Komunikace s periférie mi

Studijní materiál pro předmět Architektury počítačů a paralelních systémů

Ing. Petr Olivka, Ph.D.
katedra informatiky FEI VŠB-TU Ostrava
email: petr.olivka@vsb.cz

Ostrava, 2020

1 Úvod

V dnešní době, kdy je enormní nárůst využití počítačů, je potřeba se věnovat neustálému vývoji sběrnic a jejich komunikace s periferními zařízeními. Mnohdy se právě jedná o sběrnice, kdy celkovou rychlost práce s počítačem negativně ovlivňuje jejich rychlost.

Tento studijní materiál se bude zabývat technickými a programovými možnostmi komunikace mezi CPU a periferiemi. V minulosti bylo toto téma často podceňováno a i v dnešní době se najde spousta lidí, kteří otázku komunikace s periferiemi podceňují, zejména pak s rozvojem periferních zařízení.

2 Sběrnice

Patří mezi součásti každého počítače i samotného mikroprocesoru. Sběrnice slouží pro přenos informace mezi mikroprocesorem a ostatními částmi počítače, ale také mezi částmi uvnitř mikroprocesoru nebo také mezi počítačem a okolím.

Logické sítě jsou obvykle složeny z jednoho obvodu, kde je jeho výstupní signál přiveden na jeden nebo několik vstupů dalších obvodů. To znamená, že k vodiči, jímž jsou přenášeny informace, je připojen pouze jeden její zdroj (budič) a několik snímačů. Dnes je v technice počítačů použit princip, kdy všechny bloky počítače jsou paralelně propojeny souborem vodičů společnou sběrnicí z anglického **bus**. Tyto bloky počítače jsou schopny informace snímat, ale i předávat. Pro smysluplné připojení a k tomu, aby nedocházelo ke kolizi připojení několika zdrojů informace ke sběrnici, musíme určit:

- který blok bude informaci na sběrnici dodávat a který snímat,
- kdy je informace na sběrnici platná.

Mezi nejčastější účastníky každého přenosu, uskutečněném na sběrnici, bývá téměř vždy mikroprocesor, který v těchto případech rovněž určuje další údaje přenosu (např. směr přenosu, druhého účastníka, kdy má být informace na sběrnici platná a další nutné údaje). Sběrnice dělíme na *adresovou, řídící a datovou.*

2.1 Adresová sběrnice

Pokud chce mikroprocesor data číst nebo je zapisovat, musí nějakým způsobem sdělit místo čtení i zápisu. Toto místo je identifikováno *adresou*, která je přenášena po adresové sběrnici. Zdrojem této informace je mikroprocesor.

Počet bitů adresové sběrnice, neboli počet vodičů, odpovídá počtu bitů adresy. Mikroprocesor vytváří tuto adresu a určuje tak využitelný adresový

prostor. Jako příklad nám poslouží např. šestnáctibitová adresová sběrnice, která maximálně adresuje $2^{16} = 65536$ adres.

Univerzální mikroprocesory mívají dva adresové prostory. Jeden adresový prostor pro adresování paměti a druhý pro adresování vstupů a výstupů. Každý blok komunikující s mikroprocesorem musí být umístěn v některém z těchto dvou prostorů. Tyto prostory nejsou totožné a rozlišení těchto adresovatelných prostorů zajišťuje řídící sběrnice.

2.2 Řídící sběrnice

Řídící sběrnice je souhrn jednotlivých signálů aktivních v různých časových okamžicích s různým významem, které mívají různé zdroje. Některé signály jsou generovány mikroprocesorem, některé mohou být ovlivňovány jinými bloky. K jednotlivým blokům jsou pak přivedeny pouze signály jim patřící. Mezi nejčastější signály řídící sběrnice patří například:

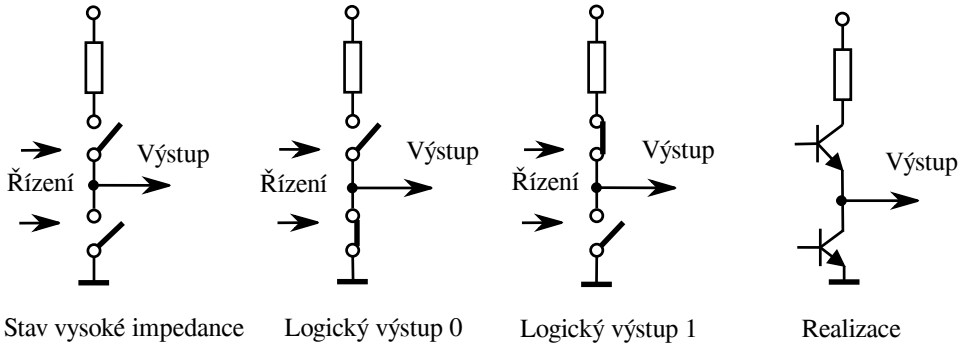
- Signál **RESET**, jež je vybaven každý mikroprocesor. Aktivován uživatelem nebo jiným přídavným obvodem. Tento signál uvede mikroprocesor do jeho výchozího stavu,
- Signál **Memory Read (MR)**, zabezpečuje časování čtení z pamětí či jiných bloků do mikroprocesoru nebo počítače,
- Signál **Memory Write (MW)**, zabezpečuje časování zápisu do pamětí či jiných bloků z mikroprocesoru nebo počítače,
- Signály **Input/Output Read/Write (IOR/IOW)**, slouží pro čtení z, nebo zápis do zařízení,
- Signál **READY**, určuje připravenost obvodu.

Řídící sběrnice může obsahovat i další řídící signály pro potřeby jednotlivých bloků, ale ty se ovšem liší pro různé mikroprocesory.

2.3 Datová sběrnice

Slouží pro přenos všech dat v počítači. Data jsou vždy přenášena mezi dvěma bloky počítače. Typickým příkladem tohoto přenosu je přenos z paměti do mikroprocesoru. Mikroprocesor se účastní, až na výjimku jako přijímač a vysílač, všech přenosů v počítači. V praxi je nutné, 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 vysílacích obvodů. Proto je tedy

nutné vybavit bloky připojované na datovou sběrnici obvody, jimiž umožňuje odpojení tohoto bloku od datové sběrnice. Takové obvody nazýváme **třístavovými budiči sběrnice**, jejichž princip a realizace je na obrázku 1. Třístavové budiče sběrnice mohou být jednosměrné nebo obousměrné. Tyto budiče mají kromě schopnosti oddělovat od sběrnice také důležitou úlohu výkonového posílení.



Obrázek 1: Obousměrná datová sběrnice

Mezi nejdůležitější parametry datové sběrnice patří její šířka neboli počet bitů a časování. Šířka datové sběrnice udává, kolik bitů je schopno se přenést najednou a patří mezi parametry výrazně ovlivňující rychlost komunikace. Ovšem nemá význam, aby bitová šířka datové sběrnice byla větší jak bitová šířka mikroprocesoru.

Jednou z možností jak „ušetřit“ počet vodičů, je možnost **multiplexování sběrnic**. Multiplexování spočívá ve vedení signálů dvou sběrnic (nejčastěji adresové a datové) ve společných vodičích takovým způsobem, že jsou jednotlivé signály aktivní v různou dobu. Signály řídící sběrnice určují význam právě přenášených dat na multiplexované sběrnici.

Sběrnice rozeznáváme:

Vnitřní sběrnice mikroprocesoru – architektura je dána převážně architekturou mikroprocesoru.

Vnitřní sběrnice počítače – propojuje mikroprocesor s ostatními prvky počítače a dovoluje tedy vytvoření různých uspořádání počítače, optimální cenově a funkčně pro danou aplikaci.

Vnější sběrnice počítače – sběrnice styku s okolím či sběrnice multiprocesorového systému se vyskytuje ve složitějších systémech, sestavených z několika počítačů. Sběrnice jsou pro takový účel poněkud rozšířeny a upraveny pro rozhodování o prioritě podřízených počítačů. Priorita se může v průběhu algoritmu dynamicky měnit podle okamži-

tých potřeb systému. Jindy se pomocí vnější sběrnice počítače připojují k danému počítači další přídavná zařízení.

3 Připojení zařízení ke sběrnici, adresový dekodér

Přenosy dat na sběrnicích se mohou uskutečnit řízením technickými prostředky nebo programovým řízením. Na přenosu dat *řízeným technickými prostředky* se podílí mikroprocesor jen částečně nebo je úplně vyřazen. Takový přenos je řízen řadičem přímého přístupu do paměti (DMA), viz. kapitola 4.4. Využívá se pro přenosy velkých bloků dat, například pro přenos dat z přídavných zařízení do paměti RAM.

Na přenosu dat *programovým řízením* se na tomto přenosu podílí výhradně mikroprocesor tím, že postupně provádí instrukce účelně seřazené do segmentu řídícího programu. Mezi velké výhody patří levná realizace a snadná změna algoritmu přenosu změnou příslušného segmentu řídícího programu. Mezi nevýhody patří pak vlastnost pomalejších mikroprocesorů, kde je celkově pomalý přenos.

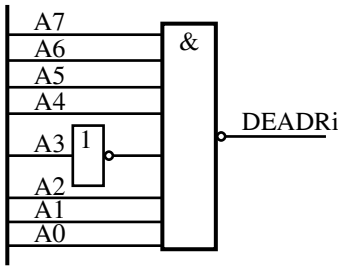
Přenos dat s programovým řízením je obecně přenosem mezi dvěma registry. U mikroprocesoru probíhá velmi často mezi některými vnitřními registry, případně mezi vnitřním registrem a buňkou vnější paměti. Přenosy mezi vnitřními a vnějšími registry lze rozdělit na přenosy paměťové a vstupně/výstupní. To znamená, že umožňuje-li architektura mikroprocesoru oba tyto přenosy, mívají pak tyto dva adresové prostory (prostor pro adresování paměti a vstupů/výstupů). Výběr prostoru je pak určen příslušnými signály pro určitý prostor.

Umístění periferních zařízení do některého z těchto prostorů nazýváme *mapováním*. Mezi výhody mapování patří mapování periferií do V/V prostoru a spočívá v rychlejším přístupu do tohoto prostoru, zpravidla instrukcemi vstupu (IN) a výstupu (OUT). Výhoda mapování do paměťového prostoru spočívá ve větším množství použitelných instrukcí pro přenos dat (k dispozici jsou všechny instrukce používané pro práci s pamětmi). *Mapou paměti* si můžeme znázornit obsazení dílčích úseků paměťového prostoru jednotlivými pamětmi.

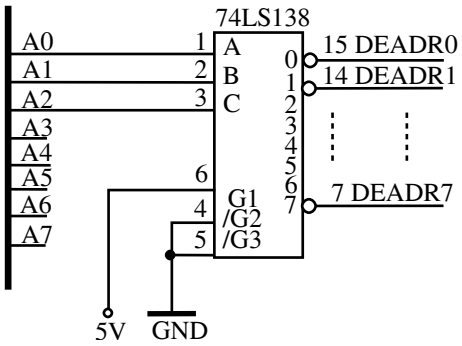
Paměťový prostor bývá obsazen více jak jednou fyzickou pamětí nebo periferním zařízením, a proto je nutné při přenosech dat s mikroprocesorem rozhodnout, které zařízení je ke komunikaci určeno. Tímto úkolem je právě pověřen *adresový dekodér*. Jeho výstupy jsou v podstatě signály CS (Chip Select) pro jednotlivé obvody. Signál CS připojuje daný obvod k datové sběrnici tak, že jeho sběrnici přepne ze stavu vysoké impedance do aktivního stavu.

Adresový dekodér může být stavěn jako dekodér pro:

- úplné dekódování adresy,



Obrázek 2: Princip úplného dekodéru adresy



Obrázek 3: Princip neúplného dekodéru adresy

- neúplné dekódování adresy,
- lineární přiřazování adresy,
- univerzální přiřazení dekódovaných adres.

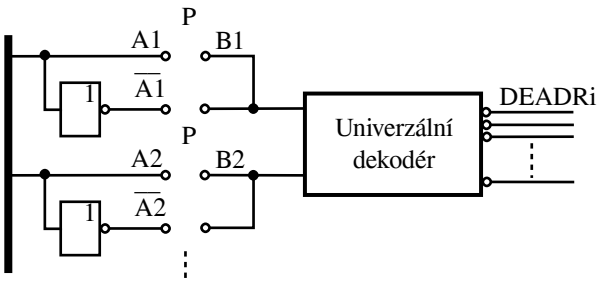
3.1 Adresový dekodér s úplným dekódováním adresy

Při úplném dekódování je jisté *unikátní adrese* ADRi přiřazen pouze jeden signál DEADRi a obráceně, kde jistému signálu DEADRi odpovídá pouze jedna adresa ADRi.

Příklad takového dekodéru je uveden v obrázku 2. Realizace je pomocí hradla AND.

3.2 Adresový dekodér s neúplným dekódováním adresy

Pro adresování dekodéru s neúplným dekódováním adresy nejsou uvažovány některé řády adresy s tím, že jistému signálu DEADRi přísluší adresový prostor ADRi, kde i=1,2,3... Takto postavený adresový dekodér je sice levnější, ale vyžaduje pozorné přiřazování adres k jednotlivým adresovaným



Obrázek 4: Princip univerzálního dekodéru adresy

obvodům. S těmito dekodéry se můžete velmi často potkat při přidělování paměťových prostorů k jednotlivým typům pamětí[1] nebo též při přidělování k přídavným periferiím v počítačovém systému. Příklad neúplného dekodéru je na obrázku 3.

3.3 Adresový dekodér s lineárním přiřazením adresy

Jednotlivé řády adresy těchto dekodérů jsou pokládány přímo za výstupní výběrové signály dekodéru DEADRi = ADRi. Je to tedy nejlevnější adresový dekodér. Jeho nevýhodou je možnost připojit pouze m přídavných zařízení u adresy s n řády.

Kombinace zde popsaných typů dekodérů patří v dnešní době mezi nejpoužívanější typy dekodérů. Jinak se sestaví dekodér pro předem známou aplikaci s konečnou, více nerozšiřitelnou sestavou počítače a jinak pro předem neznámé aplikace s požadavkem modulové rozšiřitelnosti počítače. Univerzální řešení je zpravidla složitější a dražší. Jedním z takových univerzálních adresových dekodérů je na obrázku 4, který má v každé adresní lince Ai vložen *invertor* a patřičně konstruovaný přepínač P (například DIP), u kterého můžeme libovolně zvolit skutečnou vstupní adresu. Toto nastavování probíhá jen před zasunutím modulu do rozšiřovaného počítače.

4 Řízení komunikace

Existují dva případy zahájení komunikace počítače nebo mikroprocesoru s periferiemi:

- **Zahájení komunikace z iniciativy programu** – Jde o případ, kdy počátečním příkazem k zahájení komunikace je instrukce probíhajícího programu. Příkladem může být například algoritmus programu, který potřebuje *přijímat z*, resp. *vysílat do* okolí informace. Nutnou podmínkou je schopnost dané periferie přijmout resp. vyslat informaci v daný

[1] EPROM, RAM, FLASH atd

okamžik nebo musí informaci v okamžik určený počítačem vystavit. Příkladem těchto periferií mohou být stavové spínače nebo sdělovače[2].

- **Zahájení komunikace z iniciativy periferie** Zahájení komunikace z iniciativy periferie nastává v případě, že jistá periferie chce poslat resp. přijmout svou informaci z resp. do počítače. Zde však nastává problém v navázání komunikace, neboť se počítač může nacházet v činnosti, kdy ihned nemůže s periferií komunikovat. Počítač se však nějakým způsobem musí dozvědět, že je vyžadována komunikace a která periferie je iniciátorem výzvy.

V případě, že je komunikace aktivovaná periferií, ne počítačem, lze postupovat několika způsoby:

- Obvodovým řešením daného periferního zařízení. Určitou operaci může uskutečnit tak, aniž by o tom musel vlastní počítač vědět. Toto bývá řešeno pomocí obvodů nízké a střední integrace.

- Pro inicializaci určité operace zařízením použijeme *příznakový bit (Flag)* nebo skupinu příznakových bitů. Tato hodnota je čtena počítačem. Pokud je hodnota rovna 1, může iniciovat určitou operaci. Pomocí příznakových bitů oznamuje periferní zařízení svůj stav (připraven, zaneprázdněn, porucha atd.). Skupina příznakových bitů vytváří *stavové slovo (Status Word)*. Počítač tedy nejprve přečte stavové slovo dodané periferií a na základě analýzy tohoto slova rozhodne o další činnosti. Tomuto zařízení budeme říkat **programové řízení**.

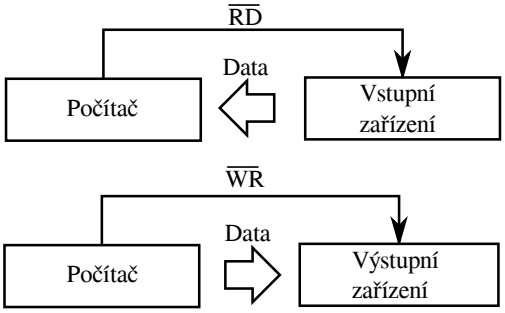
- Zařízení, které chce inicializovat nějakou operaci, pomocí speciálního signálu procesoru přeruší právě probíhající program. Procesor přeruší svou činnost a vykoná požadovanou akci komunikace. Poté se vrátí tam, kde byl procesor přerušen a pokračuje v původní činnosti. Pro obsluhu komunikace mezi více zařízeními tímto způsobem, musí mít mikroprocesor nebo počítač k dispozici obsáhlejší **systém přerušení**.

- Pokud operace, kterou chce periferní zařízení inicializovat, spočívá v přenosu bloku dat z periferního zařízení do paměti počítače nebo naopak, může se tato operace uskutečnit pomocí **přímého přístupu do paměti (DMA)**.

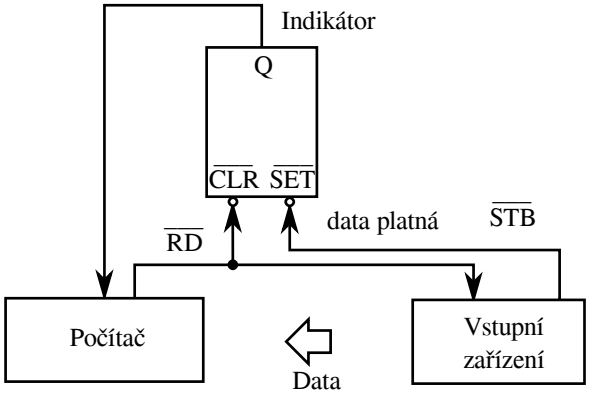
4.1 Technika V/V bran

Vstupně/Výstupní brána (z angl. *Input/Output I/O*) je obvodem, který zprostředkovává předávání dat mezi sběrnicí počítače a periferním zařízením počítače. Máme zde možnost použití brány s/bez pamětí. Oba typy bran jsou

[2] signalizační zařízení



Obrázek 5: Technika nepodmíněného vstupu a výstupu dat



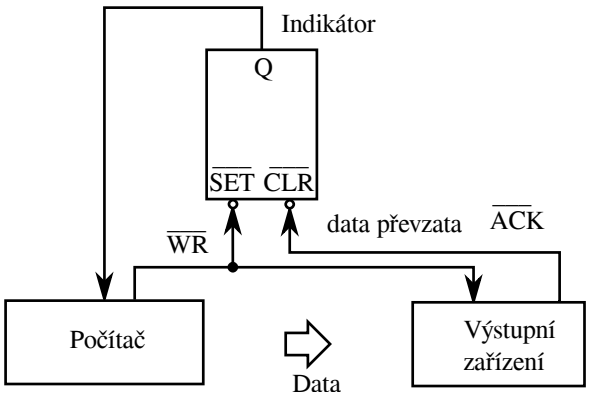
Obrázek 6: Technika podmíněného vstupu dat

výkonové zesilovače (*budiče*) jednosměrně nebo obousměrně řízené. Základem bran bývá tzv. *záchytný registr (latch)* s třístavovým výstupem. Na obrázku 5 je zobrazena technika nepodmíněného vstupu a výstupu dat.

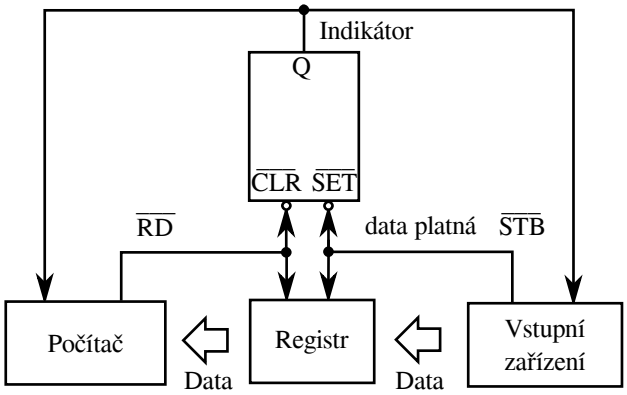
Při vstupu vyšle počítač signál $\overline{RD}$, čímž přikáže vstupnímu zařízení předat data do vstupní brány počítače. Při výstupu počítač vyšle současně data i signál $\overline{WR}$ a výstupní zařízení převezme data. Tento způsob je mimořádně jednoduchý a předpokládá stálou připravenost periferního zařízení komunikovat.

Technika podmíněného vstupu/výstupu je zobrazena na následujících obrázcích, kde na obrázku 6 je zobrazen vstup. Jsou-li poskytována platná data ze vstupního zařízení, pak se za pomocí snímacího impulsu *STB (strobe)* nastaví hodnota $Q = 1$. Tento stav Q je označován jako *indikátor (angl. flag)*. Pokud je tento stav, kdy je $Q = 1$ platný, jsou předány data počítačem za pomocí impulsu $\overline{RD}$ a po uskutečnění tohoto přenosu je tento indikátor vynulován.

Následně výstup podle obrázku 7, kdy počítač vyšle impuls $\overline{WR}$ pro



Obrázek 7: Technika podmíněného výstupu dat



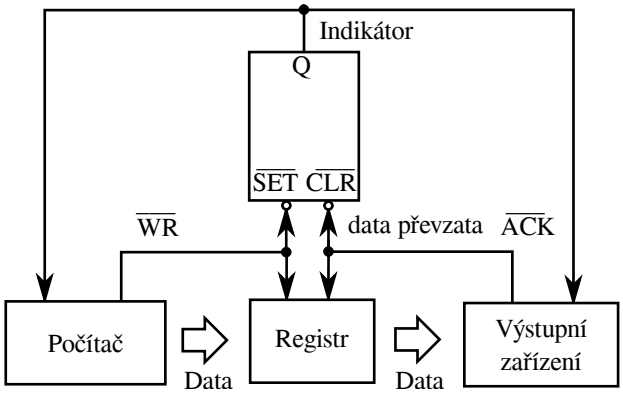
Obrázek 8: Technika vstupu dat s vyrovnávací pamětí

přepis dat do výstupního zařízení a následnému nastavení indikátoru. Výstupní zařízení po převzetí dat impulsem *ACK* indikátor nastaví na nulovou hodnotu. Touto hodnotou je počítači sděleno, že může vyslat další data.

V obou případech jde o **neúplný (jednosměrný korespondenční) režim**, kdy pomocí indikátoru jsou sdělovány informace pouze počítači o zahájení nebo ukončení přenosu a kdy vysílač dat je povinen data udržovat.

Technika **úplného (obousměrného korespondenčního) režimu** využívá vyrovnávací paměti (registru) a klopný obvod pracující jako semafor (jeho stav je testován vysílačem i přijímačem dat). Přichází zde i v úvahu možnost *vzájemného blokování (interlock)*.

Schéma vstupu dat s vyrovnávací pamětí je vidět na obrázku 8. Semafor informuje procesor o připravenosti dat ve vyrovnávací paměti. Pro periférii tento semafor představuje informaci, zda je možno do vyrovnávací pamětí zaslat další data, či zda obsah paměti nebyl ještě procesorem přečten.



Obrázek 9: Technika výstupu dat s vyrovnávací pamětí

Odpovídající schéma pro výstup dat je na obrázku 9. Význam signálu semaforu je pro procesor a periférii opačný, než v předchozím případě.

4.2 Programové řízení komunikace

Využívají se zde instrukce pro vstup a výstup ve spojení s instrukcemi umožňující testování logických proměnných a instrukcí skoků.

Použití principu programového řízení je velmi jednoduché u těch počítačů, které mají přímo vnější vstup příznakového (stavového) bitu a instrukce podmíněných skoků, umožňující větvení programu podle hodnoty příznakového bitu.

Organizaci programového vybavení obsluhy komunikace musíme přizpůsobit vzhledem k našemu požadavku na včasné zjištění hodnot stavových bitů. Typickým případem jsou data z klávesnice, kdy informace nevyžadují příliš častou pozornost mikroprocesoru.

Program určitých zařízení lze sestavit tak, že vykonává opakovaně určitou posloupnost instrukcí. Tato posloupnost je vzhledem k rychlosti dnešních mikroprocesorů a počítačů natolik krátká, že by nemělo činit problémy snímat stavový bit mnohem častěji např. každých 10 ms. Pokud tato doba trvání je delší, můžeme toto testování zařadit vícekrát (např. voláním podprogramu) do posloupnosti instrukcí algoritmu programu. Jisté případy mohou vzniknout, kdy mikroprocesor při určité fázi komunikace s periferním zařízením nastaví stavový bit a bude v určité krátké době vyžadovat odezvu na tento stav, jsou řešeny *čekací smyčkou*. Tím je míněno, že počítač testuje daný stavový bit, dokud nebude nastaven.

4.3 Řízení komunikace pomocí přerušení

Systém přerušení mikroprocesoru usnadňuje programové řízení spolupráce s periferními zařízeními především ve fázi zjišťování žádosti o obsluhu. Jakmile periferní zařízení požaduje od počítače obsluhu (např. přenos dat, různá hlášení atd.), aktivuje přerušovací signál vstupující do mikroprocesoru. Po zjištění žádosti o obsluhu procesor přeruší právě probíhající program a přejde do obslužného programu. Po jeho dokončení se procesor vrátí do původního místa v programu, kde byl přerušen a pokračuje v dalším provádění hlavního programu, v kterém byl přerušen.

U tohoto řízení však nastává problém při obsluze více zařízení a je tak potřeba rozsáhlejšího přerušovacího systému. Informace, které zařízení žádalo o obsluhu je nejčastěji poskytnuta ve formě *vektoru přerušení*, což je adresa příslušného obslužného programu.

Princip žádostí mezi periférierni a procesorem je podrobněji vysvětlen v kapitole 6.

Některé možnosti identifikace zařízení žádajícího o obsluhu:

1. Požadavky na obsluhu jednotlivých zařízení jsou logicky sečteny a procesor přečte stavové slovo obsahující identifikační znak zařízení žáda-

jící o obsluhu. Pokud o obsluhu žádá více jak jedno zařízení najednou, musíme toto přijetí požadavků programově ošetřit.

2. Požadavky na obsluhu zařízení jsou opět logicky sečteny a signál potvrzení žádosti od mikroprocesoru je sériově veden přes všechna periferní zařízení. Priorita je tedy dána zařazením jednotlivých zařízení vůči tomuto signálu. Pokud nastane žádost o přerušení, procesor vyšle potvrzující signál nejprve prvnímu zařízení (zařízení s nejvyšší prioritou). Pokud toto zařízení nežádalo o přerušení, je tento signál propuštěn dál. Jakmile signál dojde k zařízení, jež požadavek o přerušení vydalo, je nutné zabránit dalšímu průchodu tohoto signálu k dalším zařízením. Procesoru předáme *vektor přerušení*, tj. adresu, na které se nachází obslužný podprogram. Tento způsob již vyžaduje přídavné řešení technickými prostředky.

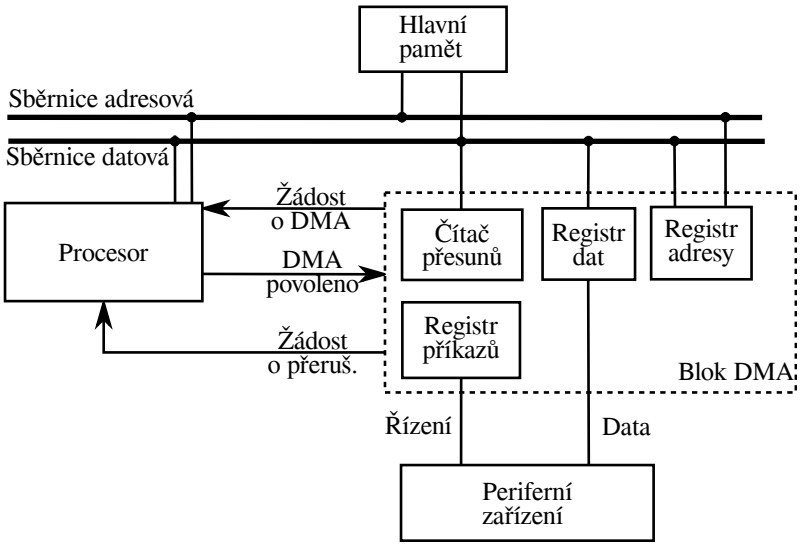
3. Další možností je použití *řadiče přerušení*, jehož princip spočívá ve vyslání identifikačního znaku daného zařízení až po přijetí žádosti o přerušení. Dále pak v přiřazování priority a její dynamické změny jednotlivým zařízením při více žádostech najednou a také zajišťuje maskování zdrojů.

Žádosti o přerušení jsou nejprve porovnány s registrem masky, který rozhoduje o jejich maskování. Tyto žádosti jsou nejprve uloženy v registru žádosti o přerušení. Poté jsou přijaty tyto požadavky na přerušení blokem priorit, rozřazeny podle jejich priority a vybrán jeden požadavek s nejvyšší prioritou. Tento požadavek s nejvyšší prioritou, který pošle procesoru kompletní informace o adrese, na které je program obsluhy přerušení. Tento program je potřeba před jeho činností naprogramovat. Naprogramování spočívá v nastavení priority jednotlivých zdrojů, masky a v neposlední řadě i adresy obsluhy přerušení příslušným zdrojům.

4.4 Přímý přístup do paměti DMA

V počítači existují události, na které nelze reagovat jinak než přerušením. Princip operace spočívá v prostém přesunu informací z periferního zařízení do hlavní paměti nebo naopak. To lze realizovat výhodněji formou **přímého přístupu do paměti** označované zkratkou **DMA** (z anglického výrazu Direct memory access).

Princip DMA spočívá v přímém přesunu informací bez účasti procesoru mezi vyrovnávacím registrem periferního zařízení a hlavní pamětí. Tedy procesor se vůbec neúčastní tohoto přesunu a tudíž nemusí ukončovat své aktuálně běžící programy, aniž by je přerušoval. Jedinou nutnou podmínkou je „uvolnění“ sběrnice pro procesor, to znamená, že procesor na dobu přesunu přepne všechny budiče sběrnic do *třetího (vysokoimpedančního) stavu*. Sběrnice nemohou zůstat po dobu přesunu bez řízení. A právě tomu zabraňuje



Obrázek 10: Struktura bloku pro přímý přístup do paměti (DMA)

existující další blok, který je schopný generovat adresu a určovat okamžiky přesunu dat po datové sběrnici. Tento blok se označuje jako blok DMA, někdy též označovaný kanál DMA.

Příklad struktury bloku DMA vidíme na obrázku 10. V tomto bloku pro přímý přístup do paměti jsou určeny tři registry pro styk se sběrnicemi označené jako:

- registr dat – obsahuje slovo, jež má být přesunuto z periferního zařízení do hlavní paměti nebo naopak,
- registr adresy – slouží pro uchování adresy hlavní paměti, což je adresa, na kterou bude zapsáno slovo nebo ze které bude dané slovo přečteno,
- čítač přesunů – požadovaný počet slov, které mají být ještě přesunuty v rámci jednoho spojení mezi periferním zařízením a hlavní pamětí.

Blok DMA může pracovat ve dvou režimech:

- přesouvání dat mezi periferním zařízením a hlavní pamětí jednotlivě,
- přesouvání dat v blocích.

Celá operace přímého přístupu do paměti obvykle probíhá v několika krocích:

1. Naprogramování procesorem bloku DMA.

2. Blok DMA spustí periferní zařízení a čeká, až bude toto periferní zařízení připraveno buď data přijmout nebo vyslat, jakmile periferní zařízení oznámí bloku DMA, že je připraveno žádat blok DMA o přímý přístup do paměti.

3. Procesor nejprve dokončí strojový cyklus a poté reaguje na žádost o DMA. Přímý přístup k paměti se pak uskutečňuje průběžně během normální činnosti procesoru tak, že blok DMA vysílá data synchronně s fází hodin Φ_1, zatímco procesor používá paměť vždy ve fázi Φ_2.

4. K přímému přístupu do paměti může dojít v okamžiku, kdy vyšle procesor vybrané jednotce signál *ACK* a uvolní sběrnice. Vybraná jednotka pak vyšle na adresovou sběrnici obsah svého registru adresy, na datovou sběrnici obsah svého registru dat a čeká na provedení jednoho cyklu paměti. Pak zvětší obsah registru adresy o jedničku a zmenší obsah čítače přesunů. Není-li obsah čítače přesunů dosud nulový, testuje, zda periferní zařízení již přesunulo nové slovo do registru dat. Pokud nebylo přesunuto nové slovo, ukončí se dočasně přesun dat a přestane vysílat žádost o DMA a tím předá řízení procesoru.

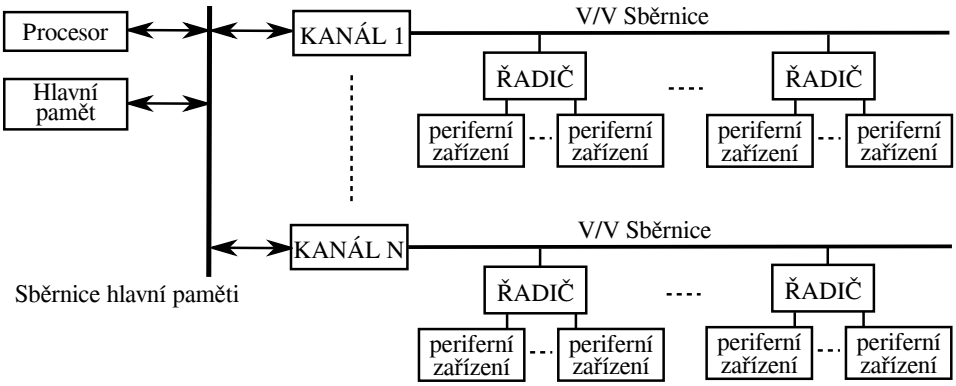
5. Procesor dále pokračuje v provádění svého programu až do okamžiku, kdy některý blok DMA vyšle novou žádost o DMA, signalizující připravenost dat z periferních zařízení v registru dat.

6. Pokud je obsah čítače přesunů nulový, blok DMA končí celý přesun a uvolní sběrnice. Navíc může vyslat žádost o přerušení, čímž si vyžádá programový zásah procesoru, který spočívá v novém naprogramování jeho vnitřních registrů.

Řadič DMA adresuje hlavní paměť a synchronizuje činnost mezi sběrnicí a periferním zařízením, zatímco data jdou mimo něj. Dále existují i takové přenosové subsystémy, které jsou založeny na podobném principu jako u DMA. Mají jistou inteligenci a jsou schopny data i kontrolovat, opakovat a přepracovávat. Jde o **kanál**, což je speciální procesor určený ke spojení hlavního procesoru s periferními zařízeními. Řídí se kanálovým programem. Kanály mohou být multiplexní a selektorové, ale podrobněji o nich v kapitole 5.

5 Kanálová architektura

Kanál je specializovaný programově řízený procesor, který je schopen se samostatně řídit V/V operacemi. Je umístěn mezi procesorem a řadičem periferního zařízení. Mezi obvyklé způsoby propojení patří jeho využití v architektuře počítače, naznačeno na obrázku 11. Jeden systém může obsahovat



Obrázek 11: Kanálová architektura počítače

i několik kanálů, které jsou připojeny na sběrnici hlavní paměti a procesoru. Hlavní paměť řídí „organizátor“ nebo „přidělovač“ hlavní paměti, jemuž se kanál musí podřídit jako každý jiný účastník. Každý kanál řídí periferní sběrnici, k níž mohou být připojena jednotlivá periferní zařízení.

U některých systémů lze navíc připojit jeden kanál k několika počítačům současně, čímž lze vytvářet multipočítačové systémy. Propojením kanálů dvou různých počítačů pomocí tzv. adaptéru kanál – kanál, jež se připojuje na V/V sběrnici, jsme schopni vytvořit multipočítačové systémy.

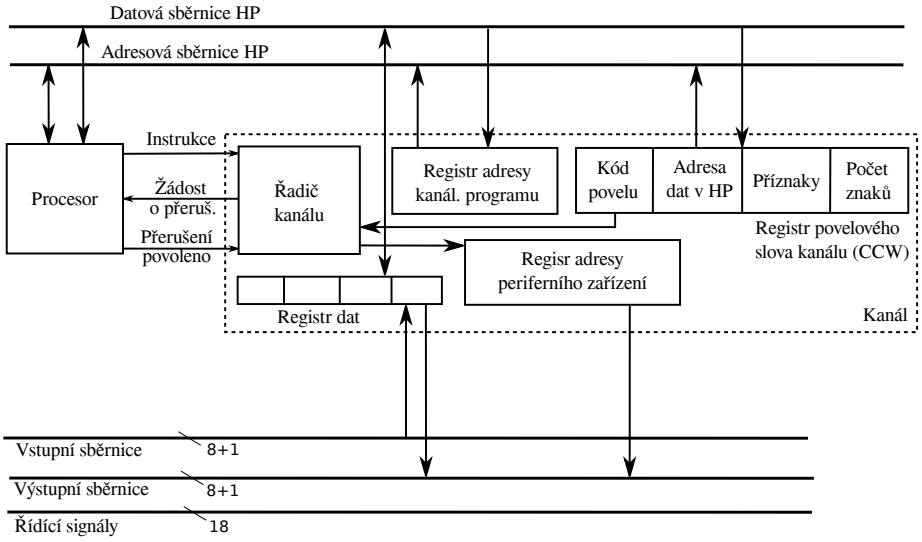
Vlastní činnost kanálu během V/V operace je pak řízena *kanálovým programem*, prováděným přímo v kanálu. Kanálový program je uložen v hlavní paměti a čte se po *povelových slovech (CCW)*, která se jednotlivě přenášejí do registru CCW v kanálu, kde je řízena jeho činnost. Prováděné operace jsou použity především pro řízení přenosu dat mezi registrem periferního zařízení a hlavní pamětí.

5.1 Struktura a funkce kanálů

Kanál má též možnost přímého přístupu do hlavní paměti a v mnohém se neliší od DMA. Jednou z částí, kterou se odlišuje kanál od DMA, je přítomnost vlastního řadiče, náležící kanálu, který mu umožňuje provádět kanálový program. Vnitřní struktura kanálu je znázorněna na obrázku 12.

Struktura kanálu je závislá na tom, zda se jedná o *selektorový* nebo o *multiplexní kanál*. Hlavním rozdílem mezi selektorovým a multiplexním kanálem je to, že multiplexní kanál může obsluhovat více periferních zařízení současně [3], kdežto selektorový pouze jen jedno zařízení. Selektorový kanál se používá k připojení velmi rychlých periferních zařízení, především disků.

[3] několik kanálových programů



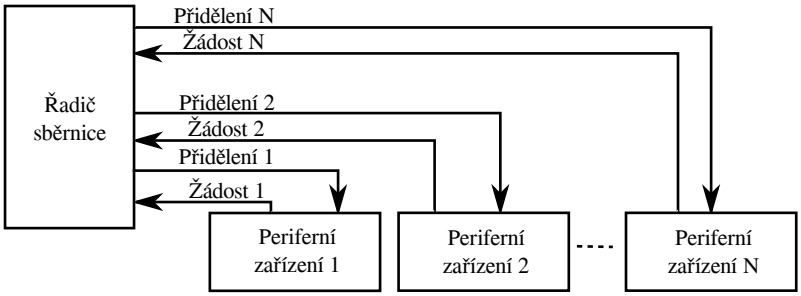
Obrázek 12: Struktura kanálu

Průběh přesunu dat řízený selektorovým kanálem probíhá tak, že nejprve kanál naváže spojení s určitým periferním zařízením, potom vydá příkaz ke spuštění V/V operace, provede se přesun bloku dat z hlavní paměti do periferního zařízení nebo naopak. Poté se spojení ukončí. Po ukončení spojení se může navázat spojení s dalším periferním zařízením, které je na něj připojeno. Během tohoto spojení se přesune opět celý blok dat. Selektorové kanály se přestaly používat a nahradily je blokové multiplexní kanály stejných přenosových rychlostí a mají navíc některé další výhody.

Aby byl schopen provádět několik kanálových programů současně, je rozdělen kanál na *podkanály*, které jsou přiděleny jednotlivým periferním zařízením. Podkanálů bývá až 256[4]. Multiplexní kanál se označuje jako *slabikový* nebo *blokový* podle toho, zda-li přesouvá jednotlivé slabiky nebo celé bloky mezi hlavní pamětí a periferním zařízením.

Slabikový multiplexní kanál se používá pro obsluhu pomalých zařízení jako je klávesnice, tiskárna nebo komunikační linky či spuštění určitého zařízení. Přesun několika slabik po sobě jdoucích z jednoho periferního zařízení se označuje jako *dávkový (burst mode)*. Mezi jeho výhody patří zvýšená přenosová rychlost téměř o řád až několika stovek tisíc slabik za sekundu.

[4]maximální počet připojitelných periferních zařízení adresovatelných ve standardním styku



Obrázek 13: Samostatné žádosti

.

6 Vyhodnocování žádostí a jejich priorit

K řízení činnosti počítačového systému se často používají **žádosti**, což jsou signály signalizující připravenost zařízení vykonávat určitou činnost [5]. Tyto žádosti mohou přicházet z různých zařízení zcela nekoordinovaně a tím vzniká nebezpečí vzájemné konkurence těchto signálů. Řešením těchto konfliktních situací je zavedení systému **priorit (předností)**, které umožňuje procesoru rozhodování o pořadí jednotlivých žádostí.

Mezi nejznámější metody vyhodnocení priority žadatele jsou používány druhy *samostatných žádostí, zřetězení* a *cyklických výzev*.

6.1 Samostatné žádosti

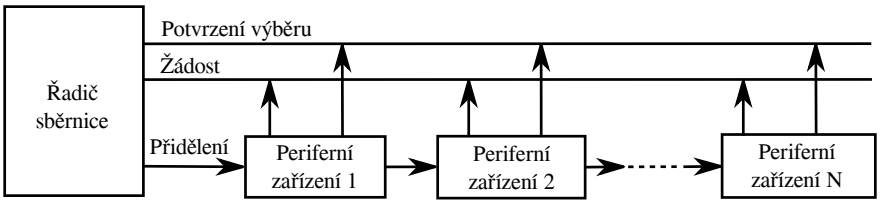
Samostatné žádosti jednotek se posílají přímo po radiálních vodičích do procesoru. Ke každému vodiči „žádost“ je přiřazen jeden vodič s označením „přidělení“, probíhající radiálně k žádající jednotce, viz. blokové schéma této metody na obrázku 13.

Tímto jednoznačným přiřazením jednoho vodiče každé jednotce je dána informace procesoru, která jednotka žádala o sběrnici. Vyhodnocení priority se může provést jednoduchým prioritním dekodérem. Jednotlivé priority jsou přiděleny pevně, což umožňuje kombinačnímu obvodu zpracování přicházejících žádostí s minimálním zpožděním. Další možností je vyhodnotit přijaté žádosti programem, který cyklicky prohledává registr, a v kterém jsou tyto žádosti zapsány. Výhodou použití programu je možnost jeho snadné změny.

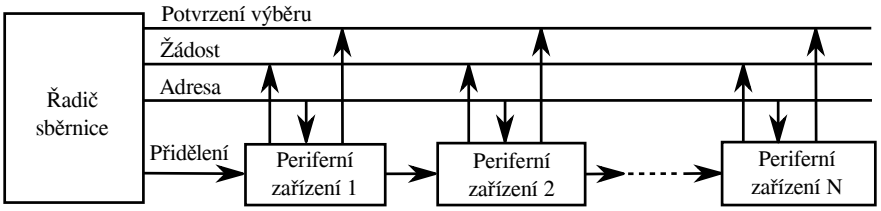
6.2 Zřetězení

Požadavky na obsluhu zařízení vysílají všechny jednotky po jediném vodiči, takže procesor dostává žádost anonymně a sám ji nemůže vyhodnotit. Tyto

[5] např. žádost o přerušení, žádost o DMA, žádost o sběrnici apod.



Obrázek 14: Vyhodnocení priority zřetězením



Obrázek 15: Vyhodnocení priority cyklickými výzvami

požadavky jsou opět logicky sečteny a signál potvrzení žádosti od mikroprocesoru je sériově veden přes všechna periferní zařízení. Pořadí, v němž jsou jednotky tímto vodičem propojeny, odpovídá pořadí priorit (první jednotka, následující přímo za procesorem, má nejvyšší prioritu, poslední jednotka má nejnižší prioritu).

Po příchodu přidělovacího signálu záleží, zda jednotka žádala o přidělovací signál. Pokud žádala, zablokuje přidělovací signál a nepustí jej dál a současně potvrdí výběr. Potvrzení výběru učiní vygenerováním své adresy, kterou vyšle do procesoru. Druhou možností je vyslání jednobitového signálu po zvláštním vodiči, kterým je každá jednotka spojena s procesorem. Pokud je sběrnice přidělena určité jednotce, pak všechny jednotky za ní následující musí čekat. Jednotka, která nežádala o přidělení, propustí přidělovací signál beze změny a nijak na něj nereaguje.

6.3 Cyklické výzvy

Třetí metodou vyhodnocování priorit jsou *cyklické výzvy (polling)*. Jednotky vysílají své žádosti opět po jednom vodiči sběrnicového typu, takže procesor přijme žádost jako anonymní, viz obrázek 15. Namísto jednoho vodiče, který signalizuje přidělení sběrnice, je však použita skupina vodičů „adresa jednotky“, schopná přenášet adresu kterékoliv z připojených jednotek [6]. Když procesor přijme žádost o přidělení sběrnice (a je-li připraven této žádosti vyhovět), začne po adresových vodičích vysílat postupně adresy všech jed-

[6] pro tento účel lze využít též část adresové sběrnice

notek v předem stanoveném pořadí. Jakmile jednotka, která žádala o přidělení, rozpozná na adresových vodičích svou adresu, reaguje na ni jako signál „přiděleno“, vyšle do procesoru signál „potvrzení výběru“ a stane se řídící jednotkou.

7 Historie sběrnic PC

Typ	rok vzniku	Datová šířka [bitů]	takt [MHz]	Přenosová rychlost [MB/s]
XT bus	1987	8	4,77	až 2,38
ISA	1984	16	7,15 či 8,33	až 16,6
MCA	1987	32	10,22	40
EISA	1988	32	8,33	až 33
VESA Local BUS	1992	32	25–66	až 264
Opti Local BUS	1992	32	25–66	až 264
PCI	1993	32/64	20–66	132 až 528
AGP	1992	32	66/133	264–2112
PCI-X	2002	64	66/133	533–1066
PCI-Express	2004	1–128	250	až 32000

Tabulka 1: Přehled vývoje sběrnic PC

8 Kontrolní otázky

1. Z jakých částí se skládá sběrnice a co je účelem jednotlivých částí?
2. Co to je adresní dekodér a kdy je potřeba jej použít?
3. Jaký je princip komunikace s perifériemi pomocí V/V bran?
4. K čemu slouží u komunikace V/V bran indikátor a jaké přináší výhody?
5. Popište, jak probíhá přenos dat pomocí V/V brány s indikátorem.
6. Jaký je rozdíl mezi programově řízenou komunikací s perifériemi a pomocí přerušení?
7. Jaké výhody přináší řízení komunikace s využitím přerušení?
8. Z jakých částí se skládá řadič DMA?
9. Jak probíhá přenos dat s použitím DMA?
10. Jaké má výhody řadič DMA proti přenosu dat s využitím CPU?