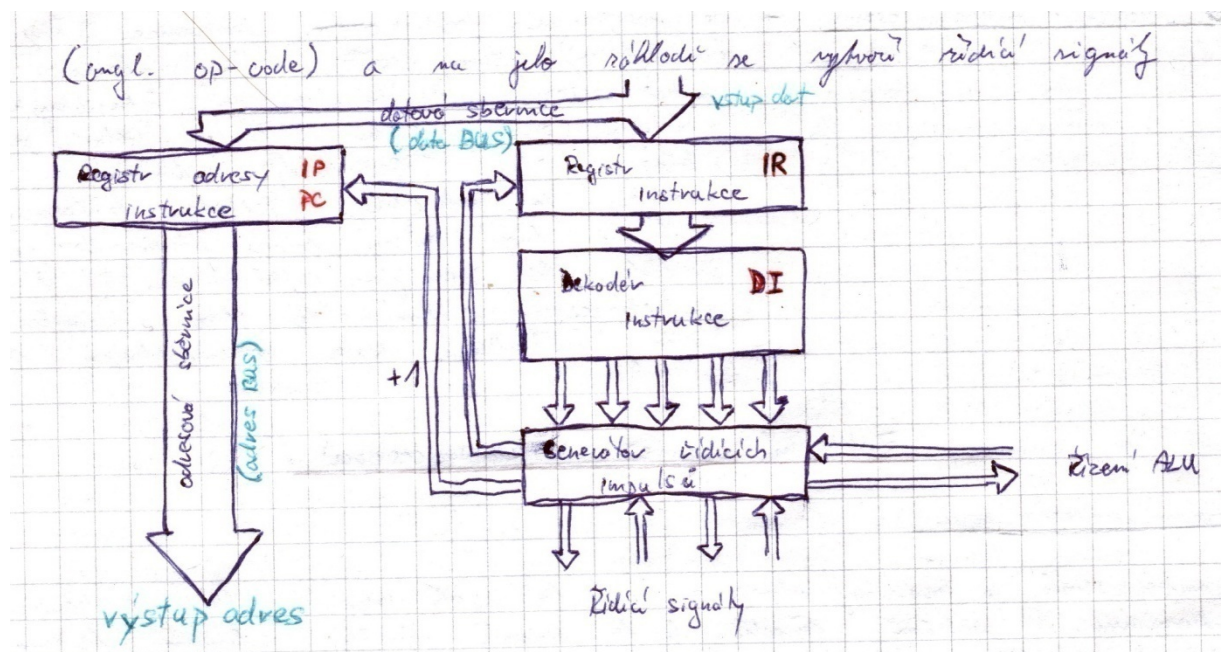
Mezi instrukčním cyklem a jednotlivými takty ještě obvykle rozeznáváme tzv. strojové cykly. Jsou to dílčí operace, ze kterých se instrukční cyklus sestává a obvykle souvisejí s činnostmi, které proběhnou mezi mikroprocesorem a jeho okolím. Prvním strojovým cyklem každého instrukčního cyklu je ve většině případech načtení operačního kódu instrukce do instrukčního registru IR. Dalším strojovým cyklem se instrukce dekoduje daný operační kód, v dalších strojových cyklech nastává čtení operandů neboli dat a vykonání instrukce.

Strojový cyklus se pak skládá z jednotlivých taktů a je jednoznačně stanoveno, jakou činnost mikroprocesor v každém taktu vykonává, kdy generuje které řídicí signály testuje. V manuálech

jednotlivých mikroprocesorů jsou časové posloupnosti uvedeny v časových diagramech zápisu a čtení z a do paměti, periférií apod.

## Řadič

- řídicí jednotka („control unit“)
- koordinuje všechny funkce mikropočítače
- má tyto tři základní úlohy:
  1. řídí pořadí v němž jsou prováděny instrukce programu
  2. dekóduje instrukce
  3. vysílá do ostatních částí mikropočítače všechny řídicí impulsy potřebné pro provádění instrukcí
- hlavním úkolem řadiče je prostřednictvím čítače vyzvednout instrukce z hlavní paměti a uložit jí do registru instrukce. Tam se dekóduje operační znak



Při zapnutí počítače se registr adresy instrukce vynuluje (program začíná na nule), tím se paměť počítače a na výstupu se objeví číslo, které odpovídá kódu první instrukce. Tento kód se převede do registru instrukce, aby však připojený dekodér instrukce dostatek času z jeho dekódování. Z dekodéru je daná informace do generátoru řídicích impulsů. Generátor řídicích impulsů vysílá impulsy do všech ostatních částí mikropočítače, především do ALU. Je-li instrukce provedena, vydá generátor povel do registru adresy instrukce, na adresovém vstupu se objeví následující instrukce. Celý cyklus tak může probíhat znovu

Ze schématu je zřejmé, že spoje vedou k registru instrukce ale i k registru adresy instrukce. Tím je umožněno, aby se speciálními instrukcemi měnil obsah tohoto registru (RAI)

## ALU

- je výkonná část mikroprocesoru v níž se vykonávají jednotlivé aritmetické a logické operace. Je ovládána řídicími signály z řadiče na základě instrukcí, které určují druh prováděné operace.

## Zpracování instrukce

Každý mikroprocesor je od výrobce vybaven souborem instrukcí. Instrukce jsou nejmenší jednotky, na kterých je složen program.

### Základní části instrukce

- a) operační kód (OP)
- b) operační kód (JUMP) = adresa
- c) operační kód (MOVE) = data

## Operační kód

První částí instrukce je operační kód. Ten přesně definuje činnost při provádění instrukce. Některé instrukce obsahují jen operační kód. V těchto případech jsou prováděny jen vnitřní operace procesoru. Ve druhém případě obsahuje ještě instrukce adresu operandu. Operandy jsou uloženy v datové paměti. Podle činnosti instrukce je můžeme dělit na několik skupin:

- 1) **Instrukce přesunů** – tato skupina instrukcí je vlastně základní. Je nutné mít možnost přesouvat data mezi jednotlivými částmi mikroprocesoru a mikropočítače. Tyto přesuny přesouvaná data neovlivňují. K přesunům dochází mezi jednotlivými registry mikroprocesoru, mezi paměťovými buňkami a mezi sebou navzájem. Mikroprocesor obvykle dovoluje přesuny tolikabitových dat, kolikabitový sám je. Nabízí i několik instrukcí o dvojnásobné délce, tyto instrukce pak vlastní přesun provádějí nadvakrát.
- 2) **Instrukce aritmetické** – mezi tyto patří instrukce prováděné v ALU. Mikroprocesory jsou vybaveny pouze jednoduššími operacemi (sčítání, odčítání, násobení, dělení). Zpracování složitějších operací je prováděno numerickými algoritmy nebo je přenecháno matematickým co-procesorům. Podle těchto instrukcí lze snadno zjistit, zda se na matematických operacích podílí střadač.
- 3) **Instrukce logické** – tyto instrukce řeší logické operace s daty (logický součet, logický součin).
- 4) **Instrukce posunu a rotace** – tyto instrukce provádějí posun jednotlivých bitů binárního čísla o jeden doleva nebo doprava. U těchto instrukcí se velmi často s výhodou využívá skutečnost, že posuv doprava s doplněním nejvyšší pozice nulou představuje dělení dvěma a posuv doleva s doplněním nejnižší pozice nulou představuje násobení dvěma. U rotace probíhá taktéž posun s tím, že bit, který opouští nejvyšší pozici je zařazen do pozice nejnižší a naopak.

- 5) **Instrukce skoku** – tato skupina instrukcí ovlivňuje programový čítač = registr adresy instrukce. To znamená, že program nepokračuje následující instrukci, ale instrukcí, která se nachází na adrese, která je uvedena jako parametr instrukce skoku. Rozeznáváme několik druhů instrukcí skoku:
- a. **nepodmíněný skok** – nastavuje vždy programový čítač na hodnotu danou parametrem instrukce
  - b. **podmíněný skok** – vykoná skok pouze v případě, je-li splněna podmínka (podmínka = parametr instrukce). Není-li splněna podmínka, program pokračuje následující instrukce
  - c. **volání podprogramu** – funkce této instrukce je podobná nepodmíněnému skoku s tím rozdílem, že před modifikací programového čítače novou hodnotou se původní hodnota uloží do zásobníkové paměti. Tím se pak umožní návrat na instrukci následující za instrukcí volání podprogramu.

## Paměti mikropočítače

**Paměť programu** – je nejčastěji umístěna v pevné paměti (BIOS, ROM, PROM, EPROM, EEPROM). Jsou v ní uloženy instrukce, které prostřednictvím řadiče řídí činnost MP.

**Paměť dat** – je tvořena pamětí RWM, nejčastěji statickou, méně často dynamickou. Do této paměti se dále umísťuje i zásobníková paměť.

**Zásobníková paměť** – tato paměť je tvořena posloupností paměťových míst v paměti RWM. Z této paměti je přístupné vždy jen jedno paměťové místo – VRCHOL ZÁSOBNÍKU. Na toto místo ukazují tzv. ukazatel zásobníku. Dnem zásobníku se označuje místo, do kterého se uskutečnil první zápis. Hloubka zásobníku je počet paměťových míst v zásobníku. Tato paměť odpovídá typu LIFO. Ukazatel zásobníku je uložen v registru, do něhož se při zápisu dat přičte jednička a při čtení se jednička odečítá. Výhodou této paměti je to, že se do ní při čtení a zápisu nemusí vysílat adresa, a tím je ušetřen čas při zpracování.

**Rychlá vyrovnávací paměť (CACHE)** – používá se ke zrychlení toku dat mezi RWM a CPU. Při programování není nutné s existencí cache počítat. Programátorovi se jeví, jako by v systému vůbec nebyla. Přesun dat mezi cache a ALU řídí speciální řadič. V dnešních počítačích se běžně používají dva druhy cache: L1 – interní cache – je paměť, která slouží k vyrovnání rychlosti výkonných procesorů a pomalejších pamětí. Je integrována přímo na čipu procesoru a je realizována, pomocí paměti SRAM. L2 – externí cache – tato paměť je umístěna mezi pomalejší operační paměti a rychlým procesorem. Tato paměť je vyrobena jako SRAM. Tato paměť je osazena na základní desce PC (výjimku tvoří procesor Intel Pentium II – ten má paměť integrovanou v pouzdře).

Práce s pamětí cache vychází ze skutečnosti, že program má tendenci se při své práci určitou dobu zdržovat na určitém místě. Je-li požadována nějaká informace z paměti, je nejdřív hledána v cache (v interní a následně v externí). Pokud požadovaná informace není přítomna v cache paměti, je zavedena přímo z paměti operační. Kromě momentálně požadované informace se však do cache paměti zavede celý blok paměti, takže je velká pravděpodobnost, že následně požadované informace již budou v cache přítomny.



Virtuální paměť, nebo také adresování virtuální paměti, je správy operační paměti, umožňuje využívat vnitřní paměť, která je významně větší, než je skutečná fyzická velikost paměti. Děje se to tím, že se v danou chvíli nepotřebná paměť odkládá na pevný disk (odkládací prostor) a v případě potřeby se opět zavádí do vnitřní paměti. Zvýšené nároky na systémové a hardwarové zdroje jsou zanedbatelné s porovnáním s výhodami, které virtuální paměť přináší (menší spotřeba fyzické paměti, která je i přes neustále klesající ceny dražší, než pevný odkládací prostor). Systém virtuální paměti umožňuje efektivně využít menší operační paměť za cenu jen nepatrné ztráty výkonu.

Základní principy:

- všechny adresy, které proces používá jsou zpravovány pouze jako virtuální – transformaci na fyzické adresy provádí správa virtuální paměti.
- paměť je rozdělena na větší úseky, které se nazývají stránky. Správa virtuální paměti rozhoduje samostatně o tom, která paměťová stránka bude zavedena do vnitřní paměti a která bude odložena do odkládacího prostoru.
- používají se různé strategie pro řízení přesunu stránek tam a zpět

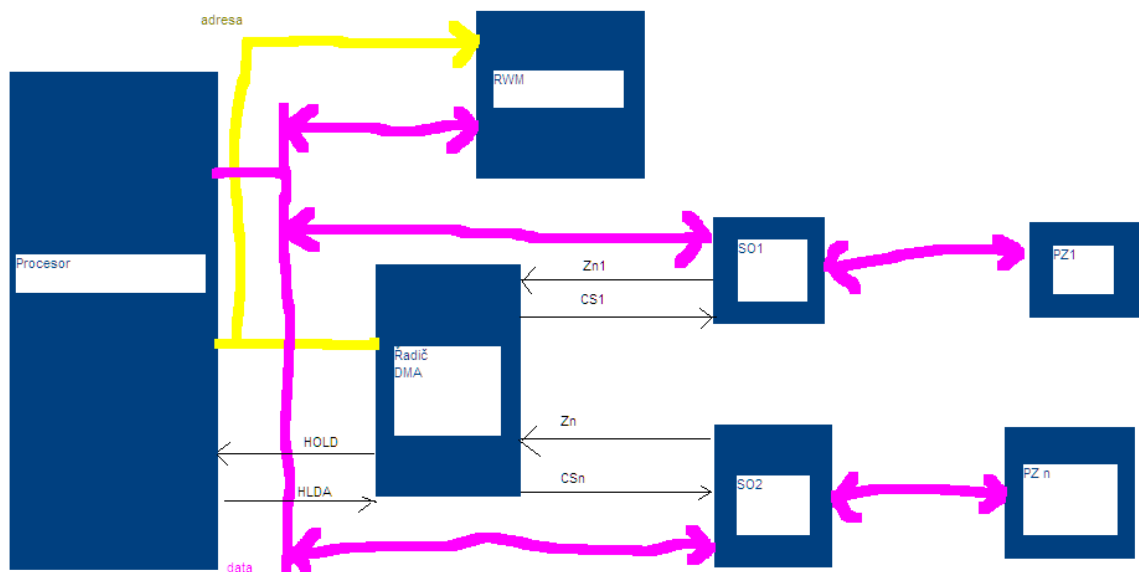
**Zápisníková paměť** – skupina rychlých registrů, jejichž adresování probíhá pomocí adresní části instrukce, slouží pro ukládání mezivýsledků operací (jsou součástí architektury procesoru)

## Přímý přístup do paměti

Při běžných operacích se data přesouvají mezi periferní jednotkou a operační pamětí prostřednictvím střadače. Tento způsob přenosu je poměrně **zdlouhavý a zbytečně zaměstnává procesor**. Proto se používá **PŘÍMÝ PŘENOS** mezi hlavní pamětí a periferními zařízeními přes pomocný integrovaný obvod, který se označuje **řadič DMA (Direct Memory Access)**

Zrychlení přenosu nastává ze dvou důvodů:

- Jsou potlačeny neproduktivní činnosti mikroprocesoru. A to hlavně z hlediska využití hlavní paměti a sběrnic pro přenos dat (čtení instrukcí pro řízení přenosu dat, **zbytečné přenášení dat** mezi hlavní pamětí a periferním zařízením **přes střadač procesoru**)
- Zkrátí se cyklus čtení a zápisu, protože rychlost řadiče DMA je větší než rychlost mikroprocesoru.



Řadič DMA po přijetí žádosti na přesun dat vyšle do mikroprocesoru signál HOLD. Tento signál zajistí přechod vnější datové i adresové sběrnice do stavu vysoké impedance, a tím umožní řadiči převzít řízení sběrnic. Řadič DMA generuje řídicí signály CS (chip select) pro výběr periferního zařízení (PZ) připojených přes stykové obvody SO. Dále generuje adresy hlavní paměti a některé řídicí signály (např. R/W).

Řadič DMA řídí samostatně přesun dat *po slovech* nebo *po blocích*. Obsahuje

- registr dat
- registr adresy
- čítač slov
- řadič přesunu dat

**Registr dat** - slouží k přechodnému záznamu slova během přesunu mezi periferní jednotkou a operační pamětí

**Registr adresy** – obsahuje adresu operační paměti, na kterou se má slovo uložit nebo ze kterého se má číst.

**Čítač slov** – obsahuje na počátku počet slov, která se mají přesunout během jednoho spojení. Nulový obsah čítače znamená, že byl přesun ukončen.

**Řadič přesunu dat** – komunikuje s procesorem a řídí samostatně přesun dat na základě řídicího slova procesoru. Řídicí slovo určuje způsob přesunu, počáteční adresu bloku slov v operační paměti a pořadí priorit přenosových kanálů.

Periferní zařízení jsou připojena k řadiči DMA připojena přes stykové obvody přenosovými kanály, kterým jsou přiřazeny určité priority.

Operace přímého přesunu dat probíhá v etapách:

- 1) přesune se řídicí slovo z procesoru do DMA řadiče
- 2) jsou-li data v periferní jednotce připravena k přenosu, vyšle DMA řadič signál HOLD do procesoru.
- 3) procesor dokončí strojový cyklus, vyhodnotí priority přenosových kanálů periferií, uvolní sběrnici a do vybrané jednotky vyšle signál HLDA o přidělení DMA
- 4) vybraná periferie vyšle na příslušnou adresu operační paměti jedno slovo a zmenší obsah čítače slov o jedničku. Tato operace se opakuje dokud obsah čítače není nula

Řízení přenosu DMA má několik variant:

- 1) klasický způsob řízení – činnost procesoru je zastavena po celou dobu přenosu DMA, po přijetí žádosti o přenos DMA mikroprocesor dokončí probíhající instrukci, odpojí se od sběrnic a vydá signál, který spustí řadič DMA. Nevýhoda – po celou dobu přenosu je procesor vyřazen z činnosti.
- 2) metoda sdílení sběrnic a paměti – dochází ke sdílení sběrnice a operační paměti mezi procesorem a DMA. Tato metoda nezpomaluje řešení úloh mikroprocesorem, protože přenos dat pomocí řadiče DMA probíhá v těch dobách instrukčního cyklu, ve kterých mikroprocesor nevyužívá paměť ani sběrnice.

## Přerušení

**Mechanismus přerušení** – kromě aktuálně běžícího programu musí MP reagovat na podněty jiných zařízení nebo program, např. reakce na stisk klávesy, informace o tisku, ... Toho se dá docílit dvěma způsoby:

- a. procesor v pravidelných časových intervalech prochází všechna zařízení a kontroluje, zda pro něj nemají nějaký signál, který by zpracoval. Tento mechanismus je nevhodný pro časovou náročnost a mnohdy zcela zbytečné operace
- b. je vytvořen systém přerušení, tzn. že každá operace, která potřebuje provést zastavení aktuálního procesu, má své číslo přerušení. V operační paměti je tabulka vektorů přerušení, který ukazuje na adresu v paměti, kde je uložen obslužný program požadovaného přerušení. Tento systém užívají v dnešní době všechny MP.

Procesor, který komunikuje s okolím třemi základními způsoby:

- a. pomocí sběrnic
- b. přerušením
- c. přes kanály přímého přístupu do paměti

Přerušení probíhá na následujícím principu:

- a. zařízení, které potřebuje obsluhu, vyvolá přerušení – HW nebo SW. Procesor dokončí probíhající instrukci a uloží si dosažené hodnoty a číslo následující instrukce do paměti
- b. vektor přerušení spustí na příslušné adrese v paměti program pro obsluhu přerušení, pro zpracování je určen speciální obvod – řadič přerušení (interrupt controller)
- c. po obsluze zařízení, které vyvolalo přerušení, se z paměti vezmou dočasné hodnoty a následující instrukce přerušeného programu a pokračuje se v jeho provádění.

Linky, kterými proudí signály přerušení, jsou součástí sběrnice, se značí IRQ (interrupt request). Jednu linku může používat jedna periferie

## Sběrnice

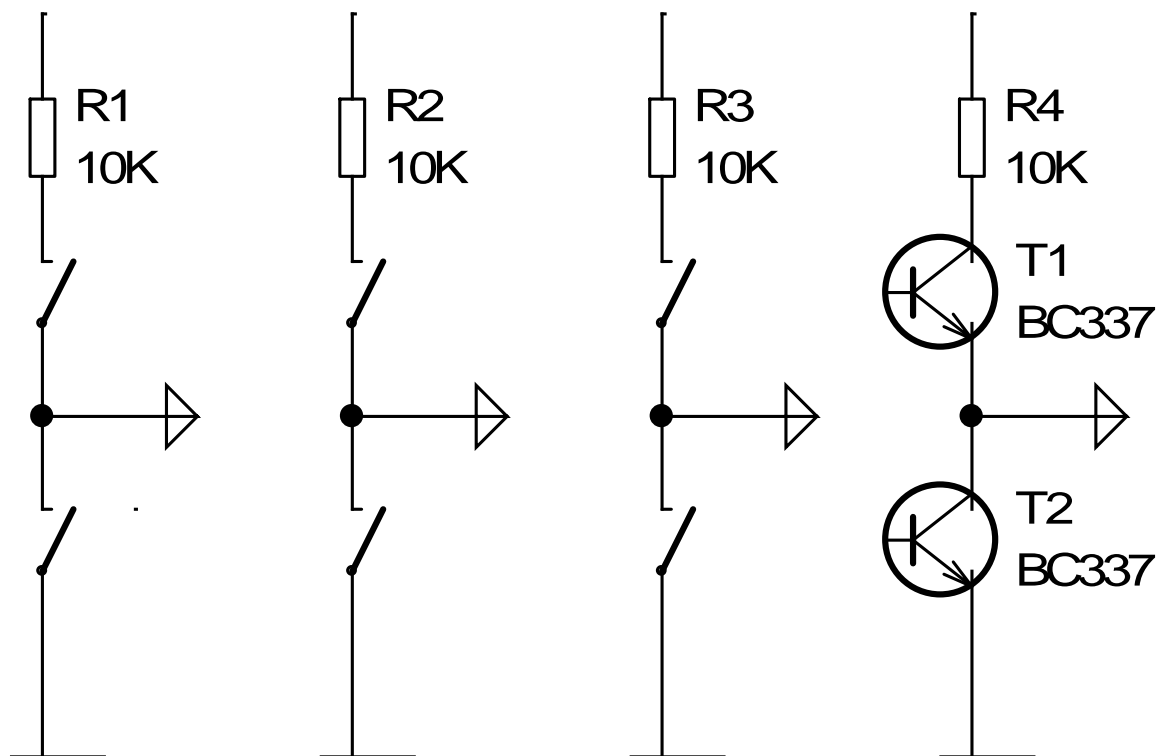
Je skupina signálových vodičů, kterou lze rozdělit na skupiny řídicích, adresových a datových vodičů. Sběrnice má za účel zajistit přenos dat a řídicích povelů mezi dvěma a více elektronickými zařízeními.

V běžných logických sítích je obvyklé, že výstupní signál jednoho obvodu je přiveden na jeden nebo několik vstupů dalších obvodů. K vodiči jimž se přivádí informace je připojen pouze jeden zdroj (budič) a několik snímačů. V technice mikropočítačů je využit jiný princip. Všechny bloky mikropočítače jsou paralelně propojeny souborem vodičů – společnou sběrnici. Tyto bloky však mají schopnost informaci jak snímat, tak ji na sběrnici předávat. Ke sběrnici je tedy připojeno několik zdrojů informace a je tedy třeba určit, který blok bude informaci dodávat a který bude snímat. Jedním z účastníků každého přenosu na sběrnici bývá až na výjimky procesor. Ten v těchto případech rovněž určuje směr přenosu, druhého účastníka přenosu, kdy má být informace na sběrnici platná a další nutné údaje. V technice mikropočítačů sběrnice dělíme na:

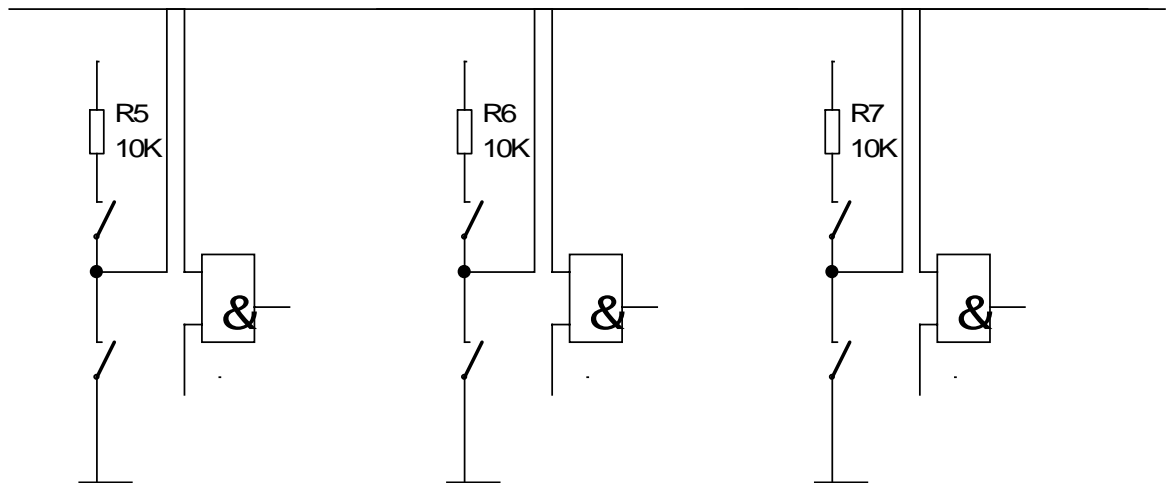
- adresová – přenáší se po ní adresy zdroje a cíle. Pokud chce MP číst data z nějakého bloku nebo zapisovat, musí nějakým způsobem sdělit místo čtení i zápisu a toto místo je identifikováno adresou, která se přenáší po adresové sběrnici. Zdrojem této informace je mikroprocesor. Počet bitů adresové sběrnice (počet vodičů) odpovídá počtu bitů adresy, kterou je schopen mikroprocesor vytvořit a určuje maximální využitelný adresovatelný prostor. Např. osmibitová adresová sběrnice adresuje max  $2^8$ . Těchto prostorů může být i více, to ale závisí na architektuře samotných mikroprocesorů. Univerzálně jsou dva adresové prostory. Každý blok, který s mikroprocesorem spolupracuje, musí být umístěn v některém z těchto dvou prostorů. Tyto prostory nejsou rovnocenné, rozlišení zajišťuje řídicí sběrnice.
- řídicí – je spíše souhrnem individuálních signálů aktivních v různých okamžicích s různým významem. Většinu signálů generuje procesor. Jednotlivé signály mají též různé zdroje informace. Některé jsou generovány mikroprocesorem, jiné mohou být ovlivňovány i jinými bloky. Jednotlivým blokům jsou přivedeny jen ty signály, které se jich týkají. Některé z typických signálů – **reset** (signál, kterým je vybaven každý mikroprocesor a který ho uvádí do základního stavu, aktivuje ho buď uživatel, nebo je generován přídavným obvodem, tímto signálem mohou být uváděny do základního stavu i jiné bloky mikropočítače), **memory read** (signál čtení z paměti, tyto signály zabezpečují časování přenosu informací z paměti do mikroprocesoru nebo mikropočítače), **memory write** (signál zápisu do paměti, tyto signály

zabezpečují časování přenosu informací do paměti z mikroprocesoru nebo mikropočítače), **zápis/čtení z/do výstupu** (některé architektury mikroprocesorů či mikropočítačů podporují oddělené instrukce pro zápis do periferních bloků, pak jejich sběrnice obsahují signály IOW/IOR, pro časování přenosu z/do těchto bloků), **ready** (připravenost obvodu, někdy je k mikroprocesoru nutno připojit obvody, které z hlediska svého časování nevyhovují a nestíhají požadovanou činnost, v takovém případě musí být možnost pozastavit činnost mikroprocesoru, např. tak, že před svou další činností vloží tzv. *čekací smyčky* a např. si na platná data při čtení počká). Řídící sběrnice může obsahovat alespoň jeden hodinový signál pro potřeby jednotlivých bloků. Uvedené řídicí signály se u různých procesorů liší.

- datová – po této sběrnici probíhá přenos všech dat mikropočítače. Data se přenáší vždy mezi dvěma bloky - např. paměť-mikroprocesor a podobně. Tohoto přenosu se účastní mikroprocesor, až na výjimku, jako přijímač a vysílač všech přenosů mikropočítače. Je nutné zabezpečit, aby v jakémkoliv okamžiku byl aktivní pouze jeden vysílač, jinak řečeno *budič sběrnice*. Při nedodržení této podmínky by na datové sběrnici došlo k neurčitosti signálu, v horším případě pak ke zničení jednoho nebo obou vysílačích obvodů. Proto je nutné vybavit bloky připojované na datovou sběrnici obvody, jež umožní odpojení tohoto bloku od sběrnice, je-li to z hlediska přenosu informace, tj. když se zrovna neúčastní přenosu, nutné. Tyto obvody se nazývají třístavové sběrnice



Příklad jednoho vodiče obousměrné datové sběrnice s třístavovými budiči. Na tomto obrázku je zachycen přenos log. 0 z bloku 1 do bloku 3



Důležitým parametrem datové sběrnice je její šířka – počet bitů. Tento parametr má vliv na rychlost, neboť udává, kolik bitů se přenáší najednou. Nemá smysl, aby tento počet byl větší, než kolikabitový je procesor. Maximální délka je omezena bezproblémovým přenosem signálu a bývá nejvýše desítky centimetrů. V některých případech je vhodné šetřit počet vodičů, a proto se provádí, tzv. multiplexování sběrnic.

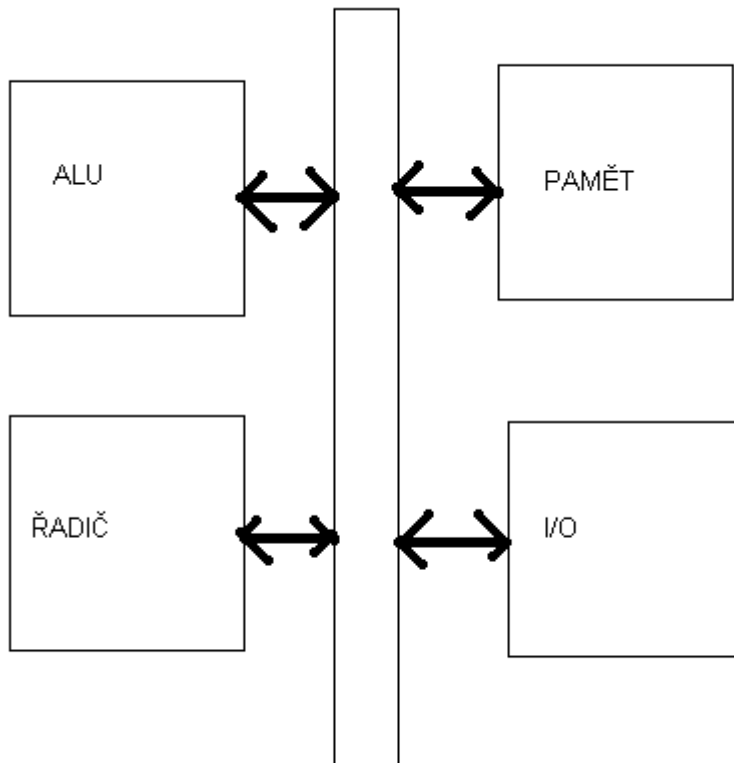
## Rozdělení procesorů

Procesory se může dělit podle různých hledisek do různých skupin:

- a) **CPU (Central Procesor Unit)** – jsou to mikroprocesory, jak je asi známe všichni. Jsou srdci našich počítačů. Jejich hlavními vlastnostmi jsou: vysoký výkon, vysoká spotřeba (jednotky amper) a ztrátový výkon => velmi se zahřívají, velká rozměrná pouzdra, vysoká cena, otevřenost procesoru, který neobsahuje žádná periferní zařízení přímo v sobě a tudíž musí být všechny periferie připojeny externě
- b) **MCU (Micro Controller Unit)** – spíše snesou označení *jednočipové mikropočítače*. Tento typ se používá téměř všude, počínaje osobním počítačem, konče v měřicích přístrojích. Vlastnosti: nízká cena, nejlevnější se dají pořídit i za 50 korun, nízká spotřeba v klidovém režimu, malá rozměrná pouzdra, mnohem menší možnosti rozšíření.
- c) **DSP (Digital Signal Procesor)** – je to určitý kompromis mezi CPU a MCU. Je určen ke zpracovávání signálů. Stručně řečeno tento procesor slouží k tomu, aby data, která do něj vstoupí, zpracoval a co nejrychleji předal na výstup. Vlastnosti: vysoká rychlost zpracování číslicových dat, velmi rychlé matematické operace, schopnost zpracovávat velké objemy dat.

Dělení podle architektury

- a) **Architektura Von Neumann** – tato architektura se stala základem všech mikroprocesorů, další architektury se pak lišili pouze v základech.



Mikroprocesor se skládá ze čtyř funkčních bloků – paměti, řadiče, ALU a I/O. Struktura zařízení je nezávislá na typu řešení úlohy – činnost mikroprocesoru je řízena obsahem paměti. Instrukce a operandy jsou uloženy v téže paměti. Paměť je rozdělena do buněk stejné velikosti, jejichž pořadová čísla se používají jako adresy. Program, je tvořen posloupností elementárních příkazů (instrukcí), v nichž zpravidla není obsažena hodnota operandu (uvádí se pouze jeho adresa), takže program se při změně dat nemění. Instrukce se provádějí jednotlivě v pořadí, v jakém jsou zapsány do paměti. Změna pořadí provádění instrukcí se vyvolá podmíněného nebo nepodmíněného skoku. Výhody: plynule se dá měnit velikost prostoru pro data a pro program v paměti. Nevýhody: všechny operace procházejí přes společnou sběrnici, což zpomaluje proces; vzniká možnost přepisu programu daty a naopak.

**b) Architektura Harvard** – vychází z VON NEUMANNOVY architektury. Zabývá se oddělením paměti dat od paměti programu. Výhody: nemůže dojít k přepisu programu daty, každá paměť je připojena k ALU přes vlastní sběrnici, umožňuje paralelní přenos dat a instrukcí => větší rychlost. Nevýhody: při návrhu nutno znát předem kolik bude potřeba prostoru pro data a pro program.

