

# FASTPort

Nová sběrnice pro připojení inteligentních karet\*  
k osmibitovým počítačům

*aneb*

Jak připojit koprocessor

*\*) inteligentní karta = karta vybavená vlastním procesorem*

# úvodní problém:

Jak připojit k osmibitovému počítači periférii vybavenou vlastním procesorem pracujícím na odlišné frekvenci, popřípadě i majícím odlišnou architekturu:

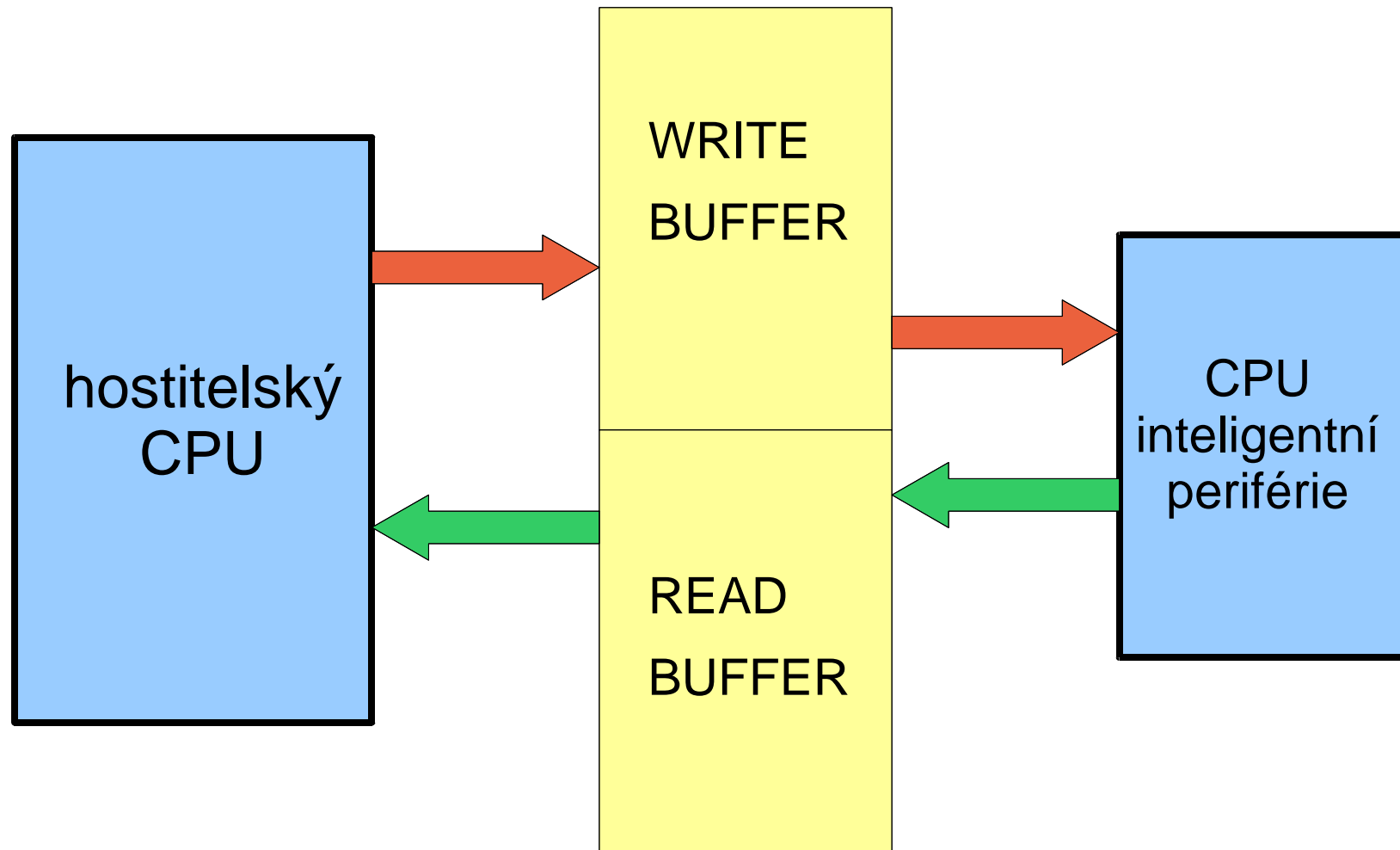
- co nejrychleji (tj. s maximální datovou propustností)
- co nejšetrněji k prostředkům hostitelského počítače

# další požadavky:

- univerzální řešení, ne jen pro jeden konkrétní projekt
- rozšiřující karty zasunuté v backplanu připojeném k počítači (místo jedné externí periférie v krabičce)
- připojení až 4 karet
- možnost zasunout kartu do libovolného slotu bez nutnosti měnit její nastavení (jumpery apod.)

*Výsledné řešení přináší takové množství výhod, že si zaslouží realizaci.*

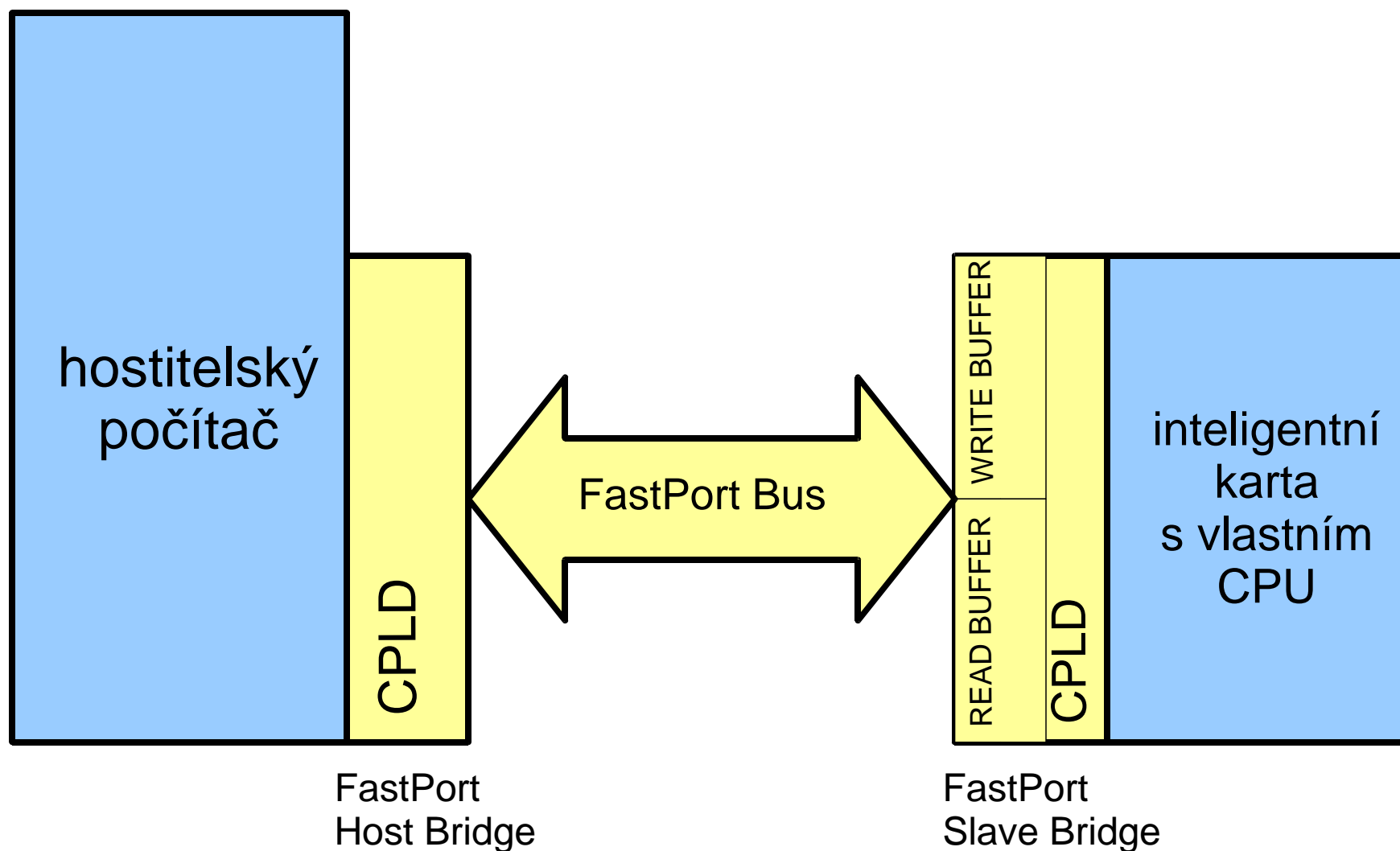
# Obečný princip použitého řešení:



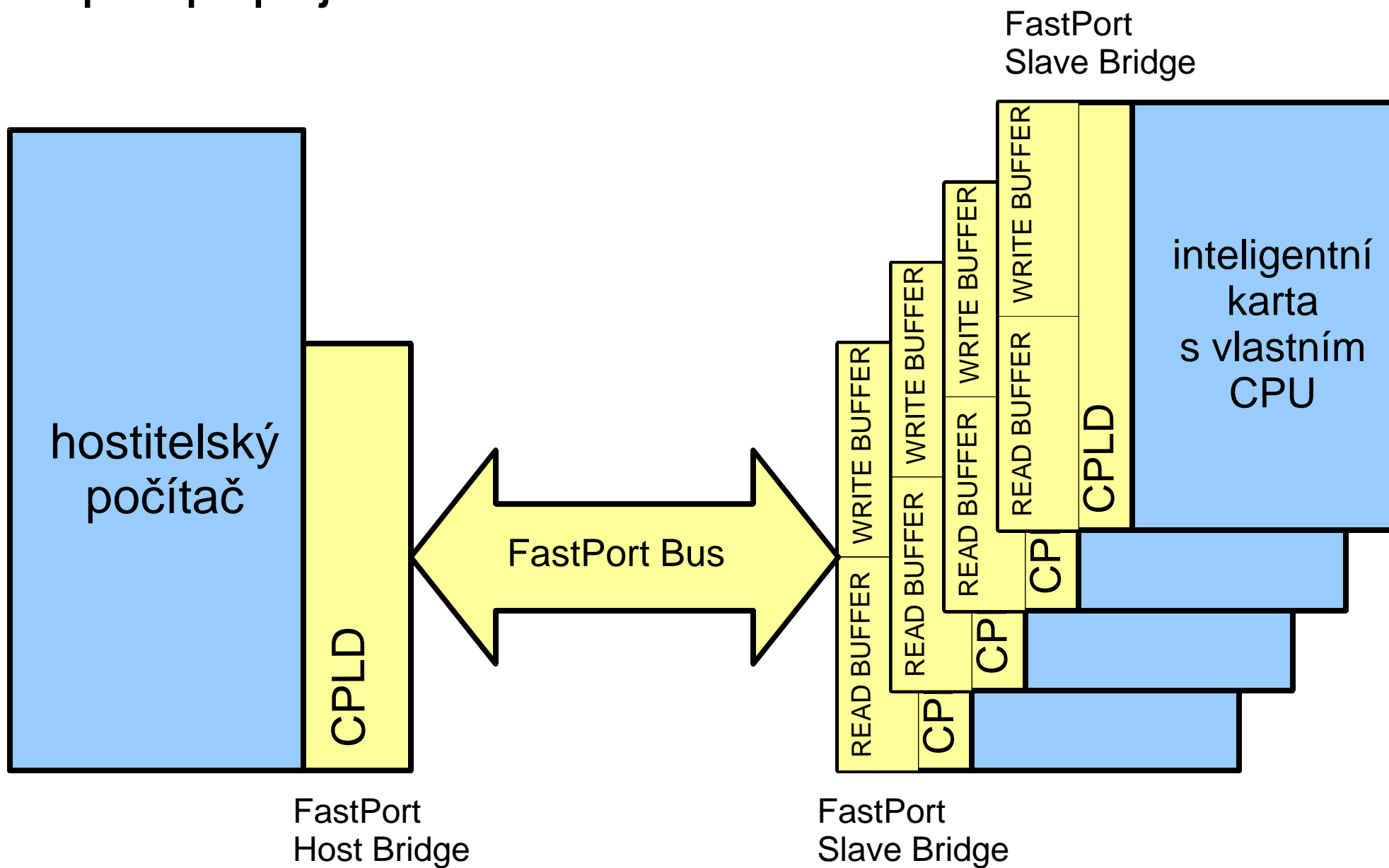
# Výhody použitého řešení:

1. procesory mohou být taktovány zcela nezávisle, komunikace mezi nimi je asynchronní

realizací tohoto řešení vznikla sběrnice FastPort...



...pro připojení až 4 karet





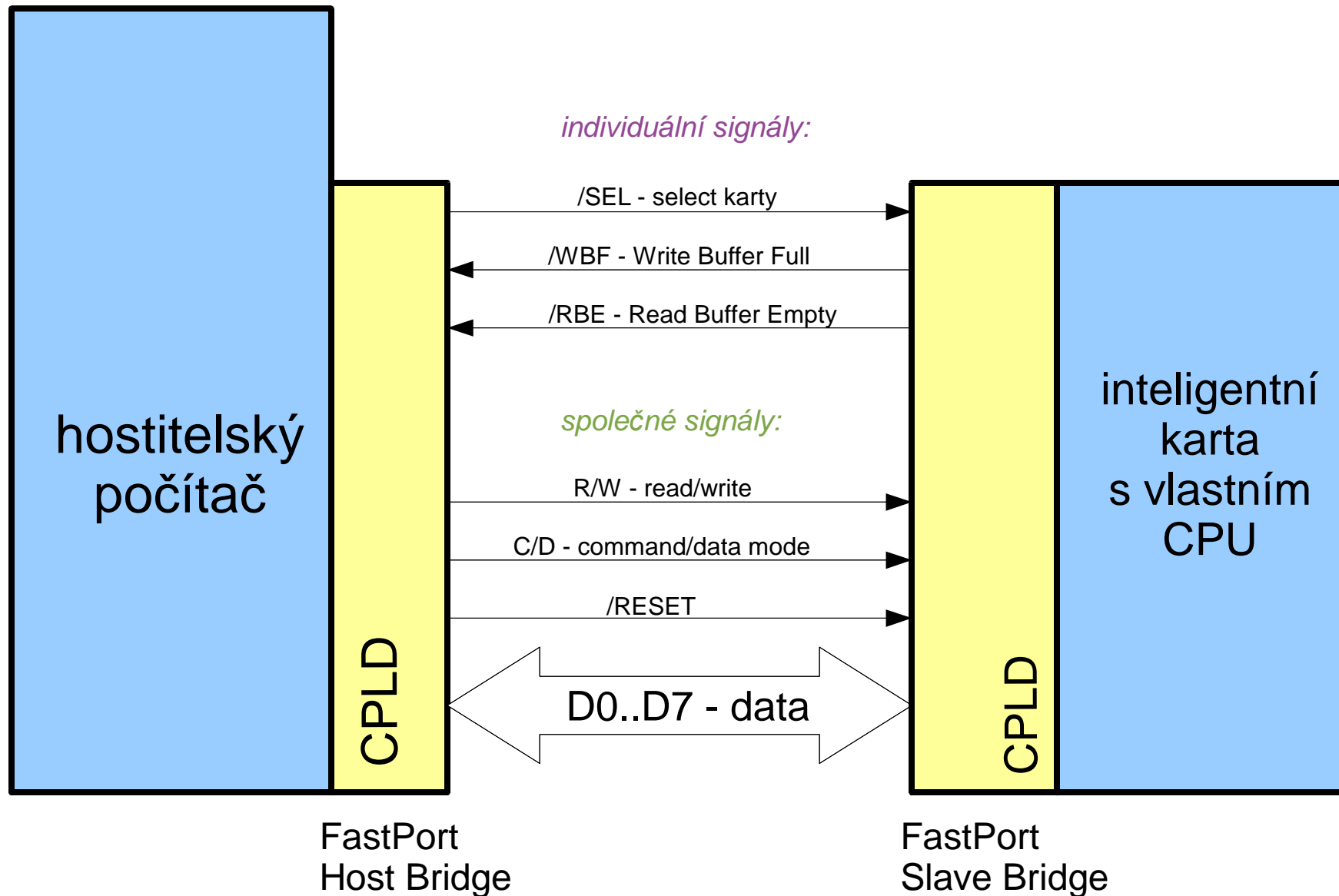
# Důležité vlastnosti výsledného řešení

- samotná sběrnice („FastPort Bus“) je platformně nezávislá, její signály neodpovídají konkrétnímu CPU, ale generuje je CPLD na straně hostitelského počítače („FastPort Host Bridge“), odpovědi generuje CPLD na straně karty („FastPort Slave Bridge“)
- oba buffery mohou mít libovolnou velikost (dokonce každý jinou). Minimum je 1 byte, (teoretické) maximum 64KB.
- „Masterem“ sběrnice FASTPort je vždy hostitelský počítač, karta se nesmí pokoušet vyvolat komunikaci - „odpovídá pouze když je tázána“. Dává ale najevo stavovým signálem, že má data k odeslání.

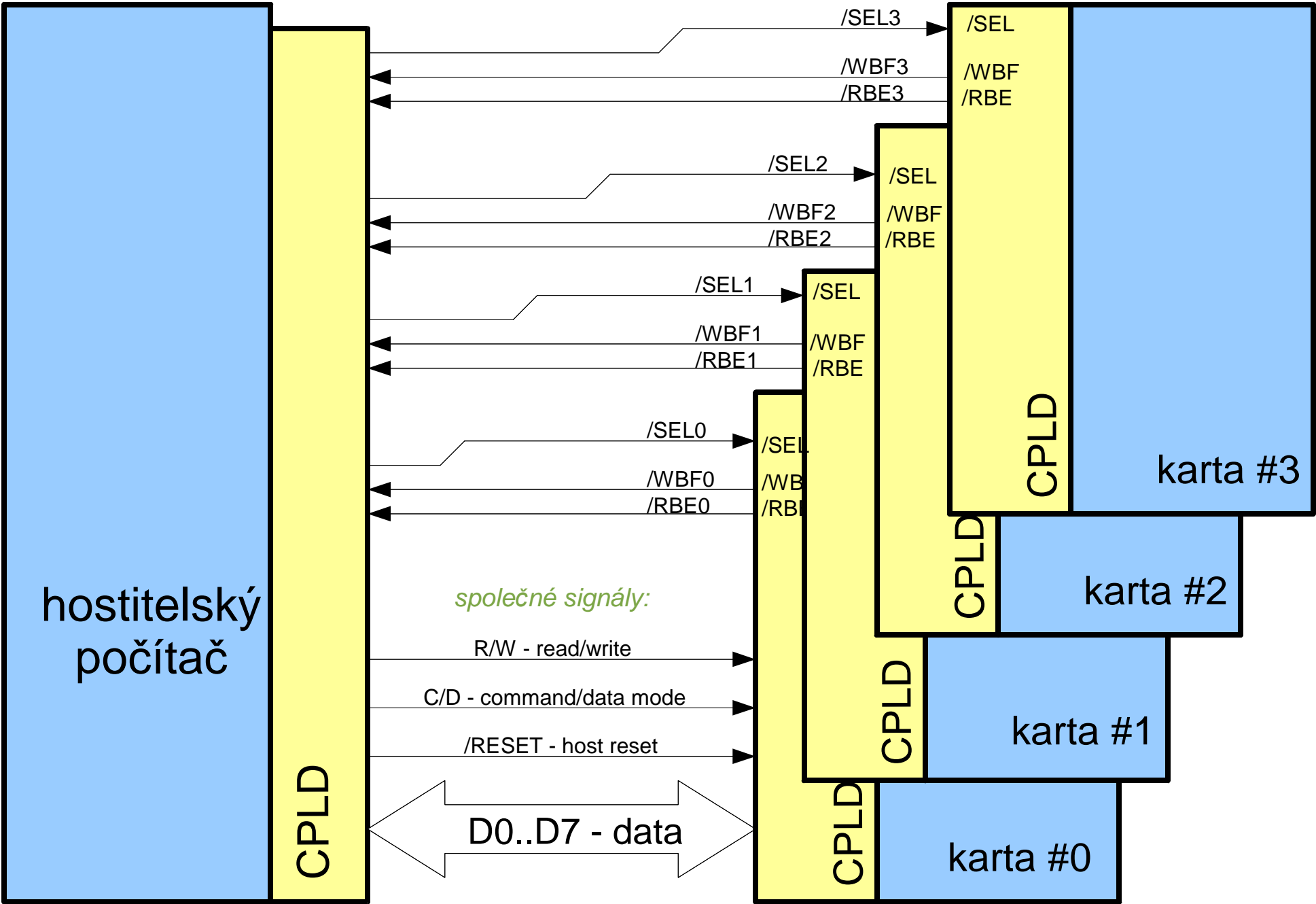
# Výhody použitého řešení:

1. procesory mohou být taktovány zcela nezávisle, komunikace mezi nimi je asynchronní
2. procesory mohou mít odlišnou architekturu
3. velikosti bufferů na kartě mohou být přizpůsobeny jejímu účelu
4. není zvětšována délka lokální sběrnice, protože všechny signály jsou nově buzeny

# signály sběrnice FASTport



# signály pro připojení všech 4 karet:



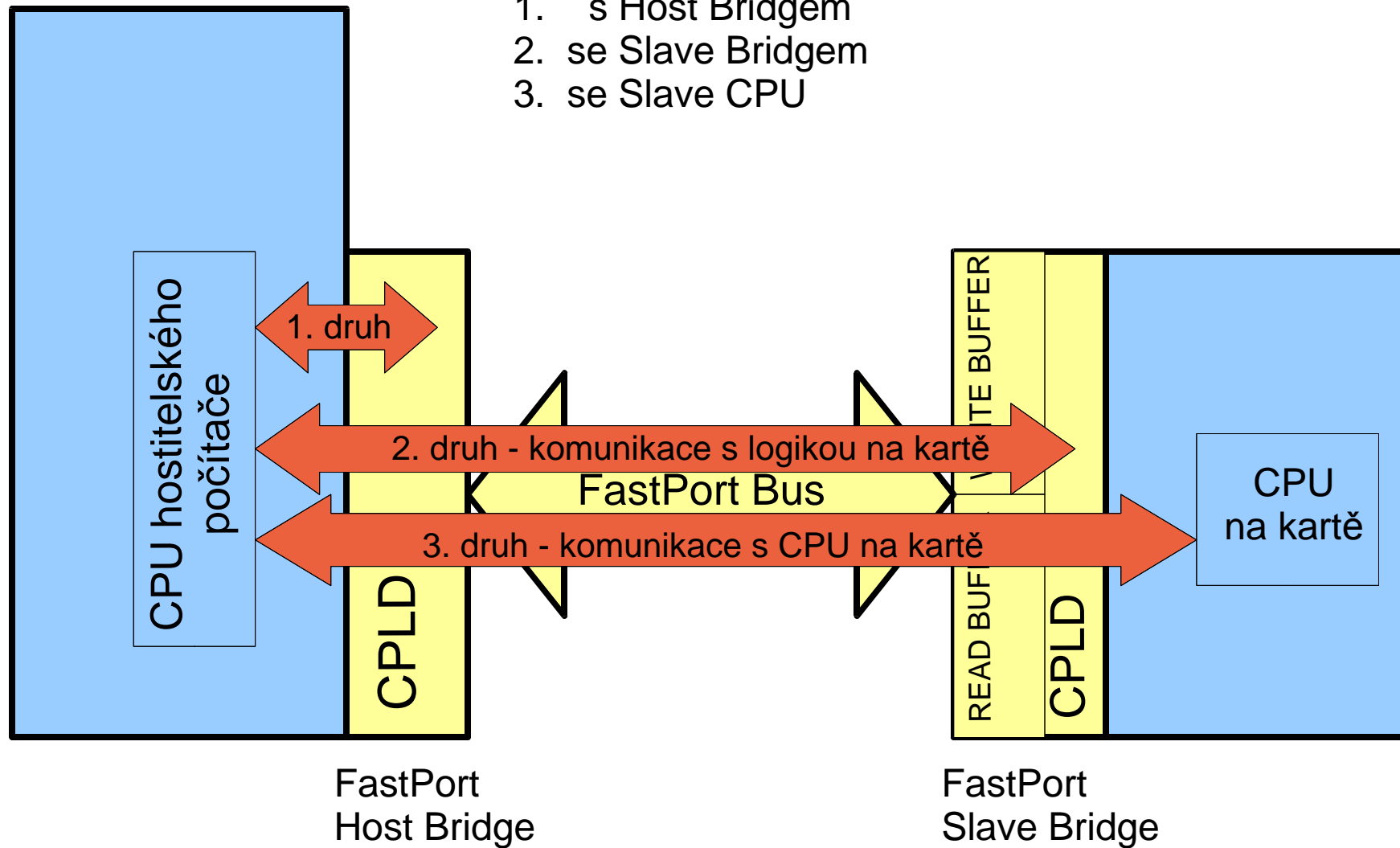
# Výhody použitého řešení:

1. procesory mohou být taktovány zcela nezávisle, komunikace mezi nimi je asynchronní
2. procesory mohou mít odlišnou architekturu
3. velikosti bufferů na kartě mohou být přizpůsobeny jejímu účelu
4. není zvětšována délka lokální sběrnice, protože všechny signály jsou nově buzeny
5. každá karta může pracovat v libovolném slotu bez nutnosti rekonfigurace
6. všechny karty sdělují stav svých bufferů najednou

# Jak to vlastně funguje

Používají se 3 druhy komunikace:

1. s Host Bridgem
2. se Slave Bridgem
3. se Slave CPU



# FastPort - druhy komunikace

1. komunikace s Host Bridgem (logikou na straně hostitelského počítače)

- *slouží k výběru karty a nastavení režimu komunikace s ní (při zápisu) a ke zjištění stavu karet (při čtení)*

2. komunikace se Slave Bridgem (logikou na kartě)

- *slouží pro zjištění ID a stavu karty a stavu bufferů (při čtení) a ovládání karty bez účasti jejího CPU (např. reset konkrétní karty)*

3. komunikace se „Slave CPU“ - procesorem na kartě

- ***hlavní komunikace s kartou - odesílání a příjem dat***

# 1. komunikace s Host Bridgem

Hostitelský počítač používá k práci s FASTPortem POUZE dva I/O porty:

## Control/Status Port

Control (zápis)

C/D	-	-	-	-	-	S1	S0
-----	---	---	---	---	---	----	----

*command/data*

*card select*

Status (čtení)

WBF3	WBF2	WBF1	WBF0	RBE3	RBE2	RBE1	RBE0
------	------	------	------	------	------	------	------

## Data Port

zápis: přepis signálů **D0..D7** lokální sběrnice na FastPort Bus jako **FP\_D0 .. FP\_D7**

čtení: přepis signálů **FP\_D0 .. FP\_D7** na lokální sběrnici jako **D0..D7**



# Výhody použitého řešení:

1. procesory mohou být taktovány zcela nezávisle, komunikace mezi nimi je asynchronní
2. procesory mohou mít odlišnou architekturu
3. velikosti bufferů na kartě mohou být přizpůsobeny jejímu účelu
4. není zvětšována délka lokální sběrnice, protože všechny signály jsou nově buzeny
5. každá karta může pracovat v libovolném slotu bez nutnosti rekonfigurace
6. všechny karty sdělují stav svých bufferů najednou
7. na straně hostitelského počítače jsou potřeba POUZE 2 VOLNÉ I/O PORTY
8. stav bufferů všech karet se zjistí jedinou I/O operací

## 2. komunikace se Slave Bridgem (logikou na kartě)

Pokud je předchozím zápisem na Control Port nastaven 7. bit (C/D) na 1 , bude v log.1 i signál C/D sběrnice FastPort a logika na kartě bude v tzv. Command/Response režimu.

V tomto režimu je bajt vyslaný na vodiče FP\_D0 .. FP\_D7 (zápisem na Data Port) brán jako jednobajtový příkaz (Command) pro logiku na kartě (a není předán procesoru). Při následujícím čtení z Data Portu logika na kartě vystaví na vodiče FP\_D0 .. FP\_D7 jednobajtovou odpověď (Response) na tento příkaz.

Ani příkaz, ani odpověď se nezapisují do bufferů a CPU na kartě se o nich vůbec nedozví.

Takto je možné vyžádat od logiky na kartě:

- sdělení ID autora/výrobce a modelu karty
- sdělení počtu volných bajtů ve Write Bufferu
- sdělení počtu čekajících bajtů dat v Read Bufferu
- sdělení stavu CPU na kartě
- provedení resetu CPU na kartě

# Výhody použitého řešení:

1. procesory mohou být taktovány zcela nezávisle, komunikace mezi nimi je asynchronní
2. procesory mohou mít odlišnou architekturu
3. velikosti bufferů na kartě mohou být přizpůsobeny jejímu účelu
4. není zvětšována délka lokální sběrnice, protože všechny signály jsou nově buzeny
5. každá karta může pracovat v libovolném slotu bez nutnosti rekonfigurace
6. všechny karty sdělují stav svých bufferů najednou
7. na straně hostitelského počítače jsou potřeba pouze 2 volné I/O porty
8. stav bufferů všech karet se zjistí jedinou I/O operací
9. jednoznačná identifikace typu karty
10. možnost odesílat/přijímat data dávkou, protože víme, kolik místa/dat zbývá
11. možnost zjistit stav CPU na kartě, případně i když je „zatuhlý“
12. možnost resetovat CPU na jedné kartě, přičemž ostatní tím nejsou ovlivněny

### 3. komunikace se „Slave CPU“ - procesorem na kartě

Pokud je předchozím zápisem na Control Port nastaven 7. bit (C/D) na 0 , bude signál C/D sběrnice FastPort v log.0 – režim přenosu dat.

V tomto stavu bude bajt zapsaný na Data Port vyslán na FastPort Bus a logikou na vybrané kartě uložen do jejího Write Bufferu, při čtení z Data Portu logika na kartě vybere z Read Bufferu bajt a přes Fast Port Bus ho pošle hostitelskému CPU.

Toto probíhá, dokud není Write Buffer zcela zaplněn, resp. Read Buffer zcela prázdný.

V takovém případě aktivuje logika na kartě signál /WBF, resp. /RBE . Stav těchto signálů může hostitelský CPU zjistit čtením Control/Status Portu a ví, že nemá pokračovat v zápisu, resp. ve čtení dat.

Pokud to architektura hostitelského CPU umožňuje, je možné nežádoucí zápis, resp. čtení blokovat - např. v případě procesoru Z80 je při pokusu o zápis při aktivním signálu /WBF aktivován signál /WAIT a zápis je zdržen, dokud se ve Write Bufferu neuvolní místo (totéž platí pro pokus o čtení při aktivním signálu /RBE).

Takto je možné např. v případě procesoru Z80 přenášet data nejen po 1 bajtu s průběžným ověřováním příznaků WBF/RBE, ale i dávkou pomocí instrukce pro hromadný vstup/výstup (INIR/OTIR), nebo pomocí DMA přenosu (protože jak INIR/OTIR, tak DMA řadič respektují signál /WAIT).

### 3. komunikace se „Slave CPU“ (pokračování)

Takto je možné předávat data třemi způsoby:

- 1) po jednom bajtu (před každým přenosem musíme zkontrolovat stavové bity WBF a RBE)
- 2) zjistit si od logiky na kartě velikost volného místa, resp. množství bajtů k přečtení (pomocí Command/Response komunikace), pak můžeme takové množství dat přenést najednou
- 3) ~~posílat bezhlavě data na port, protože logika na straně počítače nás přece včas zastaví~~  
- **nedoporučuje se, hrozí zamrznutí počítače**

**Každý z procesorů (hostitelský CPU i CPU na kartě) může používat libovolný způsob nezávisle na svém protějšku.**

Logika na straně počítače (Host Bridge) může využívat možnosti konkrétní platformy – např. generovat přerušení, pokud má nějaká karta připravena data pro odeslání do počítače, pokud to hardware a operační systém dané platformy dokáží zpracovat.

Tak může jedna a ta samá karta komunikovat s počítačem, který jí posílá data bajt po bajtu (nejpomalejší způsob), stejně jako s počítačem, který jí posílá data DMA přenosem (nejrychlejší způsob).

**Rozdíl je jen v rychlosti a v ničem jiném.**

### 3. komunikace se „Slave CPU“ (dokončení)

Komunikace ve vyšší vrstvě, tedy mezi procesory, se předpokládá po dávkách – možno říci paketech – podle přenosového protokolu, který se připravuje.

Tento protokol bude mít dvě části: jakési **povinné minimum**, kterému musí rozumět KAŽDÁ karta (např. podrobnější identifikace a zjištění stavu karty), a případné **volitelné rozšíření, jehož úplný popis musí autor či výrobce karty povinně zveřejnit** – což je analogie k vydání ovladačů ke kartě v PC světě (to je prakticky jediný nárok kladený na autora/výrobce karty, nepočítáme-li dodržení elementárních parametrů hardwaru – konektor, rozměry desky apod.).

# Výhody použitého řešení:

1. procesory mohou být taktovány zcela nezávisle, komunikace mezi nimi je asynchronní
2. procesory mohou mít odlišnou architekturu
3. velikosti bufferů na kartě mohou být přizpůsobeny jejímu účelu
4. není zvětšována délka lokální sběrnice, protože všechny signály jsou nově buzeny
5. každá karta může pracovat v libovolném slotu bez nutnosti rekonfigurace
6. všechny karty sdělují stav svých bufferů najednou
7. na straně hostitelského počítače jsou potřeba pouze 2 volné I/O porty
8. stav bufferů všech karet se zjistí jedinou I/O operací
9. jednoznačná identifikace typu karty
10. možnost odesílat/přijímat data dávkou, protože víme, kolik místa/dat zbývá
11. možnost zjistit stav CPU na kartě, případně i když je „zatuhlý“
12. možnost resetovat CPU na jedné kartě, přičemž ostatní tím nejsou ovlivněny
13. procesor v hostitelském počítači (a vlastně i na kartě) může použít nejvýhodnější způsob přenosu, který je pro něj dostupný (využití přerušení, DMA...)
14. kartu je možno použít na libovolné platformě (díky popisu jejího kom. protokolu)

# Fyzická realizace

## Elektrické parametry

Host Bridge je realizován 5V-tolerantním CPLD obvodem s úrovní signálů 3.3V, proto **úroveň všech signálů sběrnice FastPort je 3.3V** .

To přináší možnou výhodu do budoucna, protože **na kartě proto může být 3.3V logika, která není 5V-tolerantní** – což je případ mnoha v současnosti vyráběných obvodů.

Na pinech konektoru je vyvedeno napájení +5V a GND.

## Konektor

Je (pouze!) 20-pinový (využito 18 pinů), se standardní roztečí pinů 2,54mm, takže nenáročný na routování (při návrhu DPS).

## Rozměry desky

Budou ještě specifikovány, každopádně budou odvozeny od rozměrů PC karet.



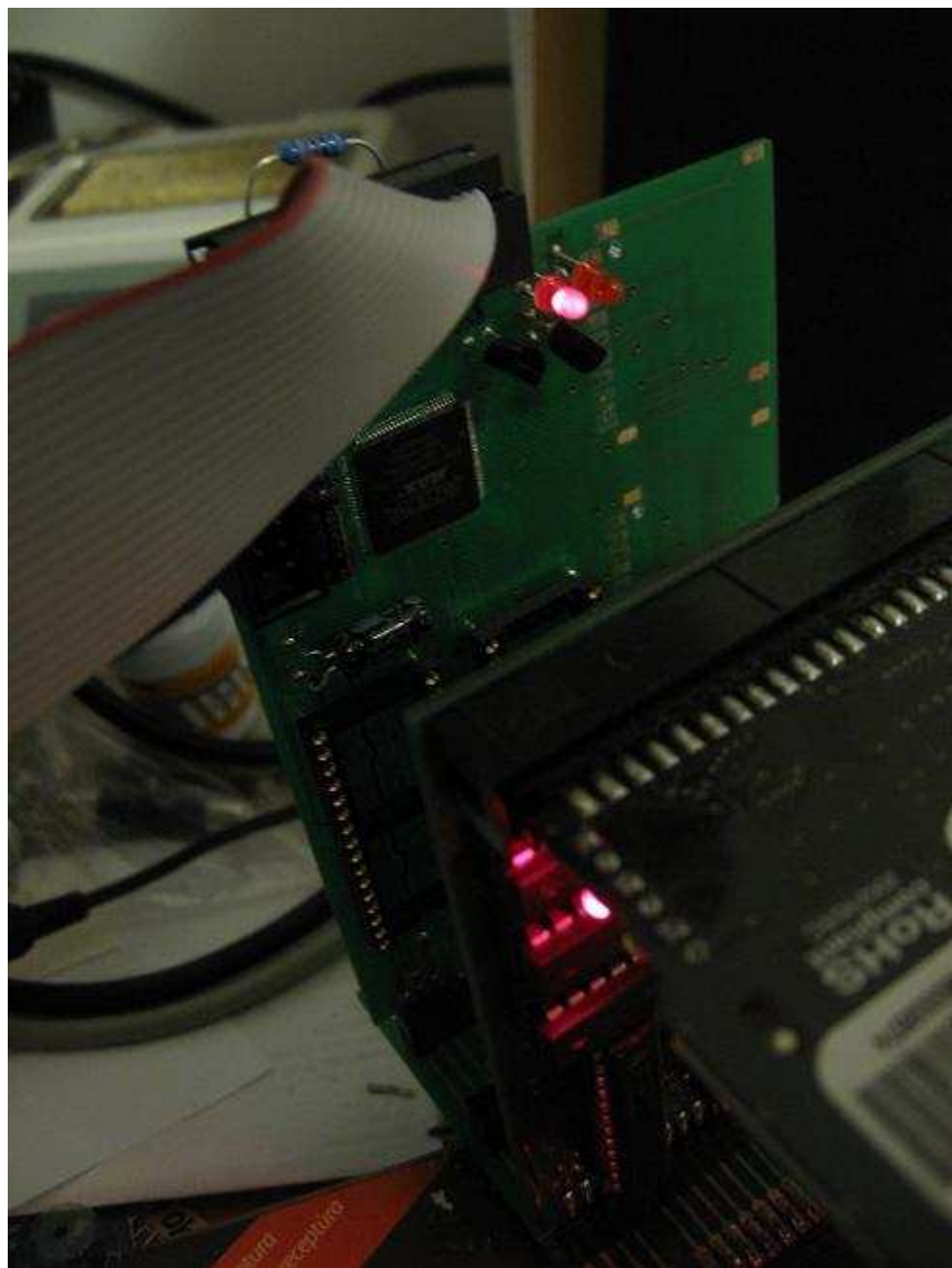
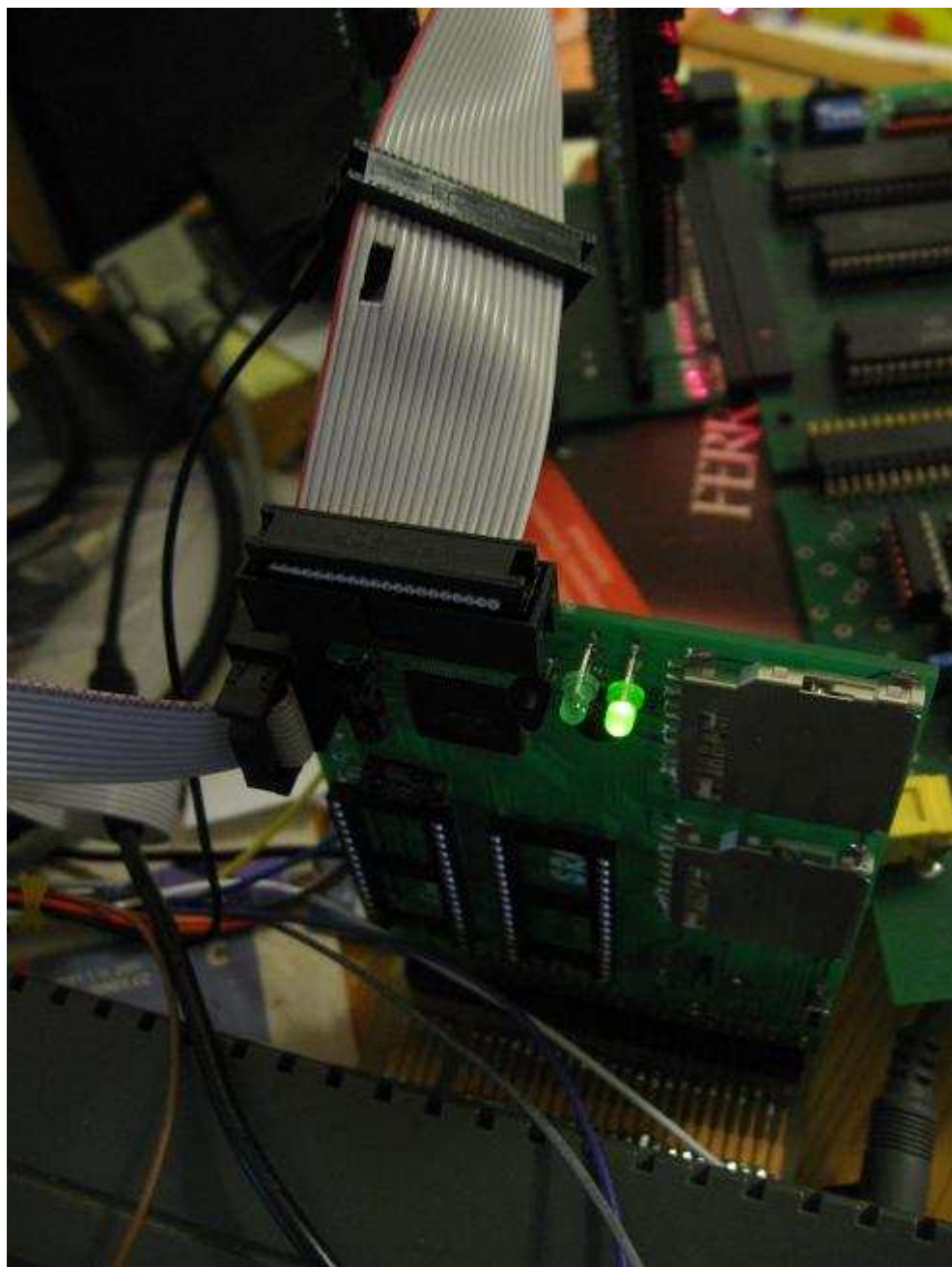
# Výhody použitého řešení:

1. procesory mohou být taktovány zcela nezávisle, komunikace mezi nimi je asynchronní
2. procesory mohou mít odlišnou architekturu
3. velikosti bufferů na kartě mohou být přizpůsobeny jejímu účelu
4. není zvětšována délka lokální sběrnice, protože všechny signály jsou nově buzeny
5. každá karta může pracovat v libovolném slotu bez nutnosti rekonfigurace
6. všechny karty sdělují stav svých bufferů najednou
7. na straně hostitelského počítače jsou potřeba pouze 2 volné I/O porty
8. stav bufferů všech karet se zjistí jedinou I/O operací
9. jednoznačná identifikace typu karty
10. možnost odesílat/přijímat data dávkou, protože víme, kolik místa/dat zbývá
11. možnost zjistit stav CPU na kartě, případně i když je „zatuhlý“
12. možnost resetovat CPU na jedné kartě, přičemž ostatní tím nejsou ovlivněny
13. procesor v hostitelském počítači (a vlastně i na kartě) může použít nejvýhodnější způsob přenosu, který je pro něj dostupný (využití přerušení, DMA...)
14. kartu je možno použít na libovolné platformě (díky popisu jejího kom. Protokolu)
15. úroveň všech signálů je 3.3V -> možnost využít na kartě moderní obvody
16. malý konektor s nízkým počtem pinů

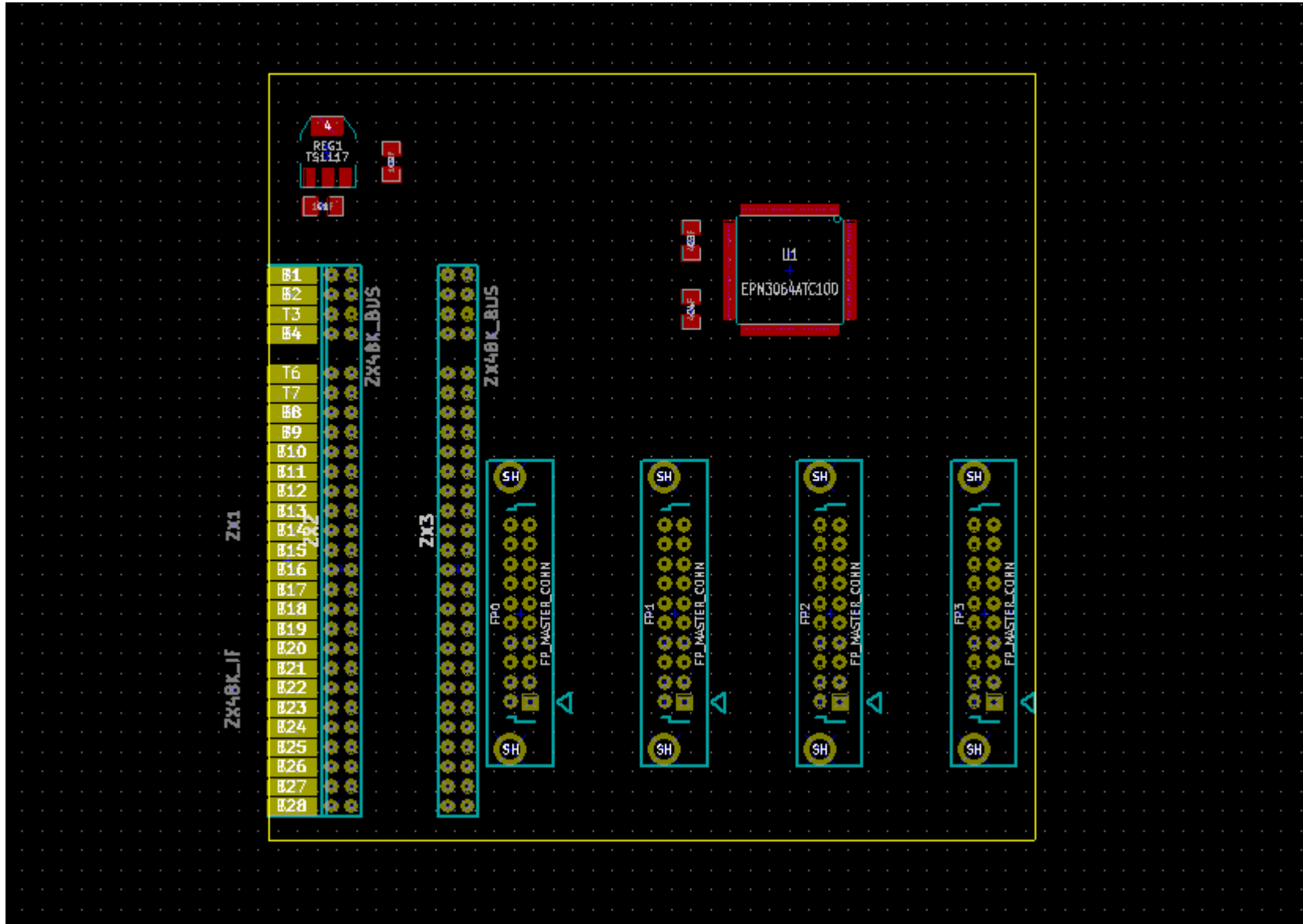
# Testování

Byl použit počítač ZX Spectrum +2 (jako hostitelský počítač), kartu s vlastním procesorem (Z80 CPU) představovalo ZX Spectrum 48K, oba s připojenou testovací kartou Uniface 1 osazenou CPLD Altera MAX3000A (EPM3128ATC). Signály FastPort Bus procházely mezi oběma kartami přes 20-žilový plochý kabel.





# příklad sběrnice desky (pro ZX Spectrum)



# Výhody sběrnice FASTPort

1. procesory mohou být taktovány zcela nezávisle, komunikace mezi nimi je asynchronní
2. procesory mohou mít odlišnou architekturu
3. velikosti bufferů na kartě mohou být přizpůsobeny jejímu účelu
4. není zvětšována délka lokální sběrnice, protože všechny signály jsou nově buzeny
5. každá karta může pracovat v libovolném slotu bez nutnosti rekonfigurace
6. všechny karty sdělují stav svých bufferů najednou
7. na straně hostitelského počítače jsou potřeba pouze 2 volné I/O porty
8. stav bufferů všech karet se zjistí jedinou I/O operací
9. jednoznačná identifikace typu karty
10. možnost odesílat/přijímat data dávkou, protože víme, kolik místa/dat zbývá
11. možnost zjistit stav CPU na kartě, případně i když je „zatuhlý“
12. možnost resetovat CPU na jedné kartě, přičemž ostatní tím nejsou ovlivněny
13. procesor v hostitelském počítači (a vlastně i na kartě) může použít nejvýhodnější způsob přenosu, který je pro něj dostupný (využití přerušení, DMA...)
14. kartu je možno použít na libovolné platformě (díky popisu jejího kom. Protokolu)
15. úroveň všech signálů je 3.3V -> možnost využít na kartě moderní obvody
16. malý konektor s nízkým počtem pinů

<http://jiiira.ajetaci.cz/FastPort/>