Principy komunikace s adaptéry periferních zařízení (PZ)

- 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**: isolované 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:

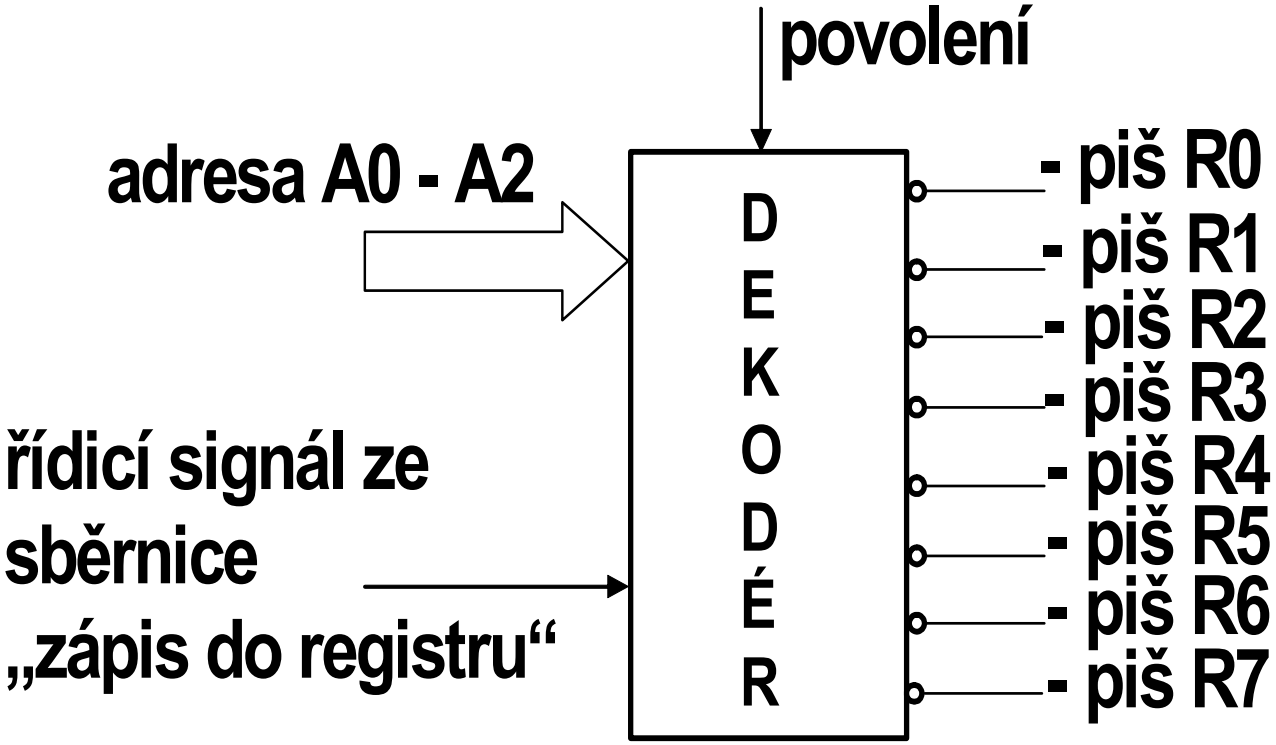
  Je používán podstatně širší repertoár instrukcí (stejná sada instrukcí jako pro práci s pamětí).

- **Rozlišení obou způsobů adresace na úrovni obvodové realizace** je zajištěno řídicími signály generovanými procesorem.

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

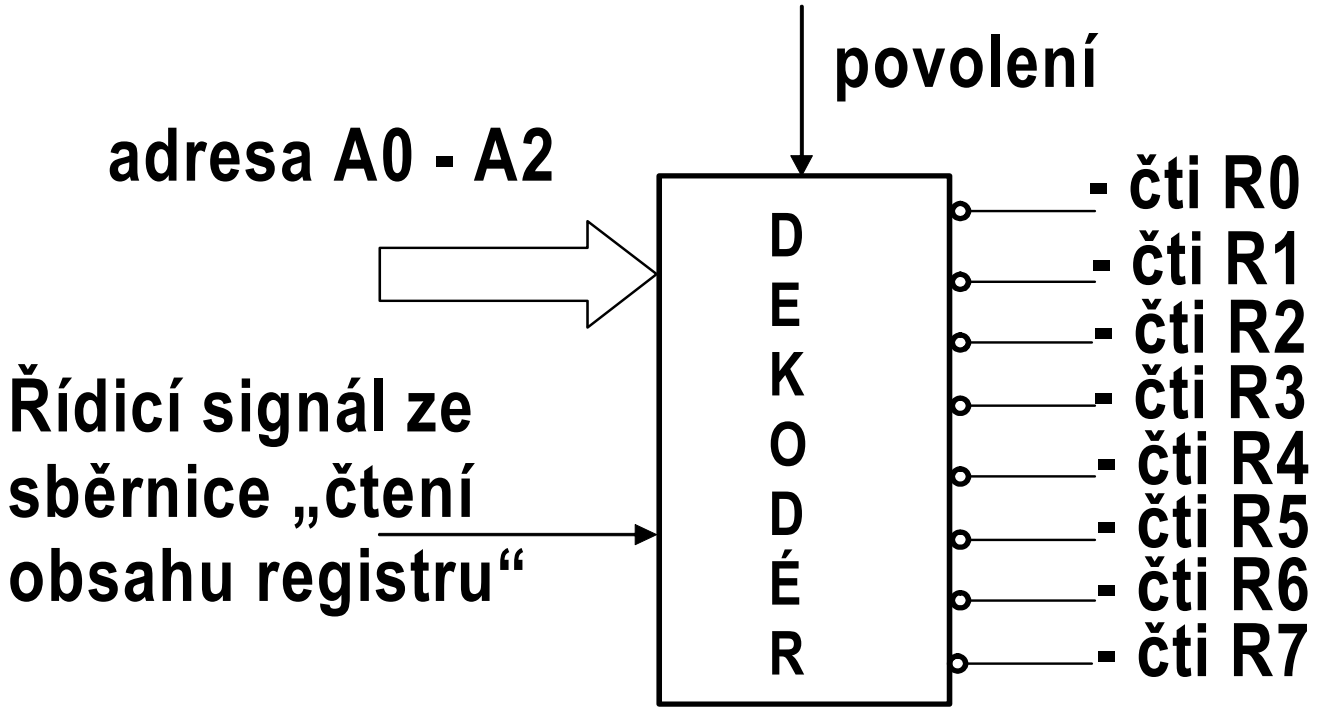
  čtení obsahu paměti, zápis do paměti – práce s pamětí

  čtení obsahu registru, zápis do registru – práce s registry



Obr. 1 Dekódování signálů pro zápis do adresovaného registru (isolované vstupy)

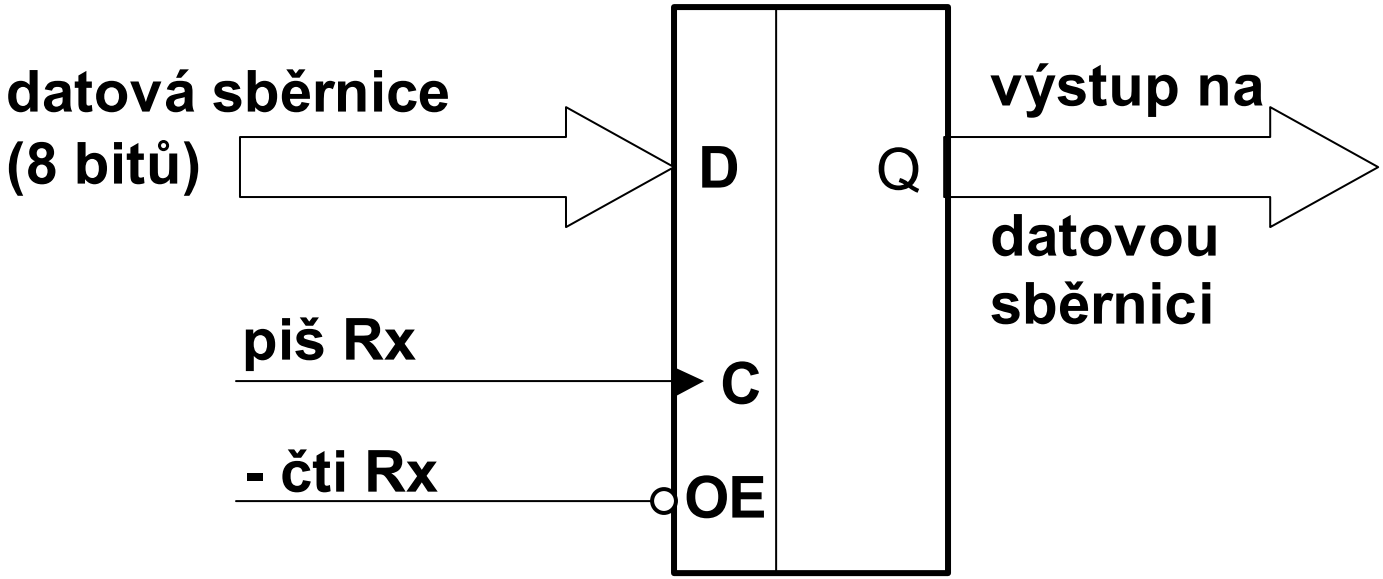
- Signál **„zápis do registru“** je odvozen od instrukce **OUT** (tzn. instrukce pro práci s registry), je generován buď procesorem nebo řadičem sběrnice.
- Signály **- piš R0** až **- piš R7** jsou vedeny na synchronizační vstupy registrů, na jejichž datový vstup (např. 8 bitový) je přivedena datová sběrnice.
- Dekodéry mají ještě vstup **CS** (Chip Select), jímž je podmíněno aktivování dekodéru. Odvozuje se od vyšších adresových bitů – „**povolení**“.



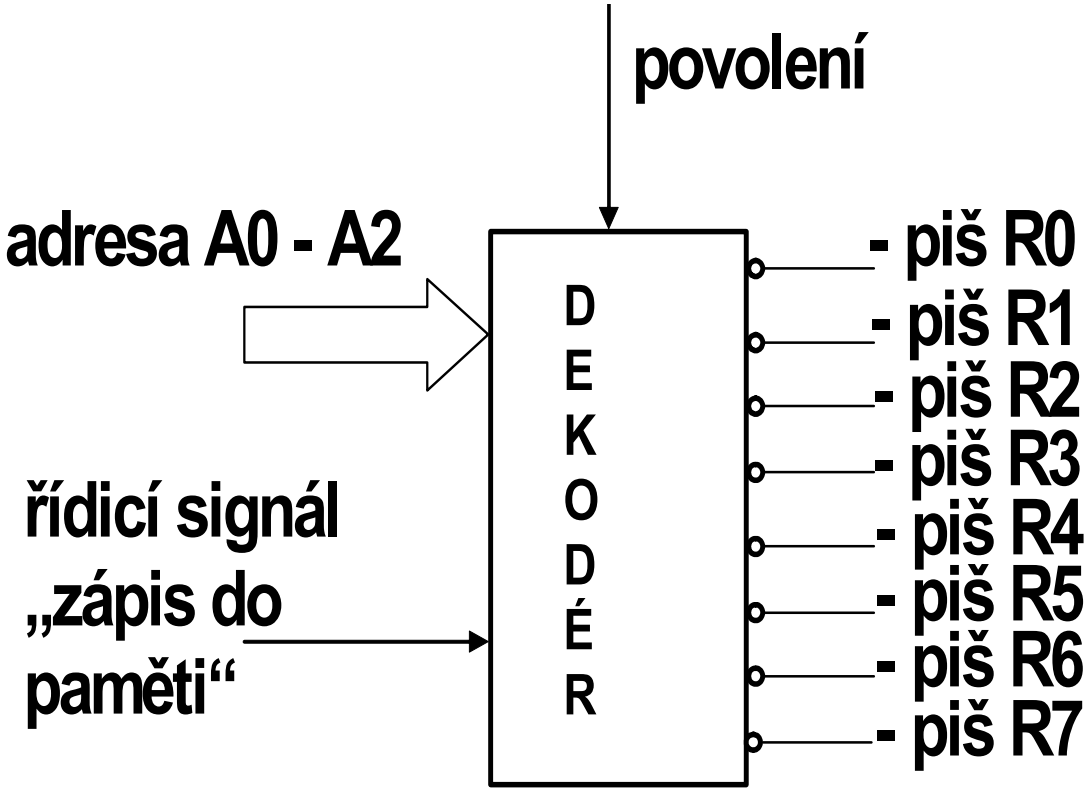
Obr. 2 Dekódování signálů pro čtení obsahu adresovaného registru (isolované vstupy)

- 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 společnou datovou sběrnici, musí mít tzv. **třístavový výstup**. Jsou-li výstupy ve třetím stavu, pak neovlivňují stav na sběrnici.
- V konkrétním časovém okamžiku ovlivňuje stav na sběrnici pouze jediný registr.

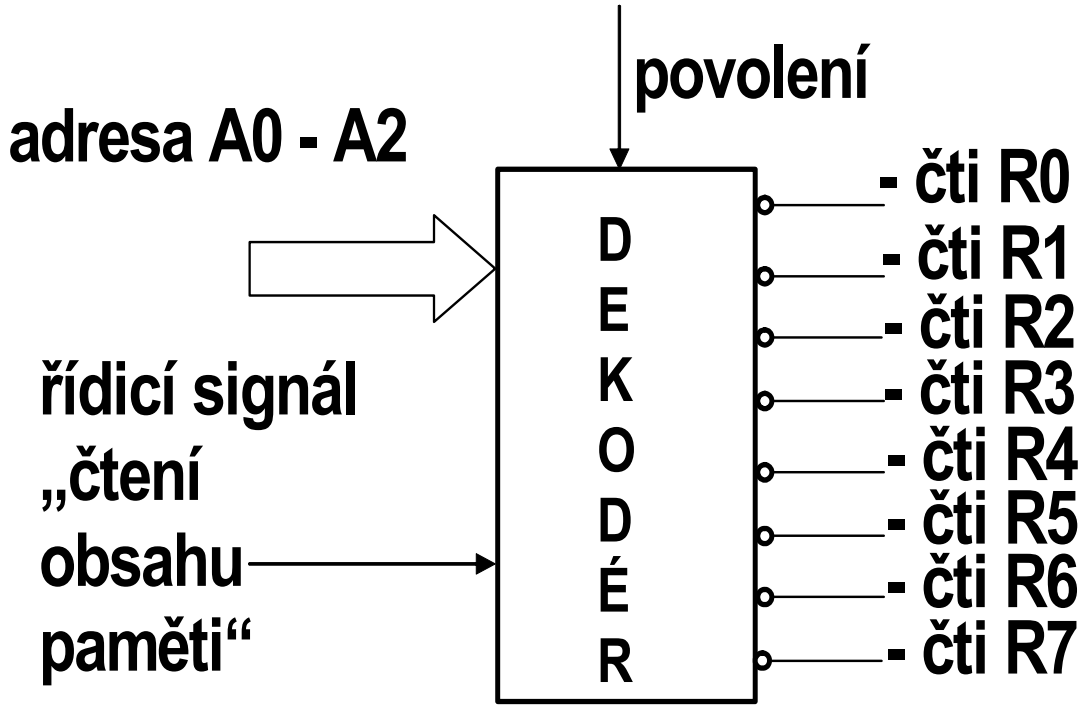


Obr. 3 Zápis/čtení do/z registru



Obr. 4 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 značně širší než repertoár instrukcí pro práci s registry.



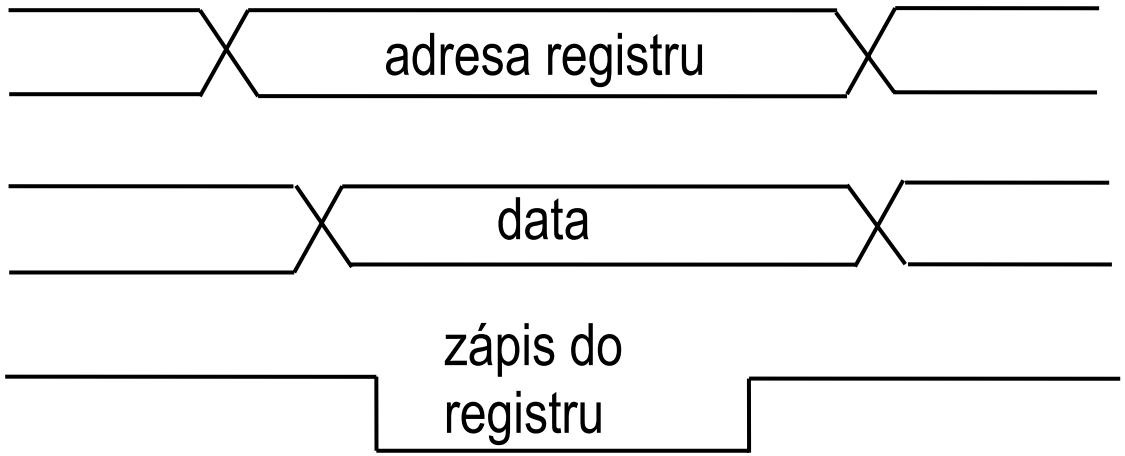
Obr. 5 Dekódování signálů pro čtení obsahu adresovaného registru (registry mapovány do adresového prostoru operační paměti)

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

Časové diagramy komunikace na sběrnici

- Popíšeme dva principy:

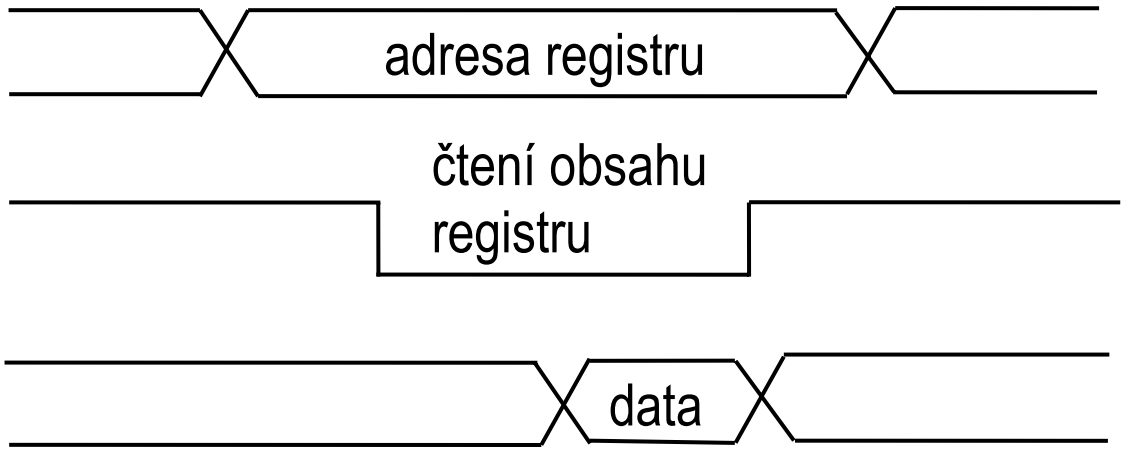
  Jednoduchá komunikace řízená řadičem sběrnice (řadič sběrnice je řízen procesorem jistým počtem bitů odvozenými od konkrétní instrukce IN/OUT).



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 patřičného registru vložena instrukcí.

- „Zápis do registru“ – jeden ze signálů řídicí části systémové sběrnice, generován jako výsledek realizace instrukce.
- Data jsou do registru na přijímací straně vložena nástupnou hranou signálu „zápis do registru“.
- Časovou relaci mezi daty a signálem „zápis do registru“ zařídí řadič sběrnice.
- Není žádná další komunikace kromě zápisu dat - výrazně malá režie.



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.

- Sběrnice PCI – režie existuje (poslání adresy, příkazu), následně však blokový přenos dat => režie se redukuje.

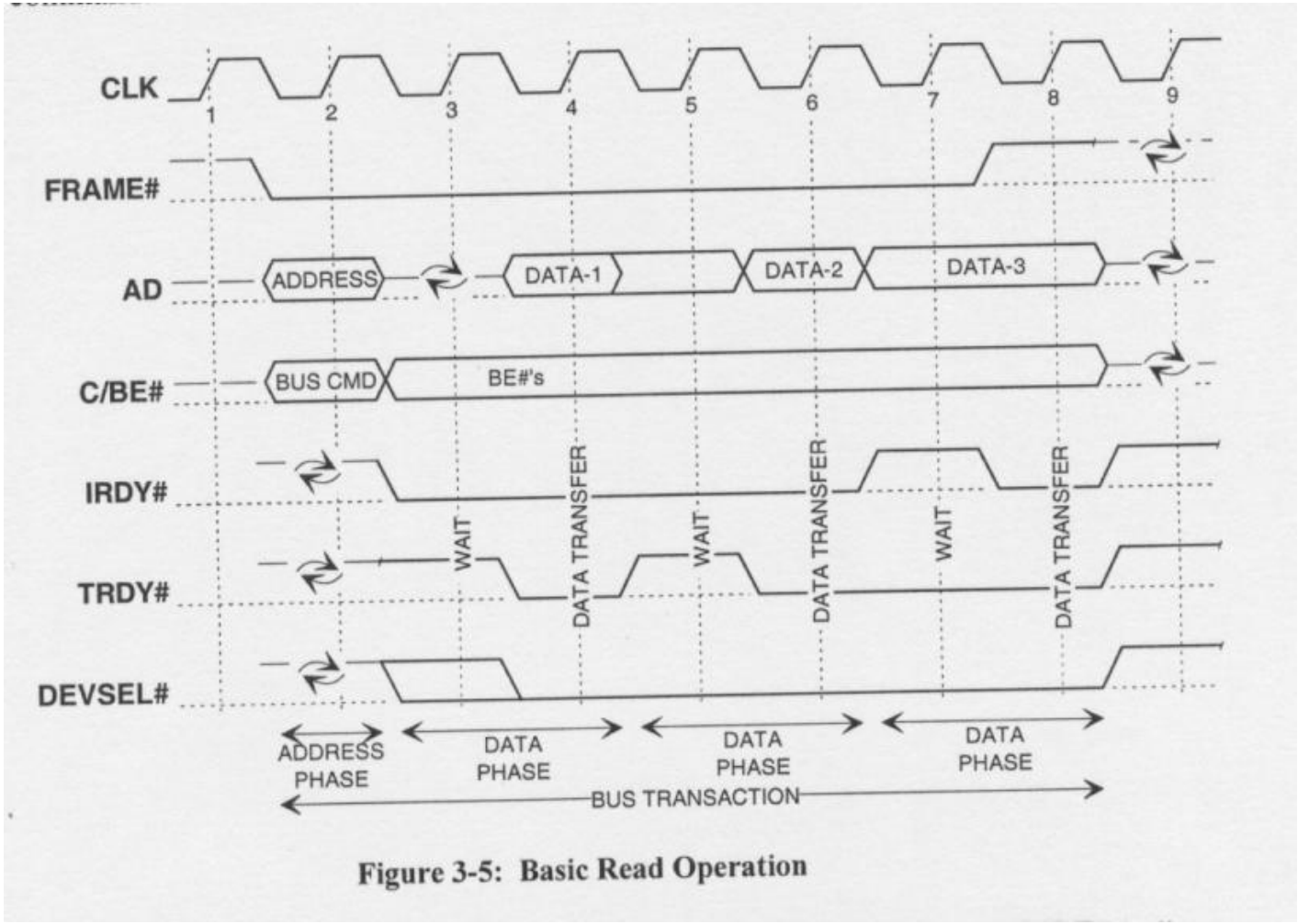
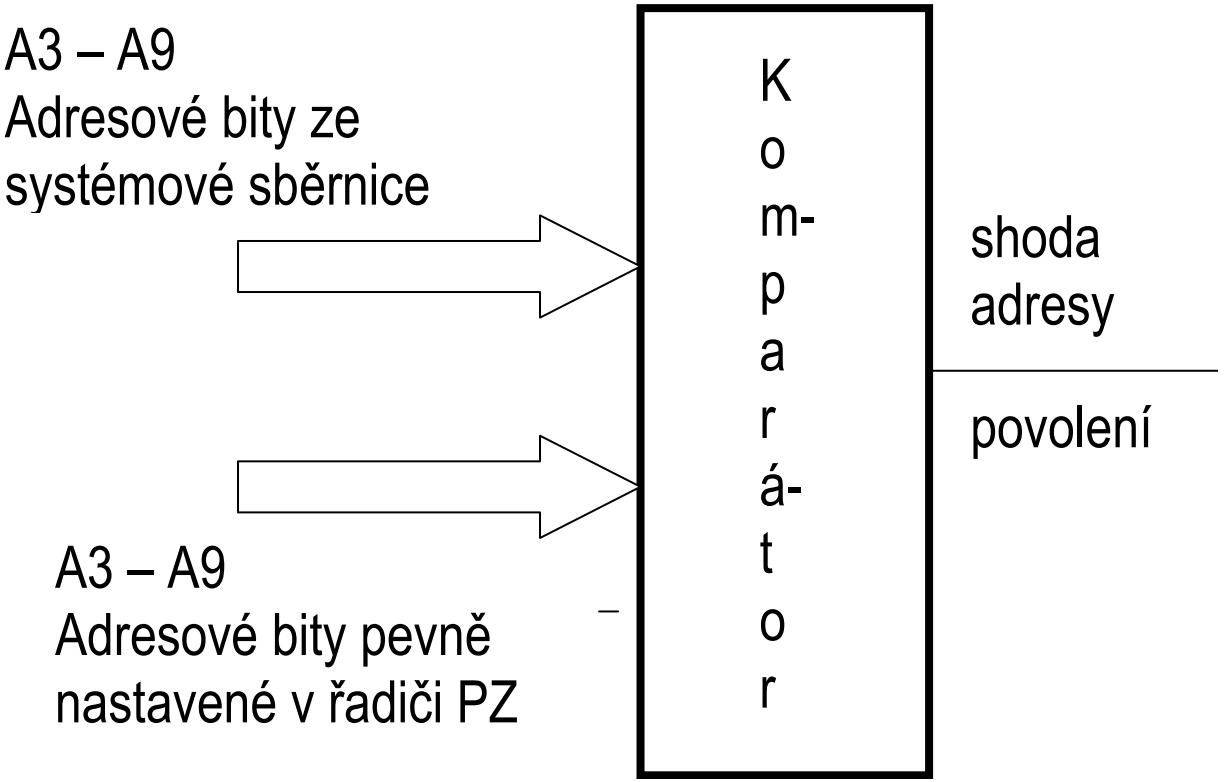


Figure 3-5: Basic Read Operation

Princip generování signálu „povolení“

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



Obr. 6 Generování signálu „povolení“

- Výsledek: pokud je shoda obsahu adresových bitů adresové sběrnice s adresou nastavenou v řadiči PZ – povolí se zápis do registrů řadiče – uvolní se dekodéry registrů v řadiči PZ.

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.

Shrnutí

- **Isolované vstupy/výstupy**
  - registry mají svůj vlastní adresový prostor (tzn. **existují 2 disjunktní adresové prostory** – paměťový adresový prostor, adresový prostor pro vstupy/výstupy),
  - 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.

Principy řízení a obsluhy periferních operací

Základní pojmy

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

  sběrnice PC XT,
  sběrnice ISA
  sběrnice EISA

sběrnice lokální (PCI, VLB)

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

- **Co musí externí adaptér umět:**

  rozpoznat, že je adresován některý z jeho vnitřních prvků,
  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 sběrnici (např. generovat IRQ – žádost o přerušení, žádost o přenos DMA).

Techniky řízení vstupu/výstupu dat

- Jsou tři:
  1. programové řízení vstupu/výstupu dat,
  2. vstup/výstup využívající přerušení,
  3. vstup/výstup přes DMA (přímý přístup do paměti).
- Všechny tyto techniky jsou odlišné => vyžadují zcela odlišnou **hardwarovou i softwarovou podporu** jak na straně počítače, tak na straně externího adaptéru.
- **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 operaci:

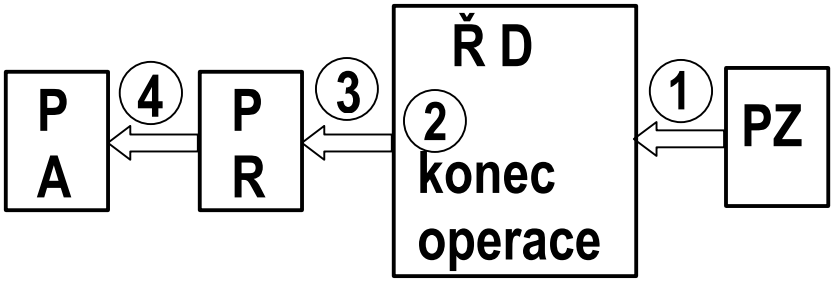
  Adaptér: na straně adaptéru je registr, do něhož se z PZ **přenese slovo dat** a nastaví se bit "**konec operace**" ve stavovém registru. Obdobně při opačném směru přenosu.

  Procesor:

  Průběžně **testuje obsah stavového registru**, konkrétně bitu "**konec operace**".

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

  Obdobně při opačném směru.



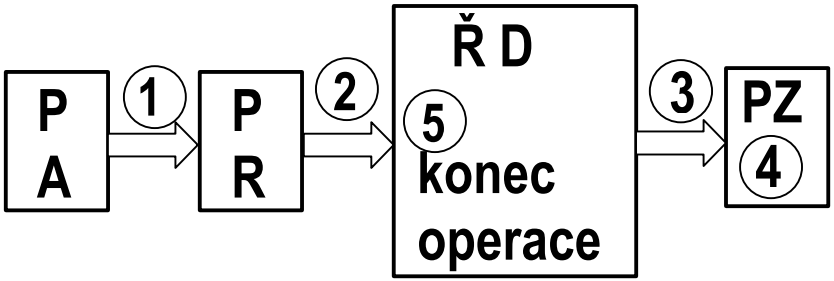
PA - paměť
PR - procesor
ŘD - řadič
PZ - periferní zařízení

Obr. 10 Posloupnost činností při přenosu slova dat z PZ do paměti:

1. přenos slova z PZ do řadiče PZ
2. nastavení bitu "konec operace" ve stavovém registru v adaptéru

v průběhu bodů 1. – 2. - průběžné testování obsahu stavového registru procesoru

3. přenos slova dat z registru řadiče PZ do registru procesoru
4. přenos slova dat do paměti



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

Obr. 11 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 výstupního 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 3) - 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 je ve správných okamžicích hardwarově nastavovat ze strany PZ (tak, aby indikoval ukončení periferní operace),
  2. musí být testovatelný přímo instrukcí podmíněného skoku
  3. ze strany procesoru musí být možné jej před zahájením periferní operace softwarově 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 byly 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**.

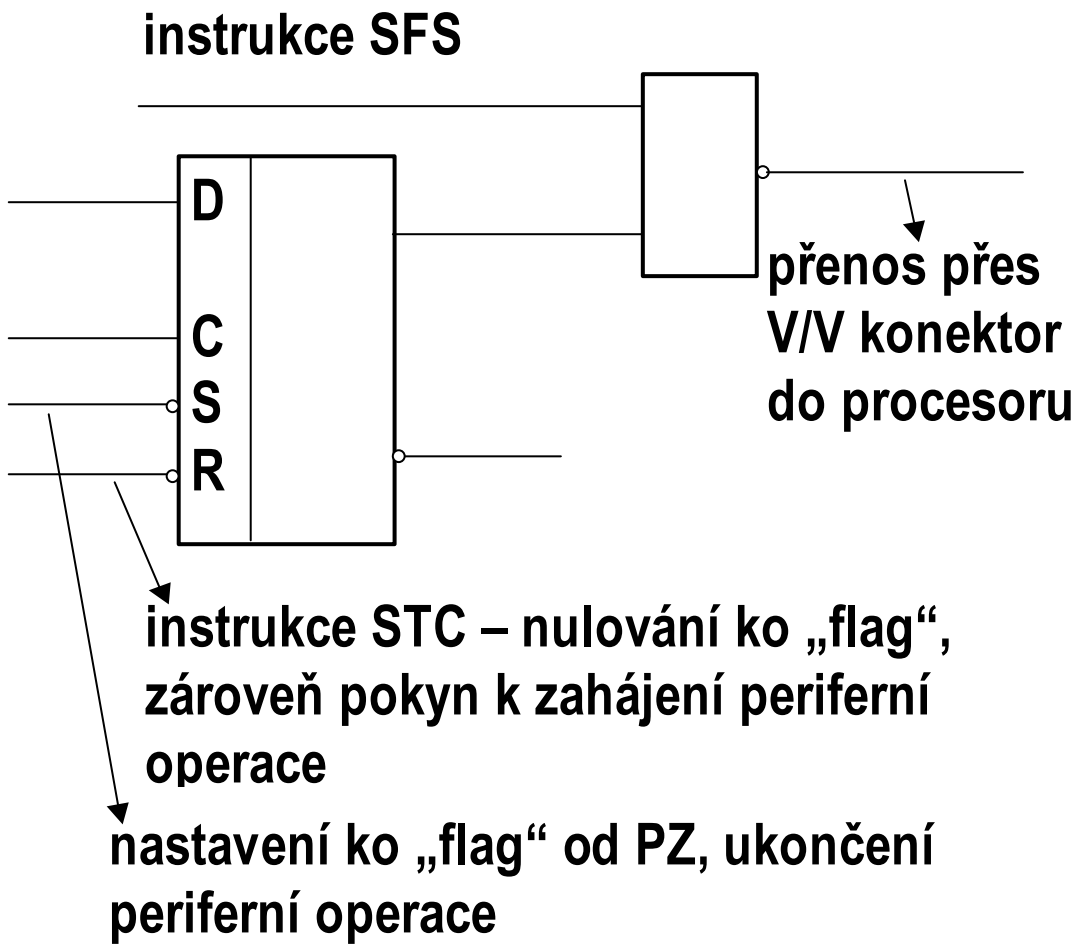
Posloupnost instrukcí pro komunikaci procesoru s řadičem PZ – **výstup dat**:

1. **OTA** - přenos slova dat z registru procesoru do registru řadič PZu
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. 12 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).

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.