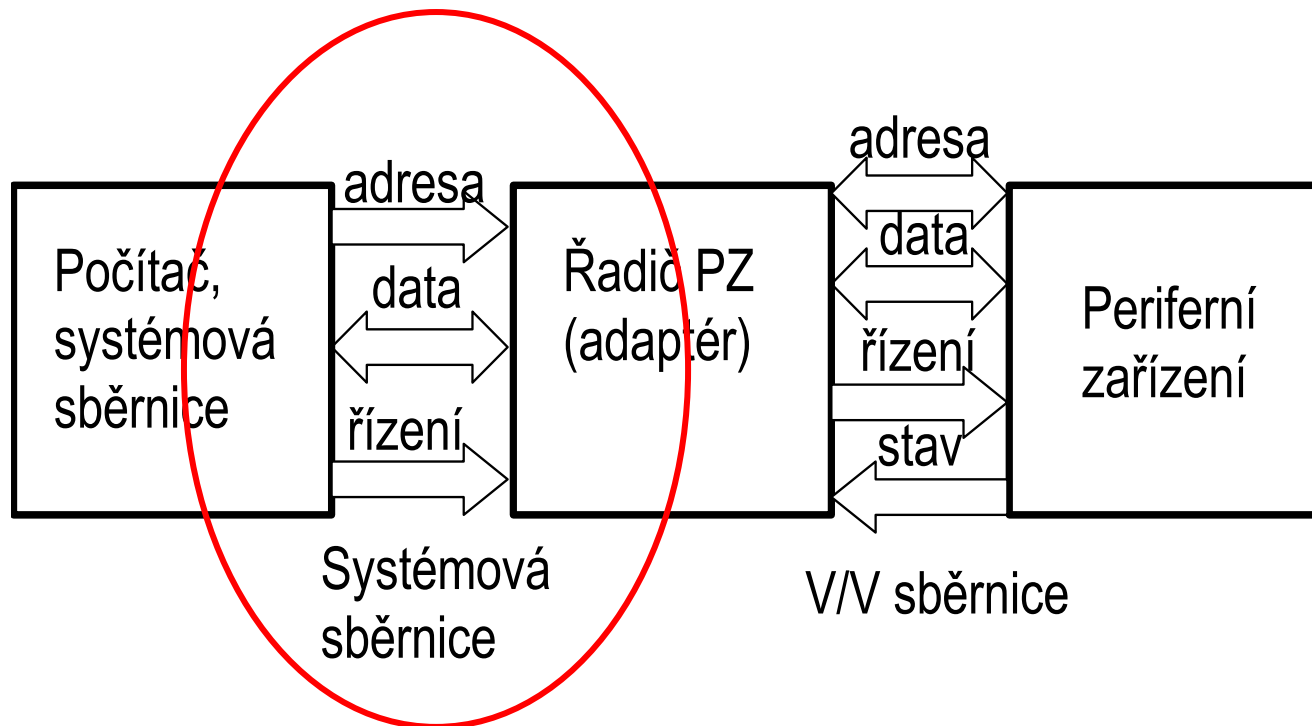


# Principy komunikace s řadiči/adaptéry periferních zařízení

## **Osnova přednášky:**

- Vysvětlit činnosti, které se odehrávají na straně procesoru při vyvolání periferní operace.
- Vysvětlit principy konstrukce obvodů na straně řadiče PZ, které na tyto činnosti reagují.
- Vysvětlit pojem universální obvody řadiče (obvody reagující na činnost odehrávající se na sběrnici) – mohou být součástí každého řadiče PZ (či jiné komponenty komunikující přes systémovou sběrnici).
- Vysvětlit principy programové obsluhy periferní operace.
- Vysvětlit souvislosti mezi činnostmi realizovanými instrukcemi a potřebným obvodovým vybavením.
- Zobecnění těchto principů.



- **Výchozí fakta:**

Komunikace s registry zařízení (např. řadiče PZ) umístěného v systémové sběrnici (obecně „klienti systémové sběrnice“):

1. na základě realizace instrukce assembleru,
2. na základě činnosti řadiče DMA,
3. na základě činnosti jiného prvku v sestavě počítače, který umí řídit sběrnici (bus master).

Alternativa 1)

- **Na straně počítače instrukce vyvolá:**

vložení dat na datovou část systémové sběrnice,

vložení adresy registru na adresovou část systémové sběrnice,

generování řídicího signálu sběrnice typu „zápis do registru“.

Podobně je to v případě instrukce „čti obsah registru“.

- Na straně řadiče PZ instrukce vyvolá (přes systémovou sběrnici):

Akci, jejímž cílem je rozpoznání, zda adresa na adresové části systémové sběrnice je adresou řadiče PZ a dále některého z vnitřních registrů řadiče PZ (dekódování adresy).

Pokud ano, pak řadič provede **zápis obsahu datové části systémové sběrnice do adresovaného registru**.

Výkonný signál uvnitř řadiče: **odvozený (dekódovaný) z adresy a signálu sběrnice „zápis do registru“**, výkonný signál je veden na synchronizační vstup registru.

Důležitý princip: **pouze jeden registr v celém počítači je rozpoznán jako adresovaný registr** a pouze do něj se vloží obsah datové části systémové sběrnice.

Výsledek: **pouze jediný registr v celém počítači dostane výkonný signál (synchronizaci), pouze do něj se zapíše informace přenášená přes datovou část systémové sběrnice**.

## Kategorizace principů komunikace s adaptéry periferních zařízení

- Několik možností kategorizace principů komunikace s externími adaptéry, např.:
  1. Podle způsobu adresace registrů, které jsou součástí adaptérů.
  2. Podle způsobu řízení a obsluhy periferní operace.
- Kategorizace podle bodu 2) je z hlediska principů řízení periferních operací důležitější.

### Principy adresace registrů

- **1. princip:** izolované vstupy/výstupy (registry mají svůj vlastní adresový prostor).
- **2. princip:** adresy registrů jsou mapovány do adresového prostoru operační paměti (sdílejí adresový prostor s prvky operační paměti).

- **Rozlišení obou principů:** na úrovni assembleru.

Isolované vstupy/výstupy:

Jsou k dispozici pouze 2 skupiny instrukcí - IN, OUT.

Vstupy/výstupy mapované do adresového prostoru operační paměti:

Je používán podstatně širší repertoár instrukcí (stejná sada instrukcí jako pro práci s pamětí).

- **Rozlišení obou způsobů adresace na úrovni obvodové realizace:**

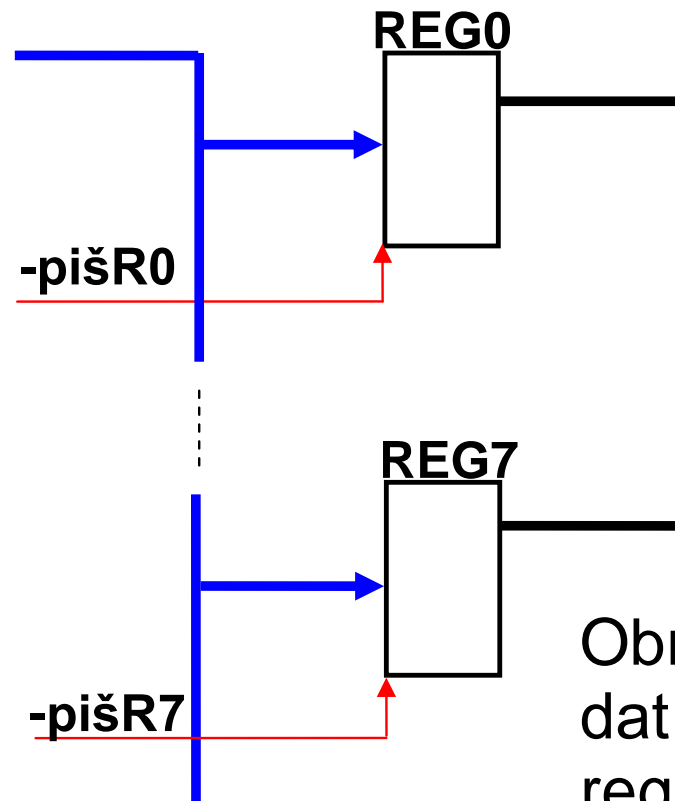
Je zajištěno řídícími signály generovanými procesorem (řadičem sběrnice) do systémové sběrnice.

Mikroprocesory nebo řadiče sběrnice generují dvě skupiny signálů odvozené od typu a kódu instrukce:

**čtení obsahu paměti, zápis do paměti** – práce s pamětí

**čtení obsahu registru, zápis do registru** – práce s registry

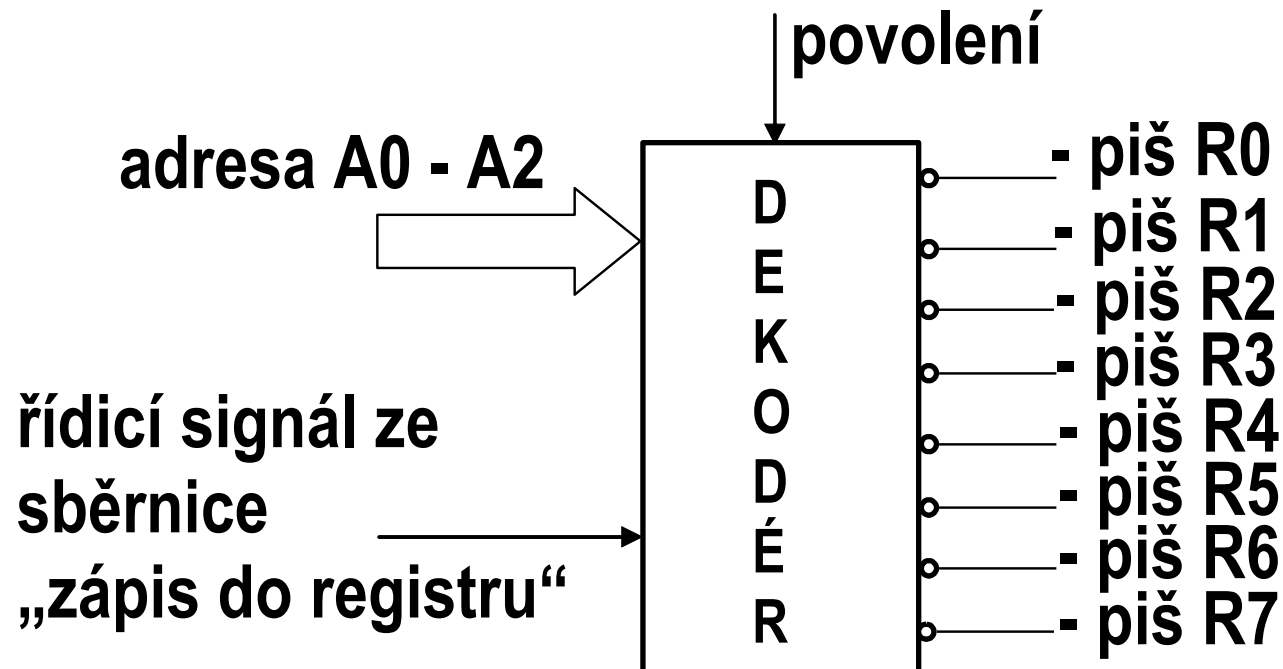
Sběrnice PCI – rozlišeno příkazy.



Obr. 1 Technika zápisu dat z datové sběrnice do registru

- Vnitřní datová sběrnice klienta sběrnice (řadiče PZ) je přivedena na datové vstupy všech registrů, pouze pro jediný z nich je generován synchronizační signál „piš registr“.

## Isolované vstupy/výstupy - zápis do registru



Obr. 2 Dekódování signálů pro zápis do adresovaného registru (isolované vstupy)

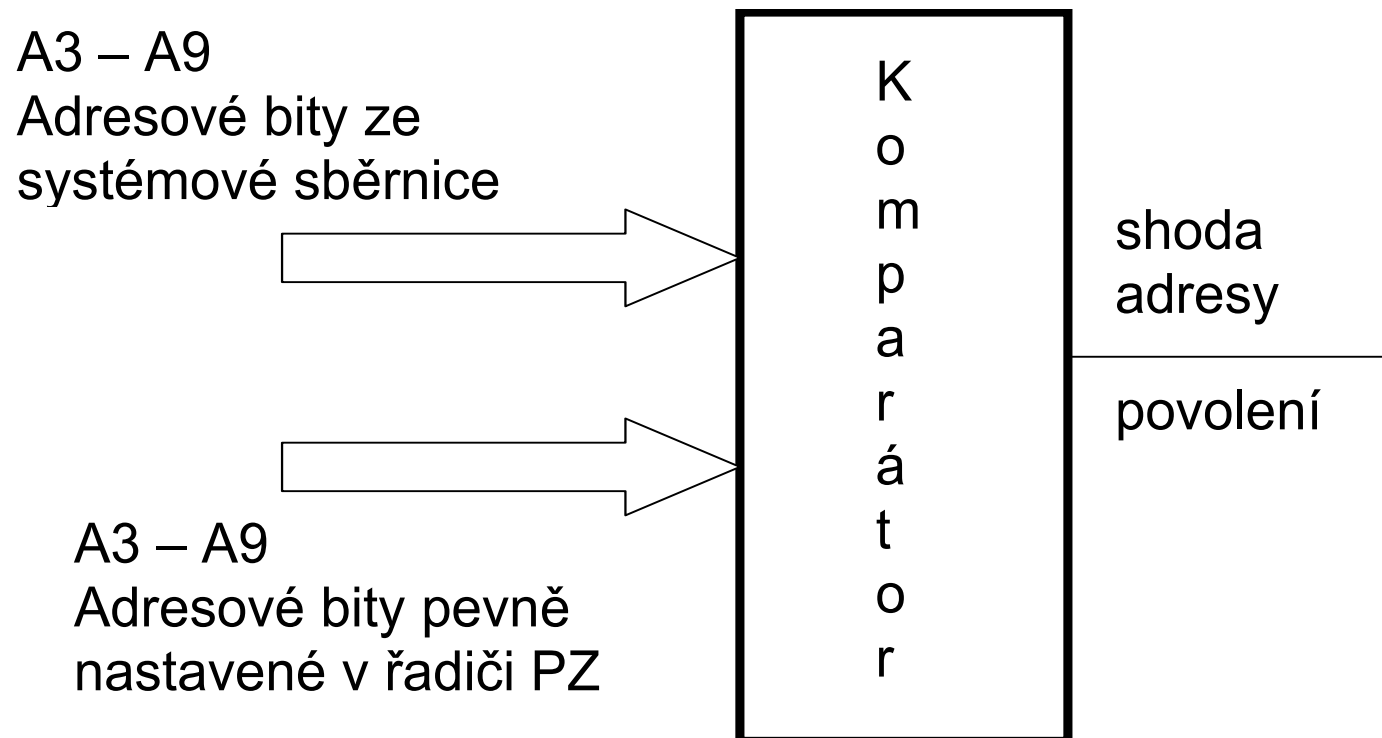
- Signál „zápis do registru“ systémové sběrnice je odvozen od instrukce **OUT** (tzn. instrukce pro práci s registry), je generován buď procesorem nebo řadičem sběrnice nebo prvkem typu bus master.



- Signály - **piš R0** až - **piš R7** jsou vedeny na synchronizační vstupy registrů, na jejichž datový vstup (např. 8 bitový) je přivedena datová sběrnice.
- Dekodéry mají ještě vstup **CS** (Chip Select), jímž je podmíněno aktivování dekodéru. Odvozuje se od vyšších adresových bitů – „**povolení**“.

## Princip generování signálu „povolení“

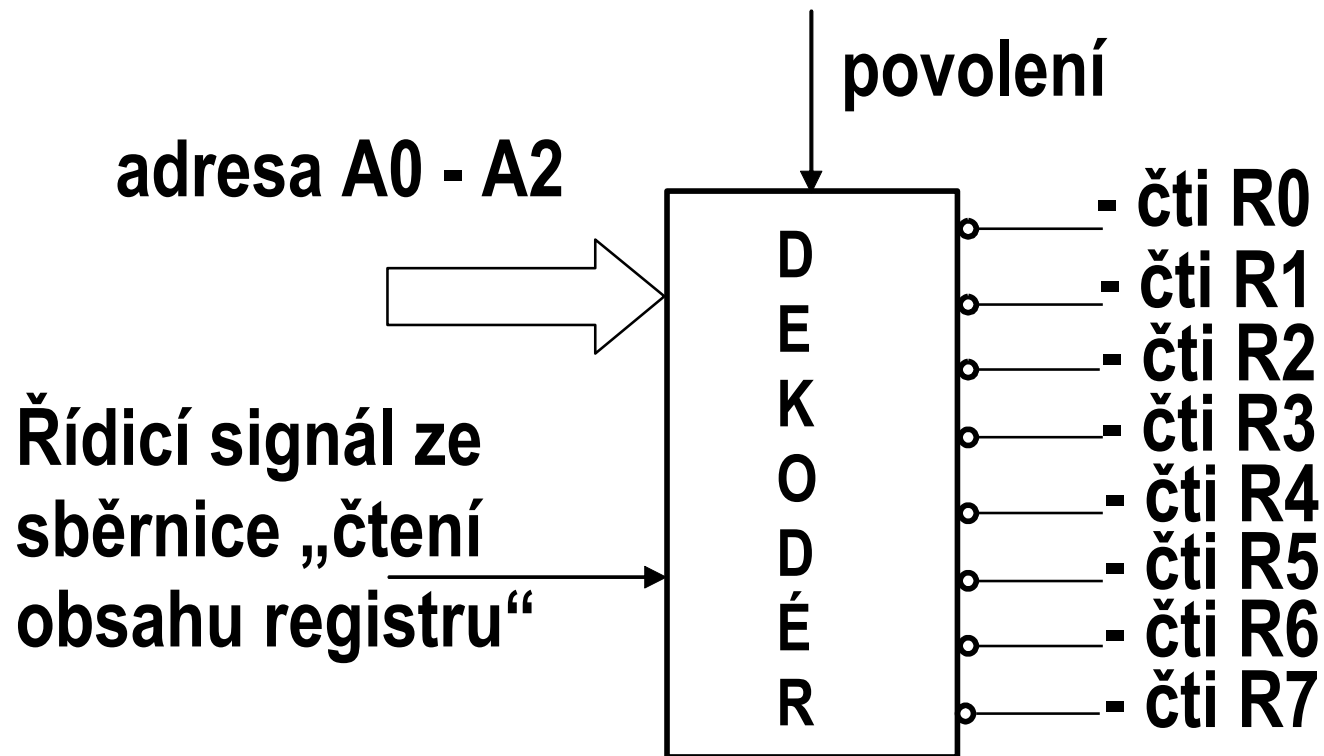
- Adresa registru sestává ze dvou částí: adresy jednotky (v tomto případě řadiče PZ) – např. bity A3 – A9, a adresy registru: např. A0 – A2.



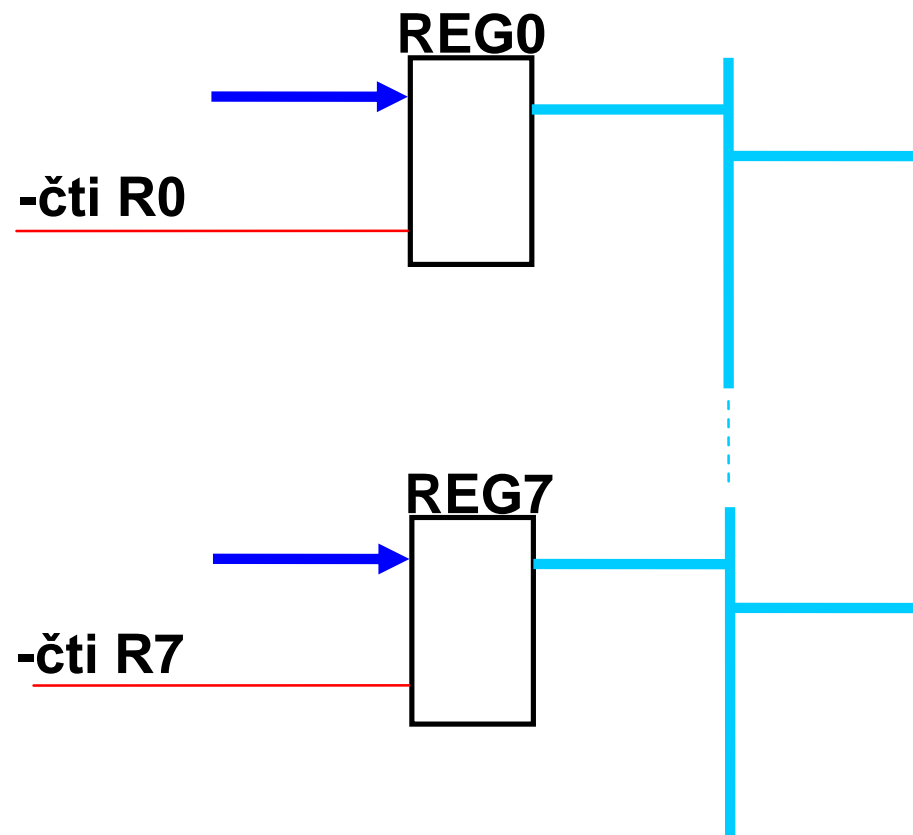
Obr. 3 Generování signálu „povolení“

- Každý prvek (řadič PZ) v systémové sběrnici musí nejprve rozpoznat, že je adresován, pak teprve povolí zápis do některého ze svých vnitřních registrů.
- Výsledek: pokud je shoda obsahu adresových bitů adresové části systémové sběrnice s adresou nastavenou v řadiči PZ – povolí se zápis do registrů řadiče – uvolní se dekodéry registrů v řadiči PZ a signály datové části systémové sběrnice jsou vloženy na vstupy všech registrů „uvnitř“ adresovaného řadiče PZ.
- Shrnutí: při studiu funkcí sběrnice je nutné se zaměřit na principy adresace klientů sběrnice (např. řadičů PZ) a jejich vnitřních prvků (registrů, pamětí).

## Isolované vstupy/výstupy – čtení obsahu registru

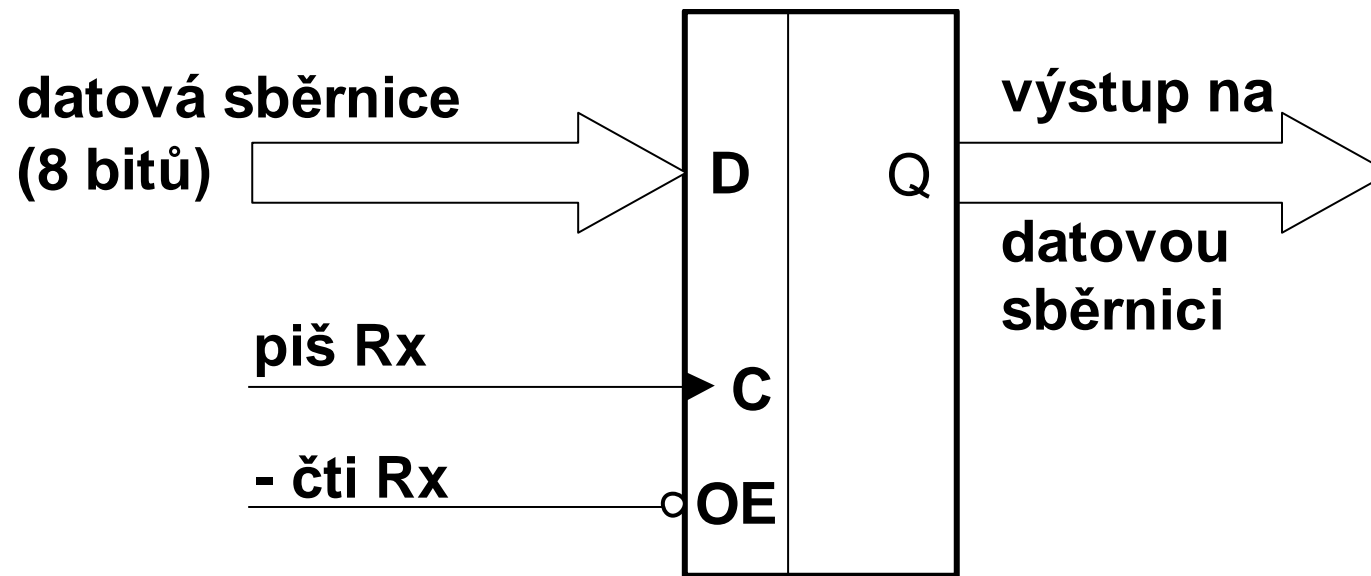


Obr. 4 Dekódování signálů pro čtení obsahu adresovaného registru (isolované vstupy)



Obr. 5 Technika vložení obsahu registru do datové sběrnice

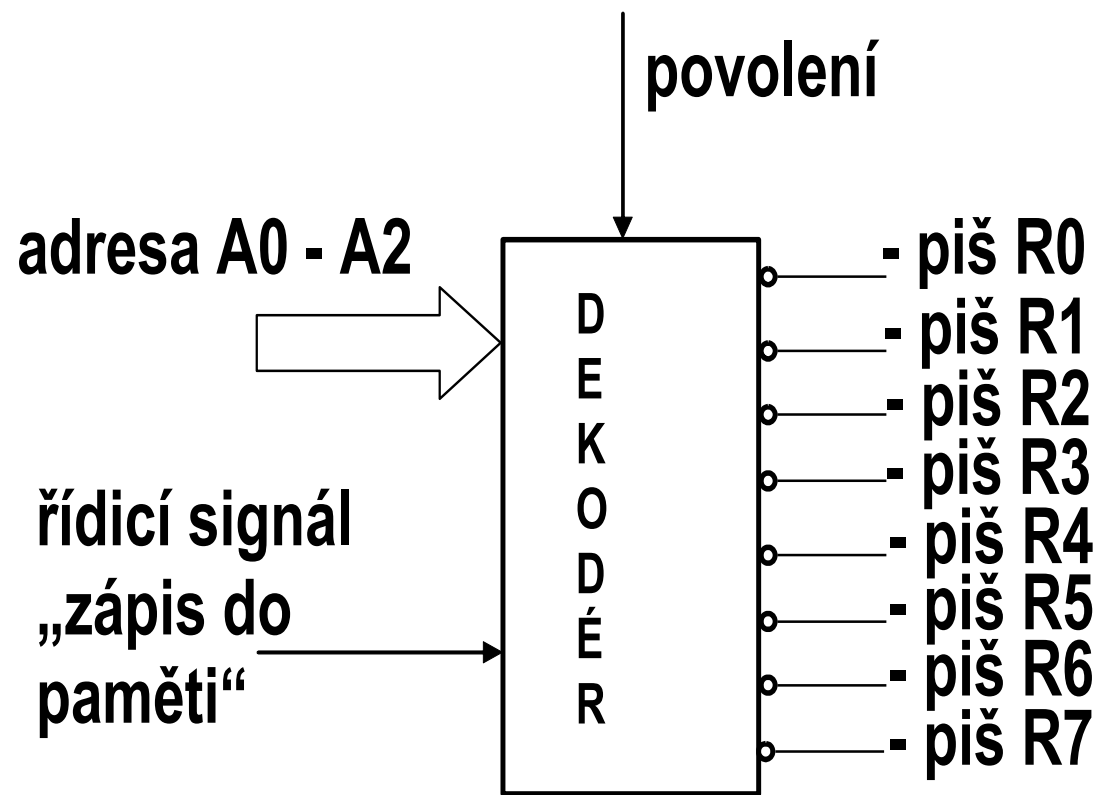
- Signál „**čtení obsahu registru**“ je odvozen od instrukce typu **IN** (tzn. instrukce pro práci s registry), je generován buď procesorem nebo řadičem sběrnice.
- Signály **-čti R0** až **-čti R7** jsou vedeny na uvolňovací vstupy registrů (např. **oe** - Output Enable), čímž se zajistí, že se na datovou sběrnici vloží obsah pouze tohoto adresovaného registru.
- Registry napojené svými výstupy na vnitřní datovou sběrnici řadiče PZ musí mít tzv. **třístavový výstup** (tzn. umějí stav „0“, „1“, „vysoká impedance“). **Jsou-li výstupy ve třetím stavu, pak neovlivňují stav na sběrnici.**
- **V konkrétním časovém okamžiku tak ovlivňuje stav na sběrnici pouze jediný registr ve výpočetním systému.**



Obr. 6 Zápis/čtení do/z registru

- **Důležité: všechny prvky, které potenciálně mohou vstupovat do sběrnice, musí mít třístavové výstupy.**

Registry mapované do adresového prostoru operační paměti - zápis  
do registru

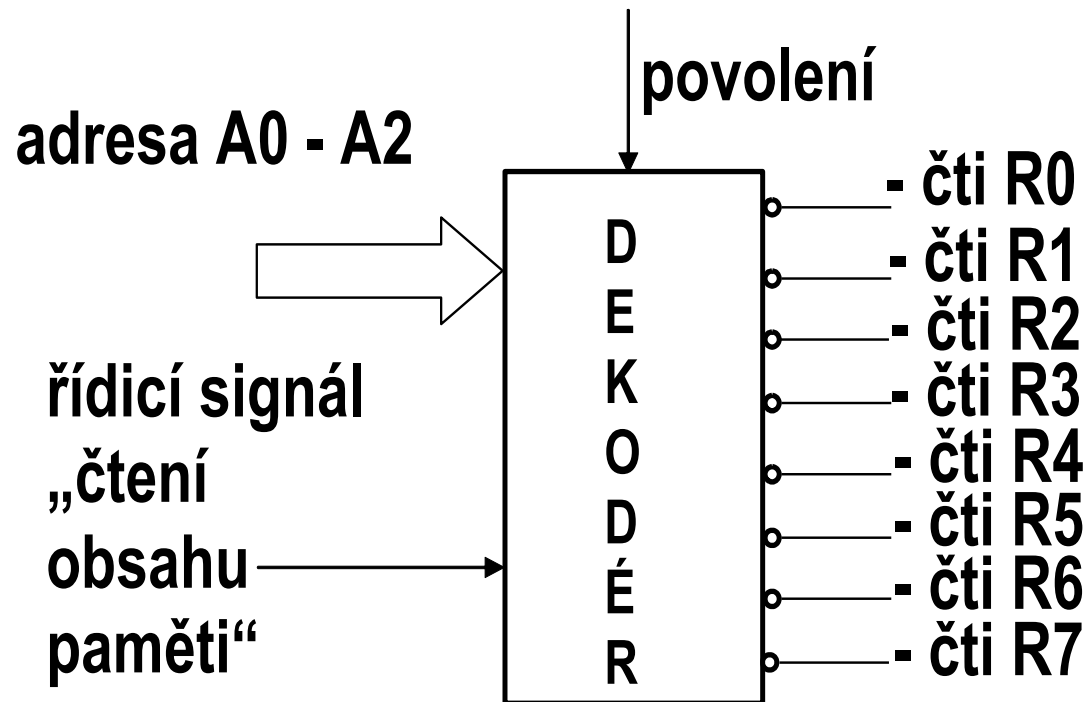


Obr. 7 Dekódování signálů pro zápis do adresovaného registru  
(registry mapovány do adresového prostoru operační paměti)



- Signál “**zápis do paměti**“ je odvozen od instrukce pro práci s pamětí – repertoár těchto instrukcí je širší než repertoár instrukcí pro práci s registry.

Registry mapovány do adresového prostoru operační paměti – čtení obsahu registru

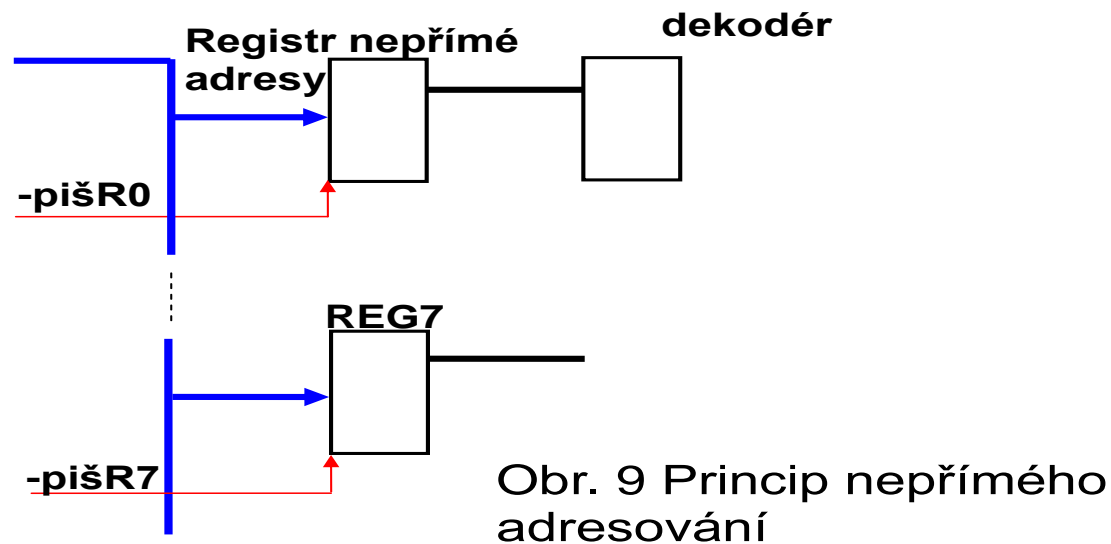


Obr. 8 Dekódování signálů pro čtení obsahu adresovaného registru (registry mapovány do adresového prostoru operační paměti)

- Signál **“čtení obsahu paměti”** je odvozen od instrukce pro práci s pamětí.

## Nepřímá adresace

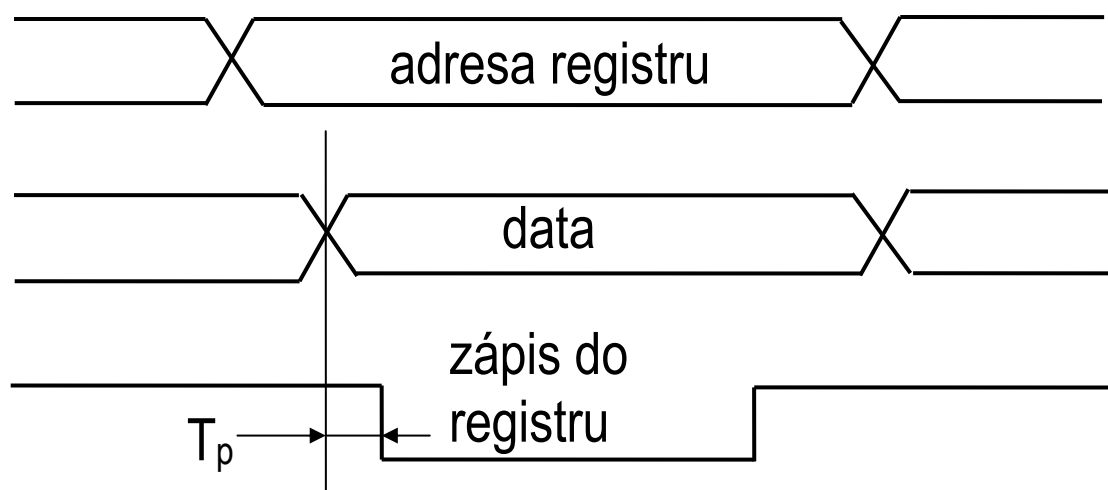
- Registr (nebo jiný prvek – např. paměť) není adresován obsahem adresové části systémové sběrnice ale obsahem registru.
- Adresu je nutné do registru nejprve vložit – instrukcí OUT.



## Časové diagramy komunikace na sběrnici

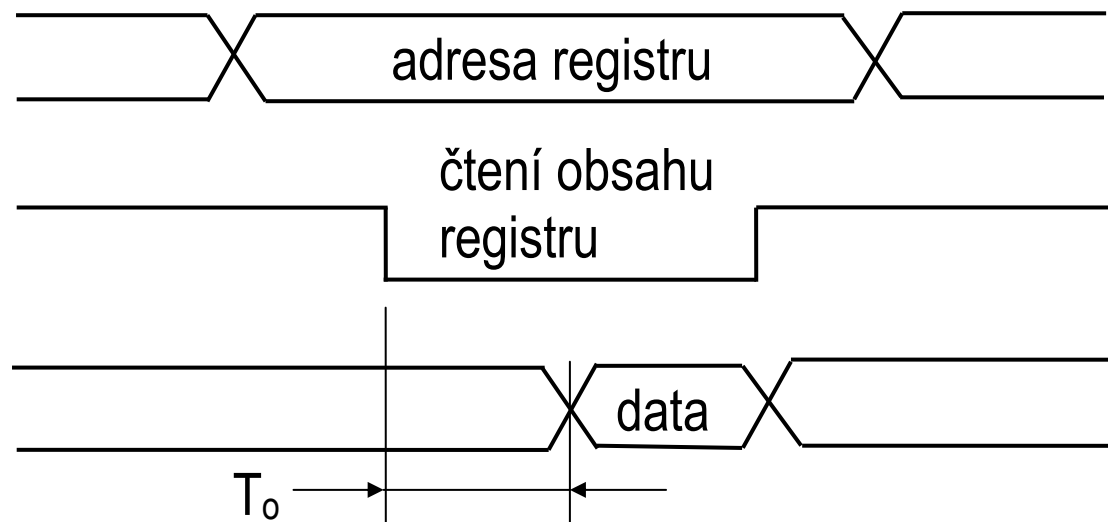
- Popíšeme dva principy:

Jednoduchá komunikace řízená řadičem sběrnice (řadič sběrnice je řízen procesorem jistým počtem bitů odvozených od konkrétní instrukce IN/OUT).



Obr. 10 Časový diagram operace „zápis do registru“

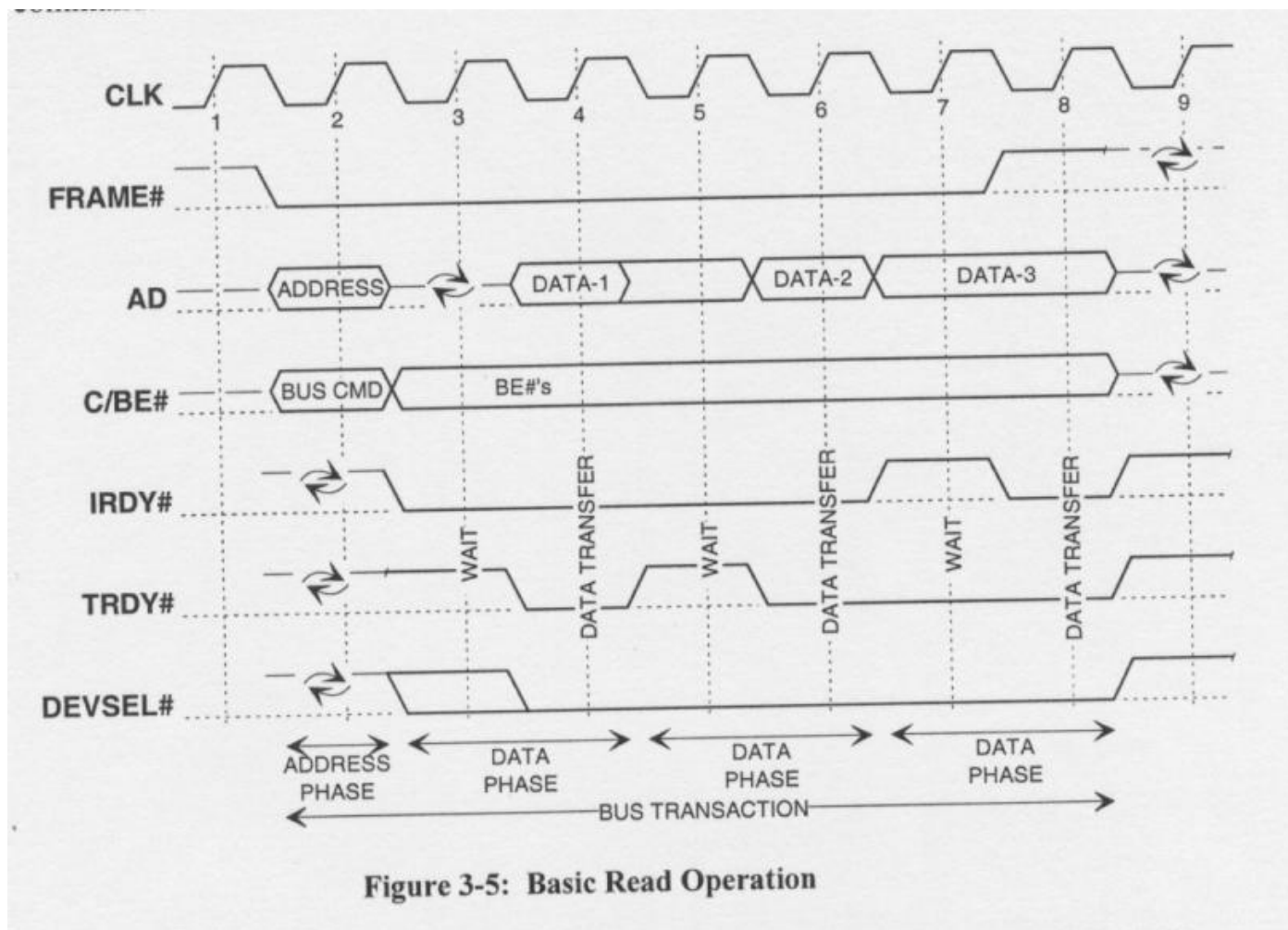
- **Adresa registru** – adresová část systémové sběrnice, adresa je do registru na straně procesoru vložena instrukcí.
- **Data** – datová část systémové sběrnice, data jsou do příslušného registru na straně procesoru vložena instrukcí.
- „Zápis do registru“ – jeden ze signálů řídicí části systémové sběrnice, je generován jako výsledek realizace instrukce.
- Data jsou do registru na přijímací straně vložena nástupnou hranou signálu „zápis do registru“ („registry citlivé na hranu“ v. „registry citlivé na hladinu“).
- Časovou relaci mezi daty a signálem „zápis do registru“ zařídí řadič sběrnice.
- $T_p$  – předstih dat na sběrnici před výkonným signálem pro zápis.
- Není žádná další komunikace kromě zápisu dat - výrazně malá režie.



Obr. 11 Časový diagram operace „čtení obsahu registru“

- **Adresa registru** – adresová část systémové sběrnice, adresa je do registru na straně procesoru vložena instrukcí.
- Signál „čtení obsahu registru“ – jeden ze signálů řídicí části systémové sběrnice, generován jako výsledek realizace instrukce, je generován řadičem sběrnice.

- Důležité: **žádná režie**, operaci zápisu/čtení nepředchází žádná úvodní fáze, např. poslání adresy, příkazu, apod.
- Sběrnice PCI – režie existuje (poslání adresy, příkazu), následně však **blokový přenos** dat => režie se redukuje.
- Pokud by se pracovalo ve **slabikovém režimu** – režie by byla velká. Pro každou slabiku by musela proběhnout úvodní fáze.
- $T_0$  – doba, za niž řadič PZ musí být schopen vložit data na datovou část systémové sběrnice.
- Pokud není schopen tento parametr dodržet, pak je nutné vkládat čekací stavy do komunikace (to si vynutí řadič PZ).
- Příčiny tohoto stavu: odlišná technologická úroveň řadiče PZ a procesoru.
- Stav dnes v PC: problém je výrazně eliminován.



Obr. 12 Časový diagram komunikace na sběrnici PCI

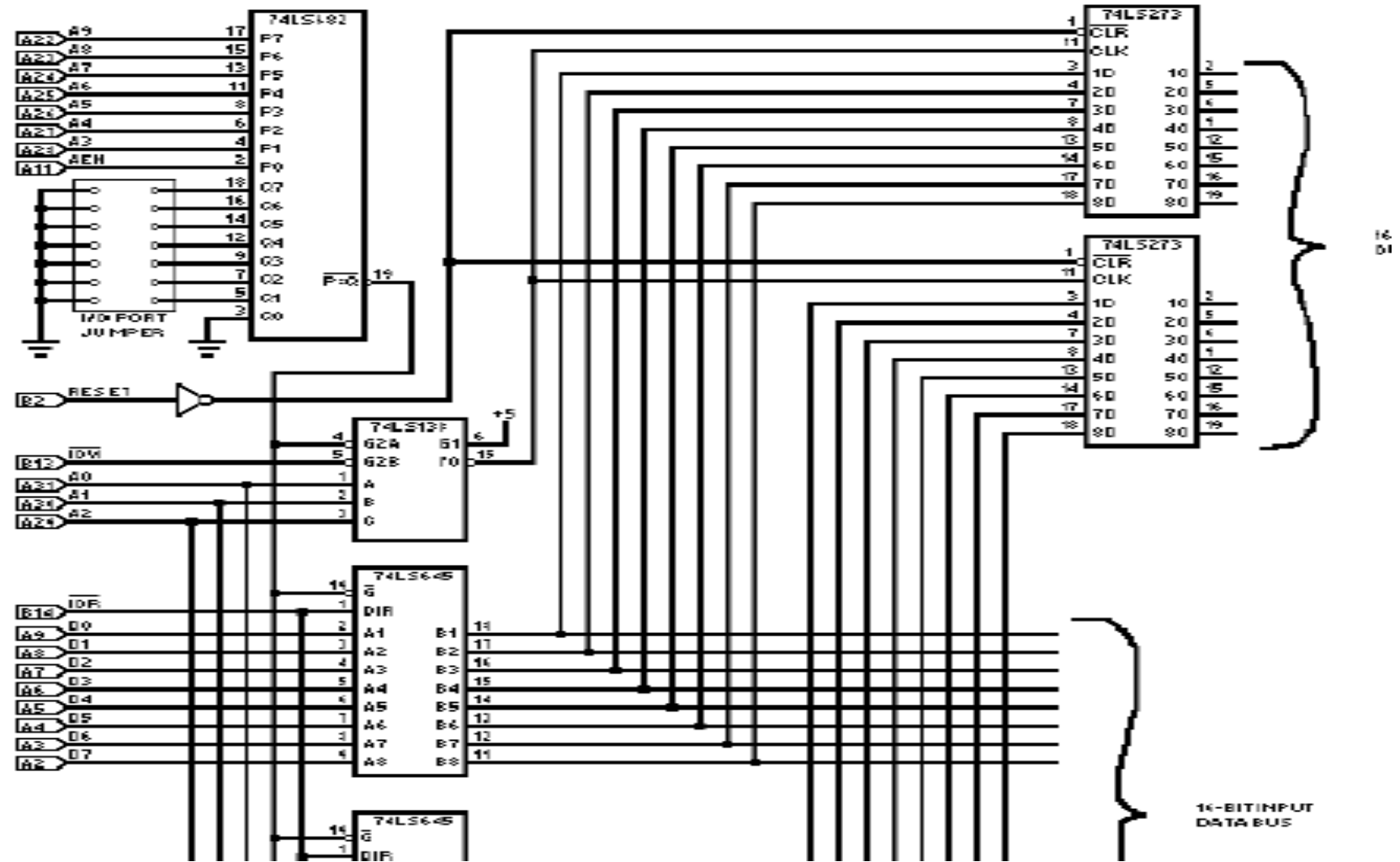


## Způsob využití signálu o shodě adres

- Signál je pouze interním signálem řadiče – takto to bylo v řadičích sběrnice ISA.
- Sběrnice PCI – signál o tom, že došlo ke shodě adres, je navíc signálem sběrnice (signál DEVSEL).

## Shrnutí

- **Isolované vstupy/výstupy**
  - registry mají svůj vlastní adresový prostor (tzn. **existují 2 disjunktní adresové prostory** – paměťový adresový prostor, adresový prostor pro registry),
  - omezený repertoár instrukcí (IN, OUT)
- **Mapování registrů do adresového prostoru operační paměti**
  - široký repertoár instrukcí,
  - adresové prostory nejsou disjunktní (tzn. existuje jediný adresový prostor),
  - adresy registrů zabírají adresový prostor operační paměti.



Obr. 13 Příklad univerzálních obvodů klienta systémové sběrnice

## Komentář k obrázku:

- **Prvek 74LS245** – prvek typu transceiver (budič – snímač, transmitter - receiver), vysílá data na sběrnici, přijímá data ze sběrnice.

Výstup srovnávacího obvodu – pokud adresa souhlasí s adresou nastavenou v řadiči (tzn. s adresou řadiče), pak se transceiver aktivuje (vstup G - gate).

O směru přenosu rozhoduje vstup DIR (direction – směr) – signál ze sběrnice IOR (čtení).

Bits A0 – A2 určují adresu registru (součástí tohoto řadiče může být maximálně 8 registrů).

Další vstup dekodéru 74LS139 – signál IOW (zápis), podle adresy A0 - A2 dostane tento signál pouze jediný registr v řadiči (ale také v celém počítači).

- Popsané obvody můžeme považovat za universální – tzn. mohou být využity v zařízení komunikujícím na sběrnici.

## Principy řízení a obsluhy periferních operací

### Základní pojmy

- Pod pojmem řadič PZ (adaptér) budeme rozumět komponentu, která jedním směrem komunikuje s počítačem přes jeho systémovou sběrnici, na druhou stranu komunikuje přes kabel s periferním zařízením nebo jeho řídicí jednotkou (řadičem).
- Typy systémových sběrnic v personálních počítačích:

sběrnice PC XT,

sběrnice ISA

sběrnice EISA

sběrnice lokální (PCI, VLB)

nyní – sériová sběrnice PCI Express

Pro každou sběrnici je typická její **mechanická a elektrická konstrukce** (rychlost v MB/s související se synchronizačním kmitočtem), počty přenášených bitů v paralelních sběrnicích – 8, 16, 32, 64, a další parametry.

- **Co musí řadič PZ umět:**

rozpoznat, že je adresován některý z jeho vnitřních prvků – např. registrů,  
zápis do registru podle zadané adresy,  
čtení obsahu registru podle zadané adresy,  
zápis do paměti podle zadané adresy,  
čtení obsahu paměti podle zadané adresy,  
komunikovat s PZ,  
komunikovat s počítačem (systémovou deskou) přes systémovou sběrnici (např. generovat žádost o přerušení, žádost o přenos DMA).

### Techniky řízení vstupu/výstupu dat

- Hovoříme o těchto 3 principech:
  1. programové řízení vstupu/výstupu dat,
  2. vstup/výstup využívající přerušení,
  3. vstup/výstup s využitím DMA (přímý přístup do paměti).

- Všechny tyto techniky jsou odlišné => vyžadují zcela odlišnou **hardwarovou i softwarovou podporu** jak na straně počítače, tak na straně řadiče PZ.
- **Důsledek**: při každé z těchto technik se dosahuje různých **hodnot parametrů** periferních operací (především rychlosti).  
Další možný parametr: úroveň **autonomního provádění periferní operace** (tzn. bez pozornosti procesoru).

## Programové řízení vstupu/výstupu dat

- Princip činnosti při vstupní / výstupní operaci:

Adaptér: v řadiči PZ je registr, do něhož se z PZ **přeneso slovo dat** a nastaví se bit "**konec operace**" ve stavovém registru.

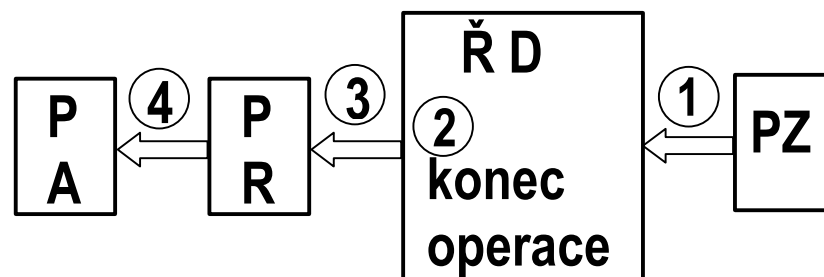
Obdobně při opačném směru přenosu.

Procesor:

Průběžně **testuje obsah stavového registru**, konkrétně bitu "**konec operace**" všech klientů systémové sběrnice – **polling**.

Jakmile zjistí, že je nastaven, **přeneso obsah datového registru** z řadiče PZ do některého univerzálního registru procesoru a odsud do paměti.

Obdobně při opačném směru přenosu.



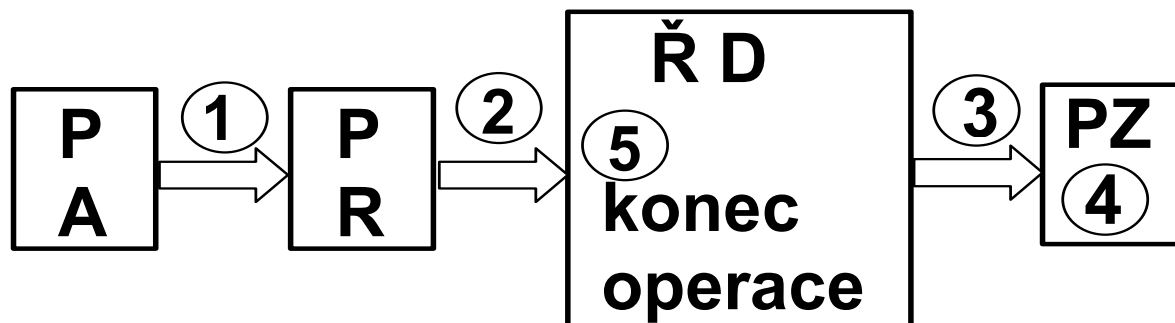
**PA - paměť**  
**PR - procesor**  
**ŘD - řadič**  
**PZ - periferní zařízení**

Obr. 14 Posloupnost činností při přenosu slova dat z PZ do paměti:



## Popis činností:

1. přenos slova z PZ do řadiče PZ,
2. nastavení bitu "konec operace" ve stavovém registru v řadiči PZ  
v průběhu bodů 1. – 2. - průběžné testování obsahu stavového registru procesorem,
3. přenos slova dat z registru řadiče PZ do registru procesoru,
4. přenos slova dat do paměti.



**PA - paměť**  
**PR - procesor**  
**ŘD – řadič**  
**PZ - periferní zařízení**

Obr. 15

Posloupnost činností při přenosu slova dat z paměti do PZ:

1. přenos slova dat z paměti do universálního registru procesoru,
  2. přenos slova dat do vstupního datového registru řadiče PZ,
  3. přenos slova dat do PZ,
  4. zpracování slova dat v PZ (např. tiskárna - vložení slova dat do vyrovnávací paměti tiskárny),
  5. nastavení bitu "**konec operace**" ve stavovém registru,
- v průběhu kroků 4) - 5) testování stavového registru, konkrétně bitu "konec operace".
- V takto koncipovaném řadiči PZ musí být tedy přítomen kl. obvod "**konec operace**", který musí splňovat tyto podmínky:
    1. musí být možné jej ve správných okamžicích hardwarově nastavovat ze strany PZ (tak, aby indikoval ukončení periferní operace),
    2. musí být k dispozici technika na zjišťování jeho stavu,

3. ze strany procesoru musí být možné jej před zahájením periferní operace programově nulovat (aby byl připraven pro nastavení periferním zařízením pro indikaci ukončení dalšího přenosu).

- **Příklad:** minipočítač řady **ADT** (kompatibilní s minipočítači Hewlett Packard).

Architektura:

- Velmi podobná s architekturou dnešních PC – deska s procesorem + rošt s externími adaptéry, na něž byla napojena periferní zařízení.
- Na desce řadiče PZ byl klopný obvod „**flag**“ pro indikaci konce periferní operace, který bylo možné **programově nulovat ze strany procesoru, hardwarově nastavovat (ze strany PZ) a programově testovat ze strany procesoru** (instrukcí podmíněného skoku).

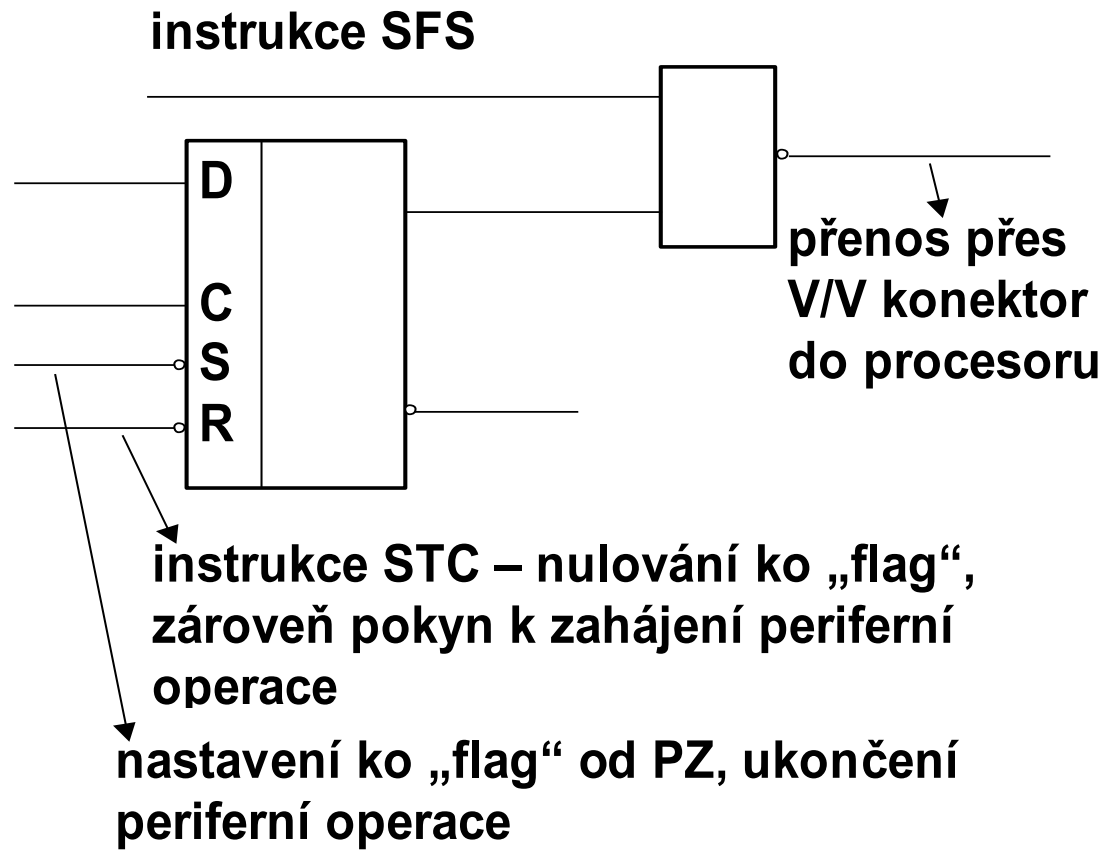
Posloupnost instrukcí pro komunikaci procesoru s řadičem PZ –  
**výstup dat:**

1. **OTA** - přenos slova dat z registru procesoru do registru řadiče PZ.
2. **STC (Set Control)** – start periferní operace, nulování ko „**flag**“.
3. **SFS (Skip if Flag Set)** – instrukce podmíněného skoku, testovanou podmínkou je stav ko „**flag**“, přenos stavu ko „**flag**“ do procesoru.
4. **JMP \* - 1**
5. **Pokračování programu**

Posloupnost instrukcí pro komunikaci procesoru s řadičem PZ –  
**vstup dat:**

Obdoba předcházející posloupnosti – rozdíl je v pořadí instrukcí

1. **STC** – start periferní operace, nulování ko „flag“
2. **SFS (Skip if Flag Set)** – instrukce podmíněného skoku, testovanou podmínkou je stav ko „flag“, přenos stavu ko „flag“ do procesoru.
3. **JMP \*-1**
4. **INA** – přenos obsahu vstupního registru do střadače
5. **Pokračování programu**



Obr. 16 Manipulace s bitem "konec operace"

- Popis obrázku:
  - instrukcí **STC** je vynulován ko „**flag**“ (předtím byl do výstupního registru vložen kód znaku),
  - PZ po převzetí znaku nastaví ko „**flag**“ na úroveň H (indikace konce periferní operace – převzetí znaku)
  - testování stavu ko „**flag**“ instrukcí **SFS** ze strany procesoru (instrukcí podmíněného skoku je možné testovat stav ko, který je součástí adaptéru – signály V/V sběrnice to musí umožňovat).

Shrnutí (techniky zjišťování stavu bitu „konec operace“ v řadiči PZ):

- Technika PC – zjišťování stavu řadiče a PZ:

Bity informující o stavu **jsou součástí registru řadiče** (včetně bitu „konec operace“), registr **je adresovatelný**, je možné jeho obsah instrukcí assembleru **přenést** přes systémovou sběrnici do universálního registru procesoru a tam jej **testovat**.



- Jiná možná technika:

Součástí řadiče je bit „konec operace“, ten je instrukcí assembleru (podmíněný skok) dostupný (testovatelný), aniž by jej bylo nutné přenést do procesoru – méně náročné na počet instrukcí, více náročné na konstrukci systémové sběrnice.

## **Instrukce IN/OUT**

- Vstup slabiky:

### **IN AL,DX**

DX je registr, v němž je uložena adresa vstupního registru.

Výsledek: z takto adresovaného vstupního (datového) registru se přenese jedna slabika (8 bitů) do registru procesoru.

- Vstup slova:

### **IN AX,DX**

DX je registr, v němž je uložena adresa vstupního (datového) registru.

Výsledek: z takto adresovaného vstupního registru se přenese slovo (16 bitů) do registru procesoru.

- Výstup slabiky:

### **OUT DX,AL**

DX je registr, v němž je uložena adresa výstupního (datového) registru.

Výsledek: z registru AL se pošle jedna slabika (8 bitů) do výstupního registru, jehož adresa je uložena v registru DX.

- Výstup slova

## **OUT DX,AX**

DX je registr, v němž je uložena adresa výstupního registru.

Výsledek: z registru AL se pošle jedno slovo (16 bitů) do výstupního registru, jehož adresa je uložena v registru DX.