

Řízení IO přenosů řadičem DMA

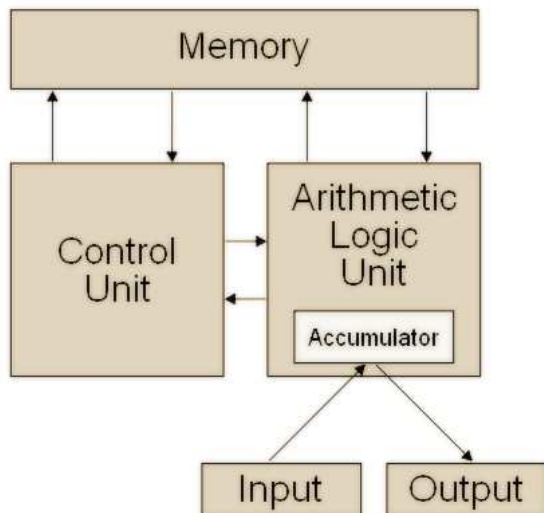
Řízení IO operací

Komunikace s IO zařízením může být řízena podle některé ze základních koncepcí:

- Programové řízení IO operací procesorem.
- Řízení IO operací s využitím přerušení.
- **Řízení IO operací řadičem DMA.**
- Použití specializovaného IO procesoru.

Historické souvislosti

- Klasický přístup: IO přenosy jsou řízené procesorem.



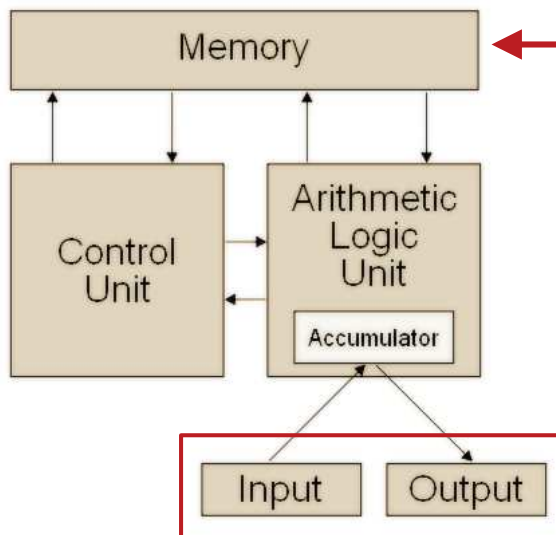
„Klasické“ schéma počítače



3

Historické souvislosti

- Klasický přístup: IO přenosy jsou řízené procesorem.
- DMA (Direct Memory Access): data z/do IO řadiče se přenáší přímo do/z paměti.



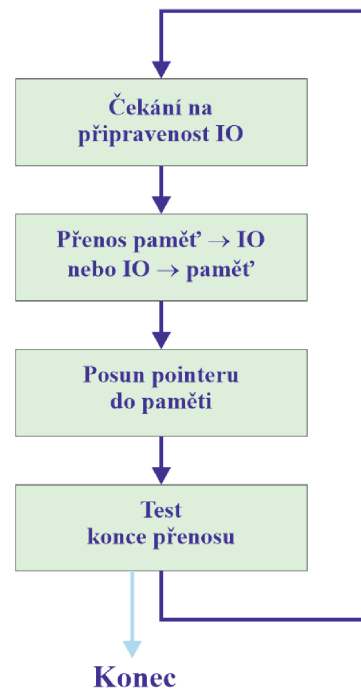
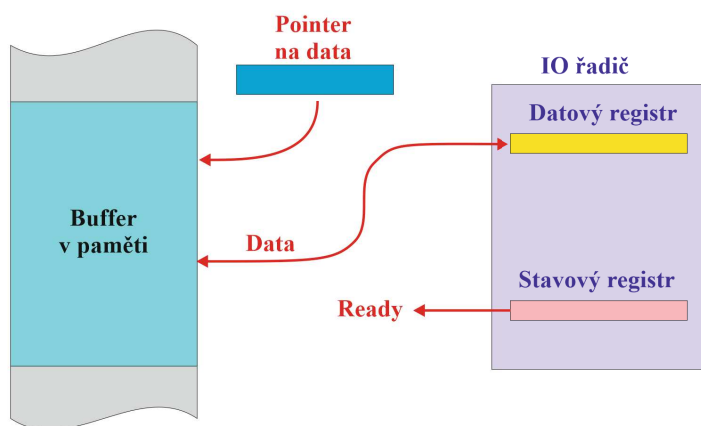
„Klasické“ schéma počítače



4

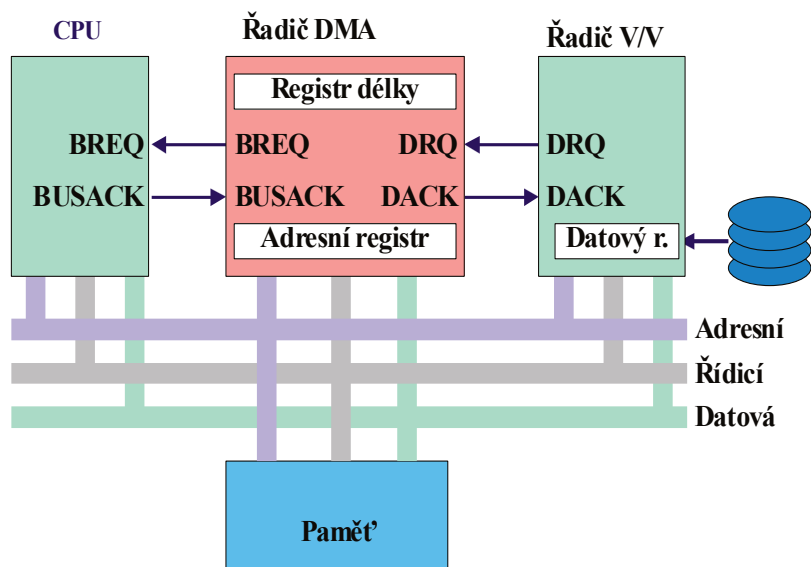
Koncepce DMA

- Při vstupu nebo výstupu dat se opakují jednoduché činnosti.
- Jednotlivé kroky lze realizovat pomocí speciálního HW.



Blokové schéma IO systému s DMA

- DMA řadič s kradením cyklů (**cycle stealing**):
 - V průběhu „normální“ činnosti je na sběrnici připojen procesor.
 - Při požadavku na DMA přenos se procesor po žádosti z DMA řadiče odpojí od sběrnice.

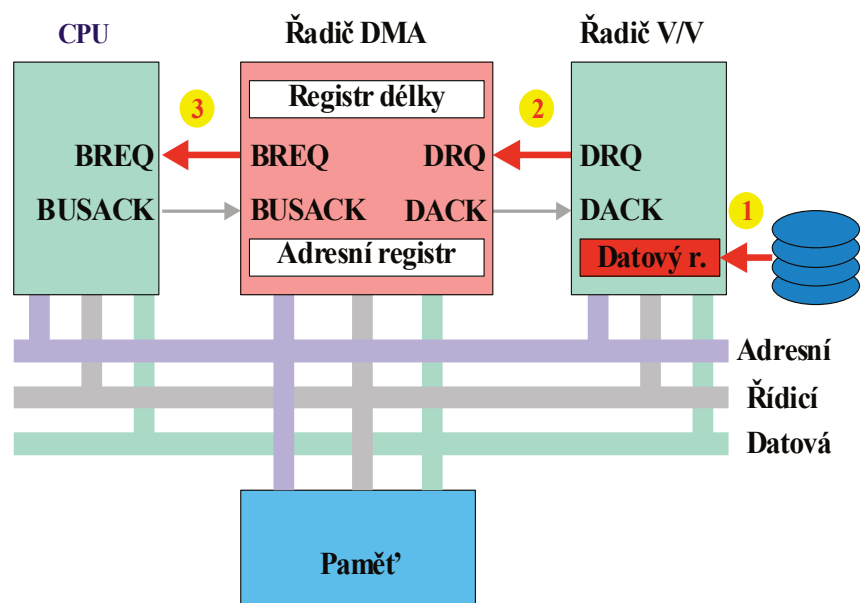


Průběh DMA přenosu

- IO řadič indikuje požadavek na přenos dat signálem DRQ (Data Requst).
 - DMA řadič žádá CPU o uvolnění sběrnice signálem BREQ (Bus Requst).
 - CPU indikuje uvolnění sběrnice signálem BUSACK (Bus Acknowl^{edge}).
 - DMA řadič požaduje přenos dat do/z IO signálem DACK (DMA Acknowl^{edge}).
 - Čtení/zápis dat z/do paměti se řídí obvyklými signály na sběrnici.
-
- Po přenesení celého bloku může DMA řadič generovat přerušení – indikace konce přenosu.

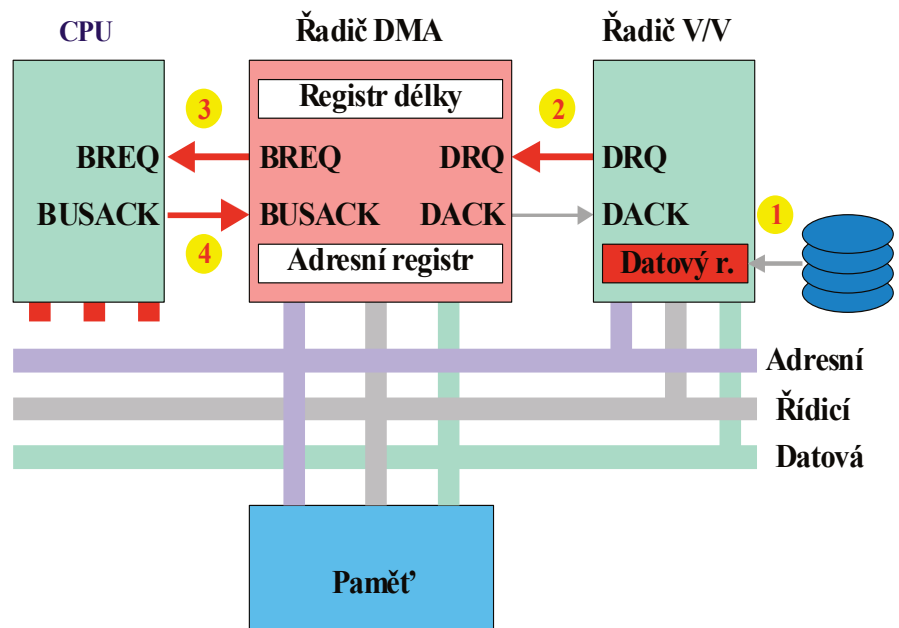
Přenos DMA – vstup dat (1)

1. Datový registr IO řadiče je plný.
2. IO řadič generuje DRQ.
3. DMA řadič generuje BREQ.



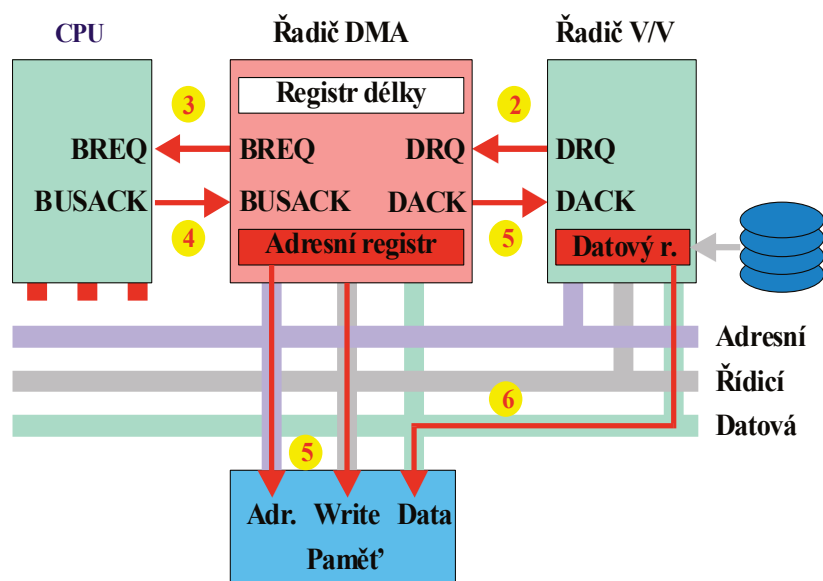
Přenos DMA – vstup dat (2)

4. Procesor dokončí probíhající sběrniceový cyklus. Potom se odpojí od sběrnice a aktivuje BUSACK.



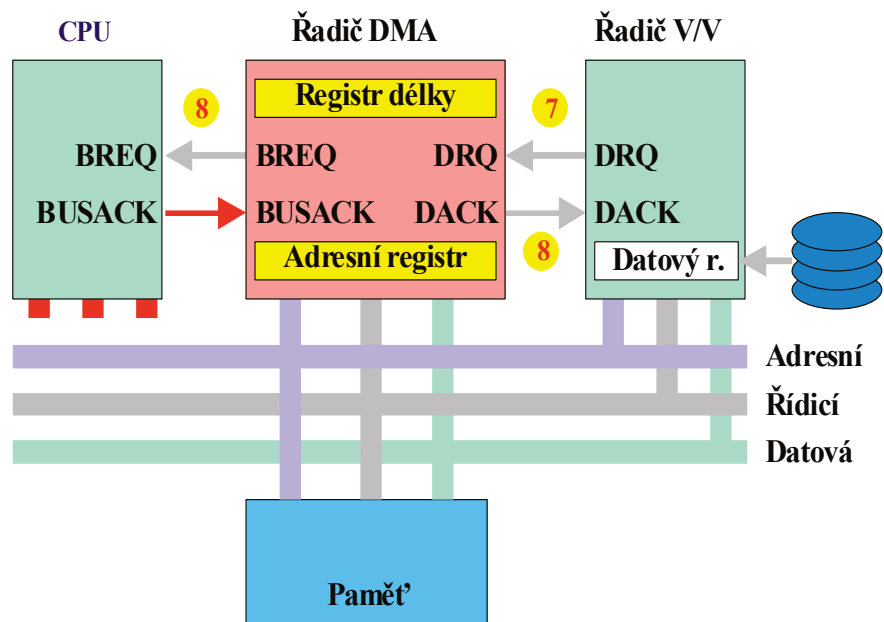
Přenos DMA – vstup dat (3)

5. DMA řadič aktivuje DACK a vysílá na sběrnici adresu a povel „zápis“.
6. IO řadič vysílá na datovou sběrnici obsah datového registru.



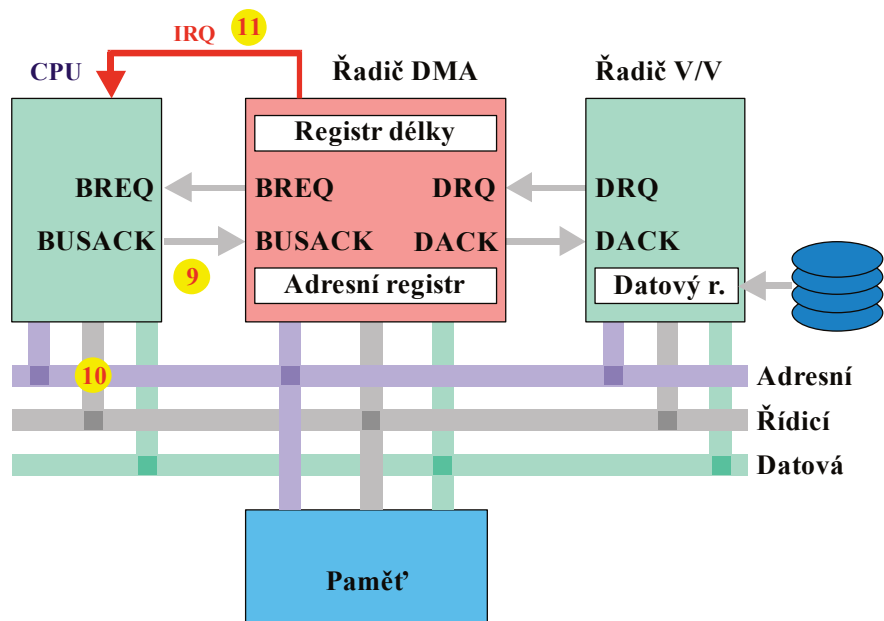
Přenos DMA – vstup dat (4)

- 7. IO řadič zruší DRQ.
- 8. DMA řadič zruší BREQ a DACK, přestane vysílat adresu a povel „zápis“. Registr adresy se inkrementuje, registr délky dekrementuje.



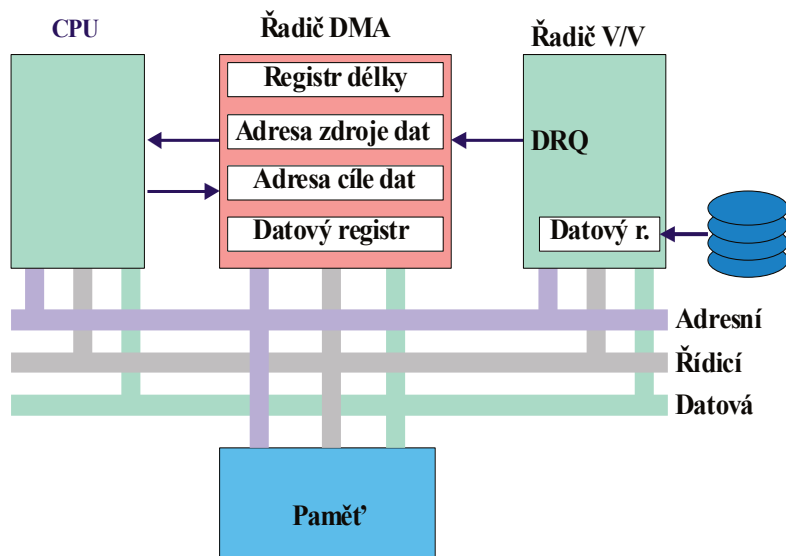
Přenos DMA – vstup dat (5)

- 9. CPU zruší BUSACK.
- 10. CPU se připojí na sběrnici.
- 11. Po přenesení celého bloku generuje DMA řadič přerušování.



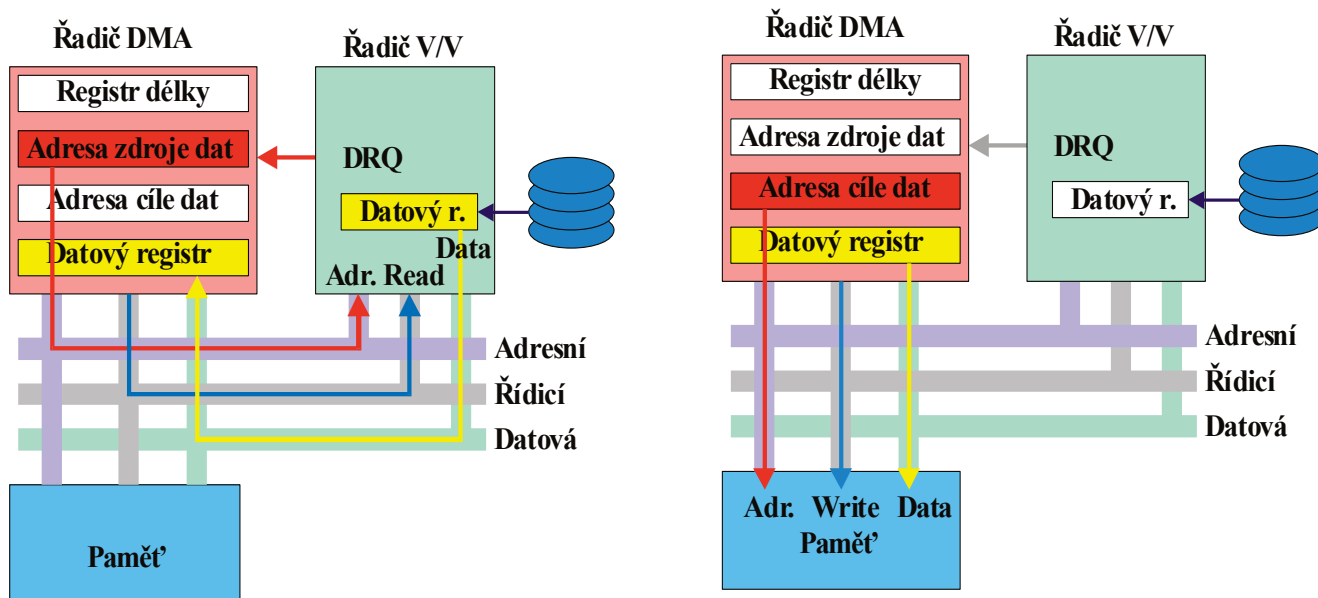
DMA řadič se dvěma cykly

- Jednodušší návrh IO řadiče.
- Univerzální použití, lze provádět přenosy paměť → paměť.



Funkce DMA řadiče se dvěma cykly

1. Čtení ze zdroje dat do interního datového registru.
2. Zápis dat z interního datového registru do cíle.

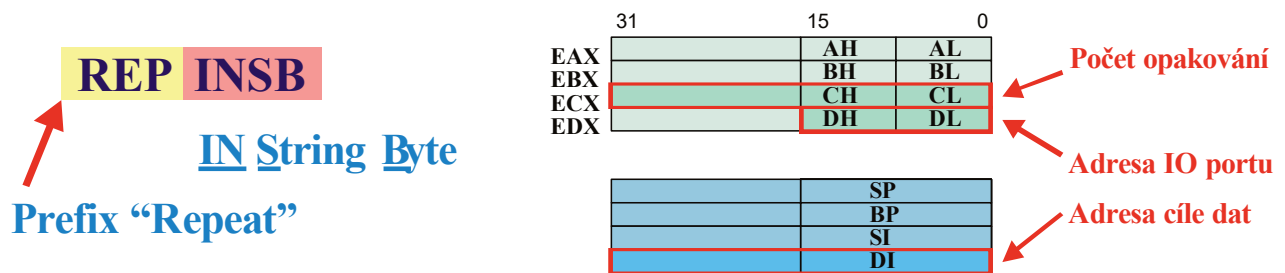


Poznámka: Komunikace s procesorem a jeho odpojení od sběrnice probíhá podobně jako v předchozím případě.

Procesor ve funkci DMA řadiče

- Některé procesory mají speciální instrukci pro blokový přenos.
 - Během přenosu se obvykle instrukce opakovaně čte z paměti ⇒ pomalejší než samostatný DMA.
- Synchronizace s periferním řadičem pomocí signálu WAIT.

Příklad: Intel IA-32 → Instrukce **INSB** s prefixem **REP**:

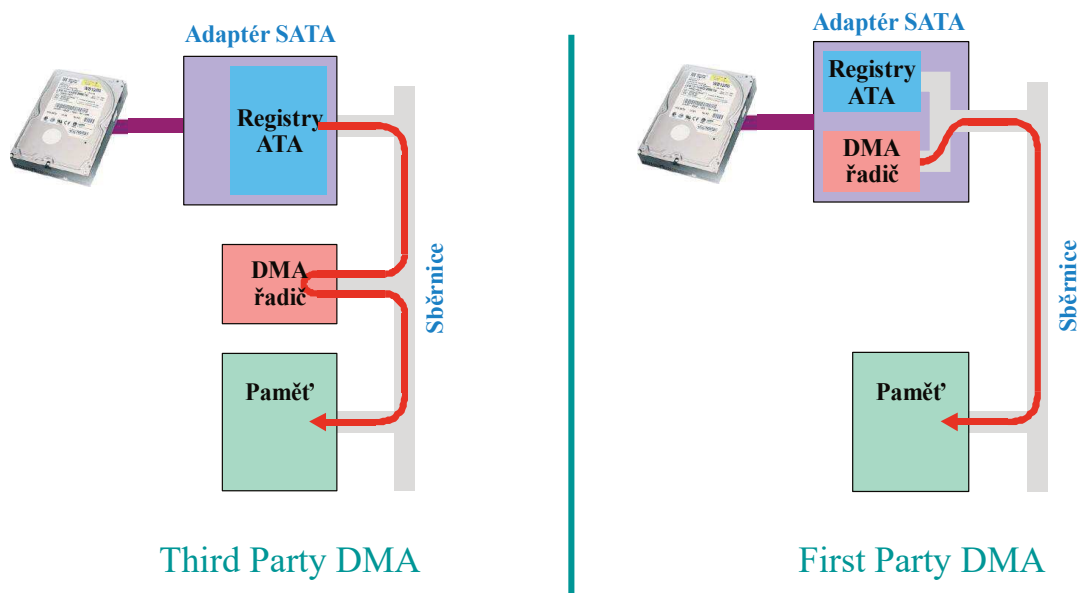


Charakteristika DMA přenosu

- Velmi rychlý přenos dat (nejrychlejší způsob přenosu).
- Vhodný pro blokové přenosy.
- Vyžaduje speciální HW prostředky.
- Komplikovaný provoz na sběrnici.

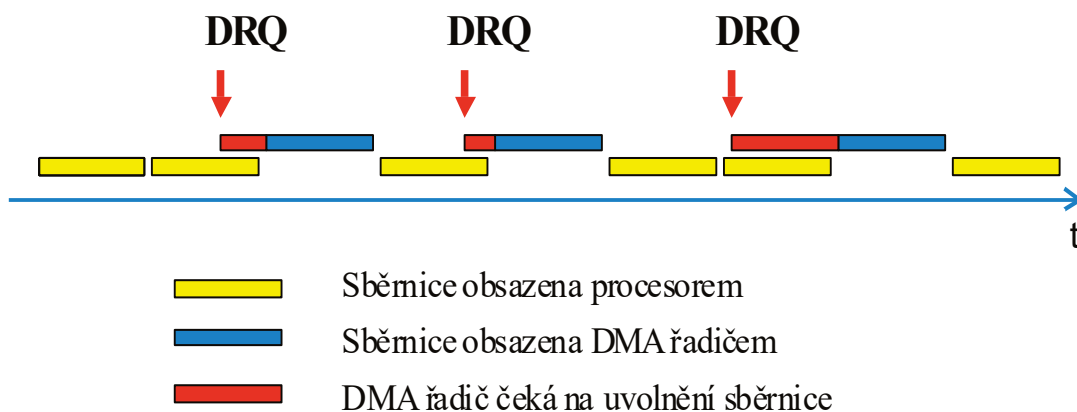
First Party DMA

- Third Party DMA: DMA řadič je samostatný prvek na sběrnici.
- First Party DMA: DMA řadič je součástí adaptéru (SATA, USB nebo jiného periferního řadiče).



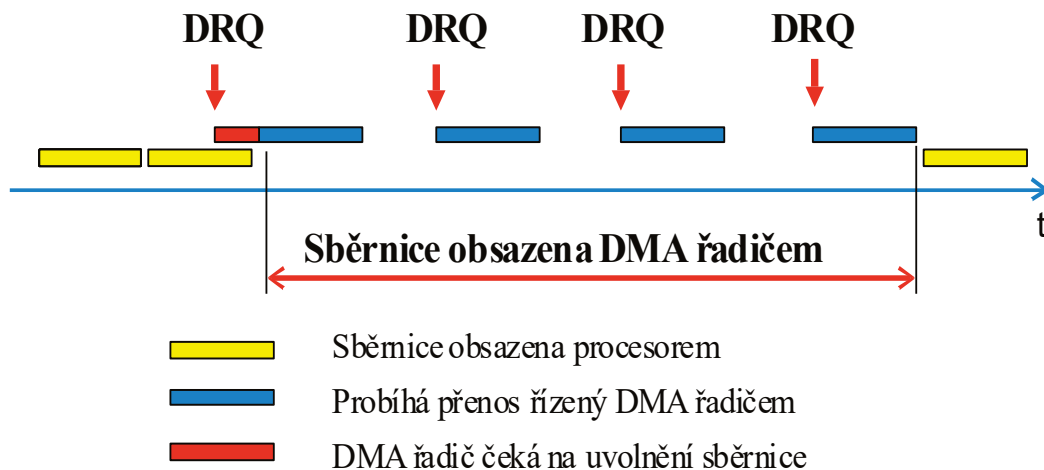
Sdílení sběrnice (1)

- CPU a DMA mohou sdílet sběrnici dvěma způsoby:
 1. DMA uvolňuje sběrnici po každém přenosu (cycle stealing).
 - Procesor není trvale blokován,
 - zpoždění před přidělením sběrnice DMA řadiči.



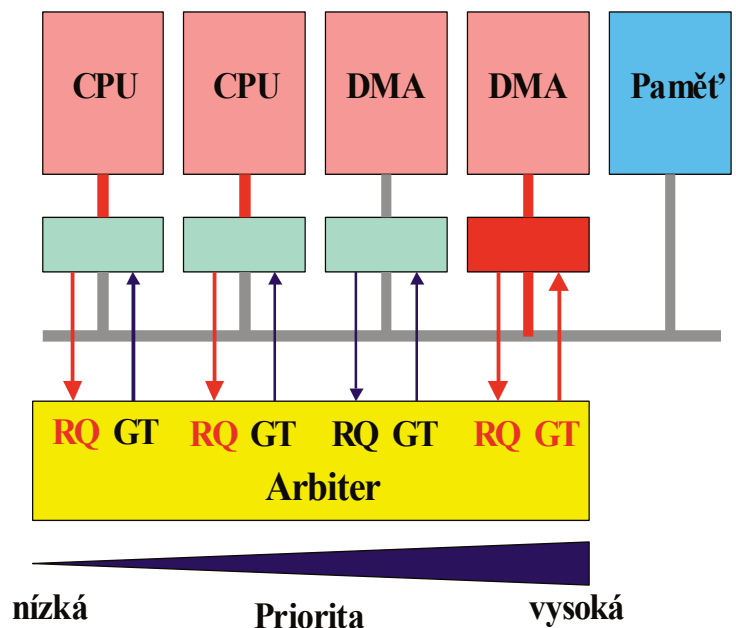
Sdílení sběrnice (2)

- CPU a DMA mohou sdílet sběrnici dvěma způsoby:
 1. DMA drží obsazenou sběrnici po celou dobu přenosu bloku (burst mode).
 - Procesor ve stavu HOLD po celou dobu přenosu,
 - nejvyšší rychlost DMA přenosu.



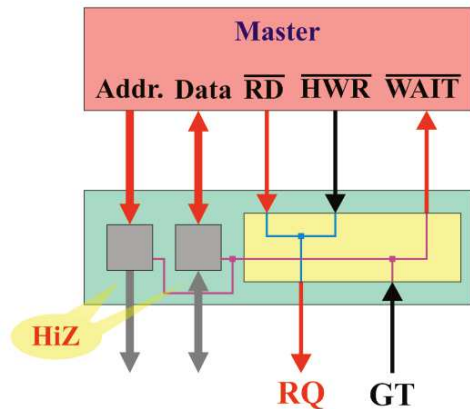
Sběrnice typu MULTIMASTER

- **Master** – má schopnost řídit přenosy na sběrnici (CPU, DMA).
- **Slave** – nemá schopnost řídit přenosy (paměť, I/O řadič).
- Před každým přenosem (tj. sběrnicevým cyklem) musí proběhnout arbitrace:
 - Arbiter povolí přístup na sběrnici jen Masteru s nejvyšší prioritou.
 - Ostatní Mastery musí počkat.

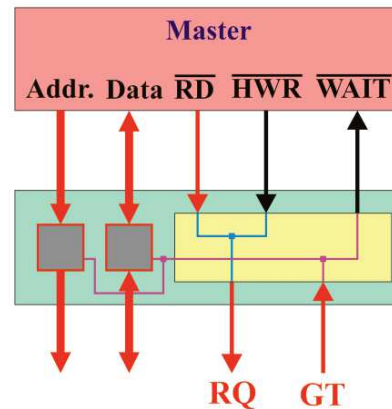


Připojení masteru

- Adresní a datové vodiče a signály \overline{RD} a \overline{HWR} jsou na sdílenou sběrnici připojeny prostřednictvím třístavových budičů.
 - Má-li master povolen přístup na sběrnici, jsou budiče v aktivním stavu.
 - Nemá-li master povolen přístup na sběrnici, jsou budiče ve stavu HiZ. Master čeká na sběrnici ve stavu WAIT.



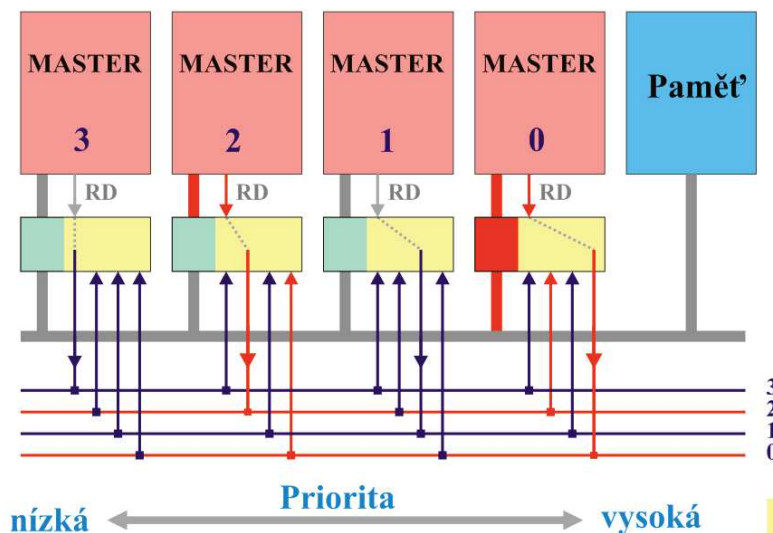
Přístup na sběrnici odmítnut



Přístup na sběrnici povolen

Decentralizovaný arbiter

- Každý master ovládá jeden arbitrační vodič. Současně sleduje aktivitu ostatních vodičů a porovnává prioritu aktivních vodičů se svou prioritou.



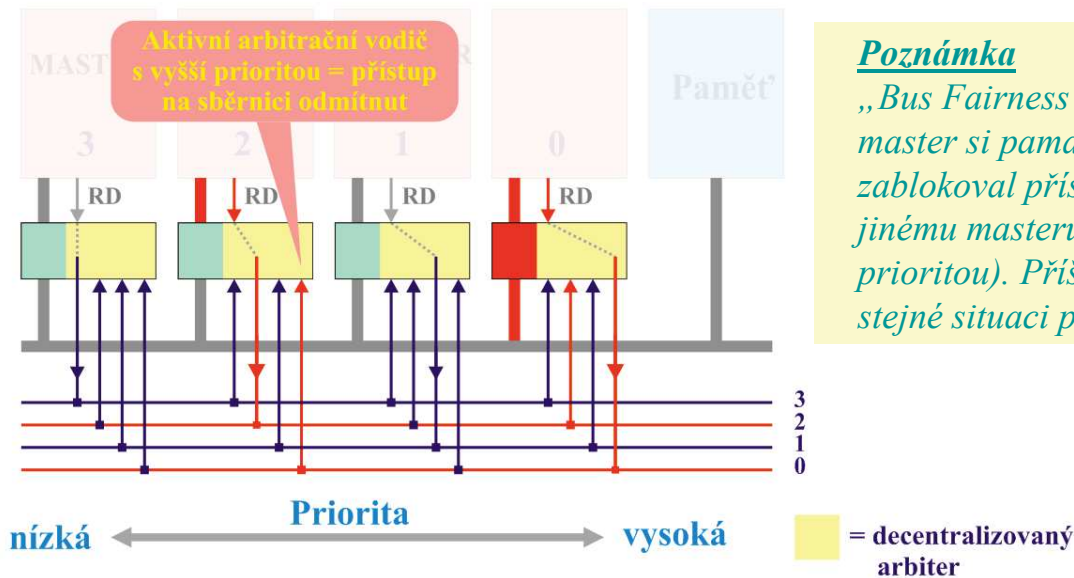
Poznámka

„Bus Fairness“: Vítězný master si pamatuje, že zablokoval přístup na sběrnici jinému masteru (s nižší prioritou). Příště mu dá ve stejné situaci přednost.

= decentralizovaný arbiter

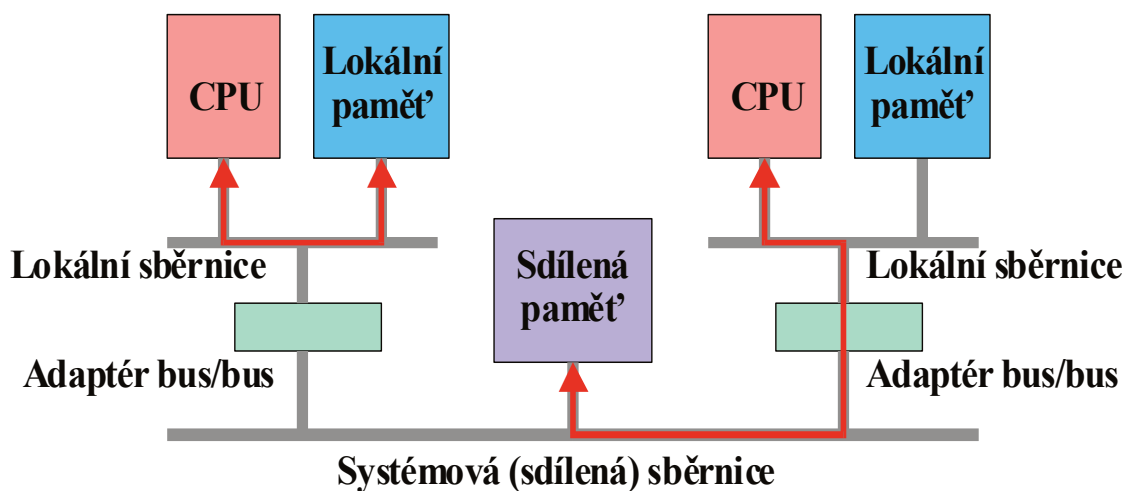
Decentralizovaný arbiter

- Každý master ovládá jeden arbitrační vodič. Současně sleduje aktivitu ostatních vodičů a porovnává prioritu aktivních vodičů se svou prioritou.



Složitější architektura

- Každý procesor může používat nezávislou lokální sběrnici a lokální paměť.
- Pro sdílená data je určena sdílená paměť, přístupná oběma (všem) procesorům.



Adresní prostor systému s lokální a sdílenou pamětí

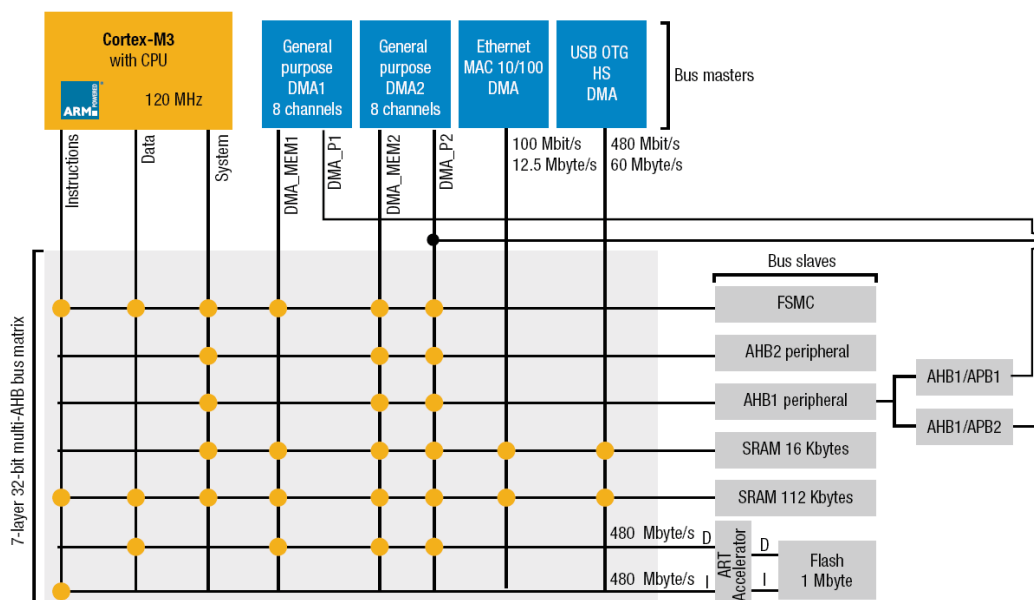
- Lokální a sdílená paměť leží ve stejném adresním prostoru.
- Každá z pamětí pokrývá jiný rozsah adres.
- Při adresování sdílené paměti žádá adaptér bus-bus o přístup na sdílenou sběrnici.

Adresní prostor



Složitější architektura sběrnice

- Příklad: procesor Cortex (STM32) se systémem několika sběrnic.
 - Současně může probíhat několik přenosů Master ↔ Slave.



Architektura PC

