Vysoké učení technické v Brně
Fakulta informačních technologií
ITP – Technika personálních počítačů
Analýza komunikácie na zbernici PCI Express

Úloha č.:
3.

Zadání:

1. Oboznámte sa so zbernicou PCI Express (spôsob komunikácie zariadení na zbernici, vrstvový model a typy transakcií).
2. Oboznámte sa so zariadením pre analýzu zbernice PCI Express a potrebným ovládacím programom
3. Oboznámte sa so spôsobom komunikácie s kartou ML555 pomocou príkazu csbus
4. Oboznámte sa s adresovým priestorom firmware na čipe FPGA
5. Zachyťte komunikáciu (32 bitové čítanie, 32 bitový zápis, kontinuálne čítanie/zápis pomocou DMA) na zbernici PCI Express

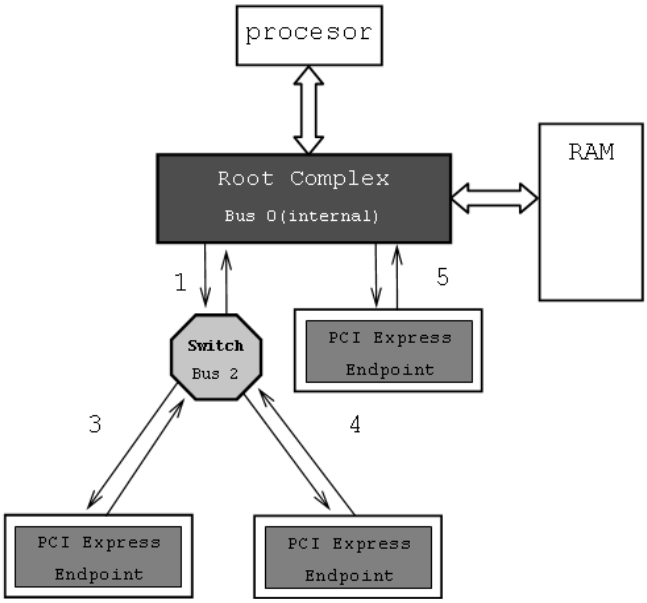
PCI Express

PCI Express je v súčasnej dobe poslednou a najmodernejšou zbernicou z rodiny PCI. Oproti predchádzajúcim typom PCI a PCI-X, tato zbernica nie je založená na širokých paralelných zberniciach, ale na vysokorychlostních plne-duplexných sériových linkách. V jeden okamžik je teda možné komunikovať na rýchlosti 2.5 Gb/s (PCI Express špecifikácia 1.0a), oboma smermi je to teda celkom 5 Gb/s. Tieto plne duplexné linky je navyše možné radiť paralelne vedľa seba a vytvárať postupne zbernicu s vyššou prenosovou kapacitou (škálovateľnosť). Linky sa postupne, podľa toho aké sú široké, označujú *x1, x2, x4, x8, x16 a x32* s priepustnosťou *0.5, 1, 2, 4, 6, 8 a 16 GB/s* (hodnoty sú uvedené už bez réžie kódovania 8/10 – viď kapitola o fyzickej vrstve).

Všetka komunikácia na zbernici PCI Express prebieha pomocou paketov. Pokiaľ je spoj medzi dvoma zariadeniami tvorený viac ako jednou linkou, sú dáta paketu rovnomerne rozosielané do jednotlivých liniek. V okamžiku keď sa PCI Express zariadenie pripojí do systému, musí sa s druhou stranou komunikačného bodu dohodnúť na spôsobe komunikácie, tzn. počtu liniek, rýchlosti a pod. Z toho dôvodu musí každé PCI Express zariadenie povinne podporovať aspoň komunikáciu po jedinej linke.

Architektúra

Základná architektúra systémovej zbernice PCI Express je znázornená na Obrázku 1. Architektúra má stromovú topológiu, kde koreňovým uzlom je *Root Complex*. Tento prvok zastáva funkciu North a South Bridge a taktiež prepojuje prvky s požiadavkou na veľmi vysokú priepustnosť ako je procesor a RAM. Prvky v listových uzloch stromu sú označované ako *Endpoint* a reprezentujú jednotlivé koncové zariadenia. Poslednou časťou topológie sú prepínače, tzv. *Switch* komponenty, ktoré tvoria hierarchiu stromu a smerujú transakčné pakety do jednotlivých vetiev.



Každé zariadenie má v rámci celkovej topológie priradenú jednoznačnú identifikáciu rovnakým spôsobom ako majú tiež zariadenia na zbernici PCI a PCI-X. Táto identifikácia je tvorená trojicou číslic:

1. *Bus Number:* udáva číslo zbernice v systému PCI. Zbernicou sa pritom rozumie vždy samostatný segment, ktorý je oddelený cez *Bridge* alebo *Switch*. Zbernice typu PCI podporujú maximálna 256 rôznych zberníc.
2. *Device Number:* udáva číslo zariadenia pripojené v rámci jednej zbernice. Keďže sú na zbernici PCI Express iba point-to-point spoje, tak na každej linke sú iba dve zariadenia s označením 0 a 1.
3. *Function Number:* udáva číslo subsystému v rámci jedného PCI zariadenia. Môžu napríklad existovať zariadenia, ktoré majú dva subsystémy, jeden pre spracovanie obrazu a druhý v podobe sieťového rozhrania. Každé zariadenie môže mať najviac 8 funkcií.

Priradenie tejto jednoznačnej identifikácie prebieha vždy v inicializačnej fáze zbernice po resetu alebo zapnutí systému a samotné číslovanie prebieha smerom do hĺbky stromu zľava. Zbernica číslo 0 je umiestnená vnútri *Root Complex* prvku. Prvá linka zľava je označená ako 1. Vnútri *Switch* komponenty je započítaná vždy jedna virtuálna zbernica, v príklade na Obrázku 1 je to zbernica číslo 2. Najľavejšia linka komponenty *Switch* je označená hodnotou 3 atď. až pokiaľ nie je ohodnotená celá topológia.

Typy transakcií na zbernici

Každá transakcia na zbernici PCI Express reprezentuje tok dát zložený z jedného alebo viacerých paketov o maximálnej veľkosti 4kB (historická veľkosť stránky). Uzol, ktorý transakciu inicializuje nazývame *Requester,* oproti tomu uzol, ktorý transakciu prijíma nazývame *Completer.* Toto názvoslovie je zhodné s PCI-X, kde bol taktiež ako i tu implementovaný model rozdelených transakcií *(tzv. split transaction).* Komunikácia môže prebiehať buďto medzi *Root* a *Endpointom*, alebo medzi dvoma *Endpointmi* navzájom (peer-to-peer). Všetky transakcie je možné rozdeliť do dvoch základných typov:

1. *Posted* – Ide o zápisové transakcie, v ktorých *Completer* nezasiela *Requesteru* žiadne potvrdenie o vykonaní transakcie. Ak nastane prípad kedy by *Completer* nemohol prijať zasielané dáta, *Requester* sa toto bez generovania ďalšej transakcie nedozvie.
2. *Non-posted* – Naopak v tomto prípade ide o transakcie, kedy *Requester* požiada o čítanie alebo i zápis dát a *Completer* mu vždy vráti odpoveď minimálne o stavu vykonanej transakcie – tzv. *Competition transakciu.* Tato transakcia môže ale nemusí obsahovať nejaké dáta. Aby *Requester* mohol rozlíšiť, ktorá odpoveď patrí ktorému požiadavku, je vždy súčasťou požiadavky i odpovede položka s označením *TAG* (viď kapitola o transakčnej vrstve). Pokiaľ nastala chyba, bude v odpovedi zaslaný i príznak chyby.

Existuje niekoľko typov transakcii, ktoré sú *Posted* i *Non-Posted* typu. Zoznam všetkých typov transakcií je na Obrázku 2. Oproti predchádzajúcim typom zberníc nám pribudla transakcia typu *Message,* ktorá slúži k zasielaniu rôznych správ. Ich hlavnou výhodou je teda to, že už nie je potrebné mať mnoho riadiacich signálov na zbernici napr. pre signalizáciu prerušenia, pre riadenie spotreby a pod. Namiesto toho je možné jednoducho zaslať správu s príslušnou informáciou.

Transakcie typu Posted sú iba operácie typu zápisu do pamäte a správy typu Message. Ostatné typy operácií ú vždy Non-Posted vrátane IO operácii Write a Configuration Write. U posledne menovaných sa predpokladá, že menia chovanie zariadenia, a preto je požadovaná odpoveď o výsledku operácie.

Názov transakcie	Typ
Memory Read	Non-Posted
Memory Write	Posted
I/O Read	Non-Posted
I/O Write	Non-Posted
Configuration Read	Non-Posted
Configuration Write	Non-Posted
Message	Posted

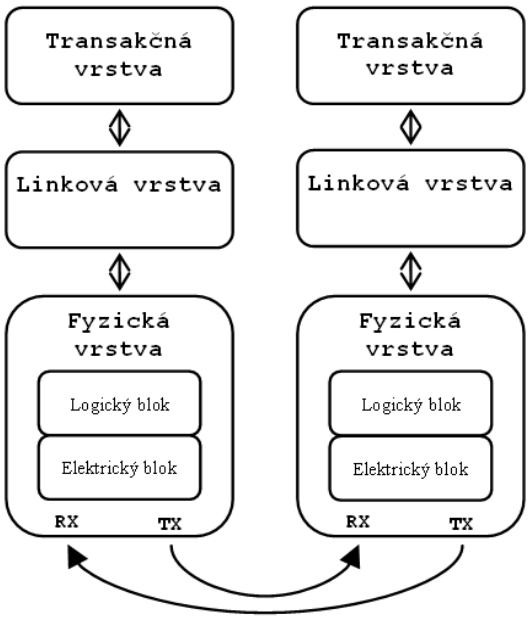
Obrázok 2: Typy transakcii a ich rozdelenie

Vrstvový model

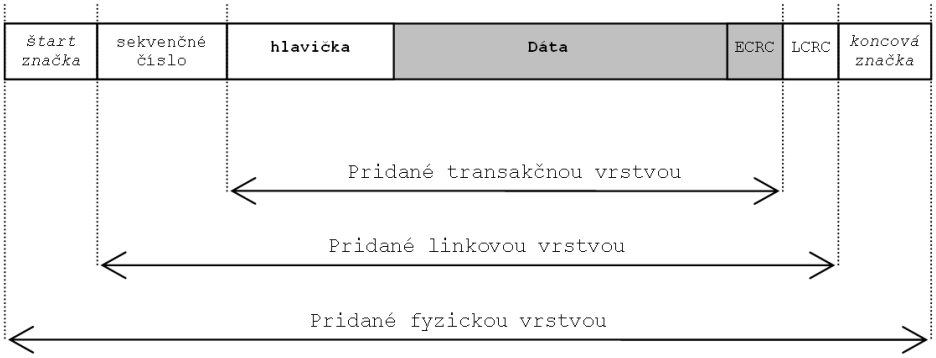
Architektúra každého PCI Express zariadenia je logicky členená do niekoľkých vrstiev podobne ako je tomu napríklad i u zariadení podľa modelu ISO/OSI. Každá z vrstiev je zodpovedná za určitú funkciu a medzi vrstvami je presne definované komunikačne rozhranie. Činnosť každej vrstvy je možné rozdeliť do dvoch osobitných častí (TX – vysielanie a RX – príjem).

Vrstvový model PCI Express (Obrázok 3) sa skladá z:

1. *Fyzická vrstva* – realizuje prenos dát po linke na najnižšej úrovni. Zahŕňa digitálnu aj analógovú (elektrickú) časť.
2. *Linková vrstva* – stará sa o spoľahlivý prenos paketov medzi dvoma susednými bodmi zbernice.
3. *Transakčná vrstva* – riadi prenos paketov medzi ľubovoľnými uzlami na najvyššej úrovni.



Obrázok 3: Vrstvový model PCI Express



Obrázok 4: Paket na zbernici PCI Express (tmavšie časti sú voľiteľné)

Fyzická vrstva

Realizuje prenos paketov na fyzickej vrstve (logická i elektrická – analógová časť). Podstatnou funkciou fyzickej vrstvy je i inicializácia a trénovanie linky, ktoré prebieha automatiky bez účasti softvéru alebo jádra PCI zariadenia.

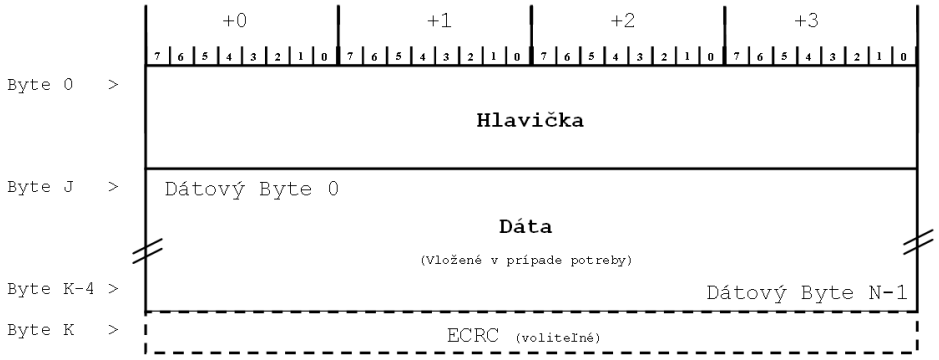
Linková vrstva

Hlavnou úlohou tejto vrstvy je zaistiť integritu dát prenesených cez jednu linku. Tá sa zaisťuje tým, že k dátam paketu, ktoré prišli z transakčnej vrstvy, pripojuje sekvenčné číslo a nad počíta CRC kód (tzv. LCRC). Sekvenčné číslo sa pripája na začiatok a LCRC na koniec paketu ako je znázornené na Obrázku 4. Paket je v takejto podobe zaslaný do fyzickej vrstvy, no zároveň je uložený i do tzv. Reply Bufferu.

Pakety okrem toho, že nesmú byť porušené, nesmú zároveň prichádzať mimo poradia. Pokiaľ je toto v poriadku, vysiela sa odosielateľovi ACK (Acknowledge) paket s rovnakým sekvenčným číslom, ako potvrdenie o správnom príjme, a vysielacia strana tak vie, že je možné paket s daným číslom uvoľniť z Reply Bufferu. Pokiaľ je však zistená chyba (nesprávne LCRC alebo mimo poradia), odosiela sa NACK (No Acknowledge) paket. Linková vrstva teda okrem bežných dátových paketov generuje i pakety riadiacich informácií (tzv. DLLP – Data Link Layer Packet). Tieto sú výrazne jednoduchšie (neobsahujú adresy) a slúžia pre potvrdzovací protokol (ACK, NACK), ďalej pre riadenie toku (Flow Control – informácie o dostupnom mieste v pamäťových bufferoch, ktorú si vymieňajú jednotlivé strany medzi sebou) a taktiež pre riadenie spotreby zariadenia (Power Management). Tieto pakety sú platné len v rámci jednej linky a teda nie sú smerované cez prepínače. Generujú sa automaticky a nezávisle od softvéru a jádra PCI zariadenia.

Transakčná vrstva

Transakčná vrstva sa primárne stará o zabezpečenie prenosu dát v podobe transakčných paketov (tzv. TLP – Transaction Layer Packet) medzi jednotlivými PCI zariadeniami. Táto vrstva taktiež zabezpečuje smerovanie TLP cez prepínače na zbernici.



Obrázok 5: Generické schéma paketu transakčnej vrstvy

Položka hlavičky	Pozícia v hlavičke paketu	Požitie položky
Length [9:0]	Byte 3, Bit 0-7 Byte 2, Bit 0-1	Veľkosť dát TLP v DW. Kódovanie: $00\ 0000\ 0001_b$ = 1DW $00\ 0000\ 0010_b$ = 2DW … $11\ 1111\ 1111_b$ = 1023 DW $00\ 0000\ 0000_b$ = 1024 DW
Attr *(Attributes)*	Byte 5, Bit 4-5	Bit 5 = *Relaxed Ordering* Ak je nastavené, využíva sa model usporiadaných transakcií podľa PCI-X. Ak nastavené nie je, používa sa striktný model usporiadania transakcii podľa PCI. Bit 6 = *No Snoop* Ak je nastavené, systém nie je nútený dodržiavať cache koherenciu. Inak dodržiava prísne model koherencie PCI.
EP *(Poisoned Data)*	Byte 2, Bit 6	Ak je nastavené, indikuje, že dáta v pakete nie sú v poriadku, i keď transakciu možno ukončiť normálne.
TD *(TLP Digest Field* Present)	Byte 2, Bit 7	Ak je nastavené, indikuje prítomnosť tzv. *TLP digest (ECRC).*
TC *(Traffic Class)*	Byte 1, Bit 4-6	Určuje Traffic Class: 000_b = *TC0* (predvolené) … 111_b = *TC7*
Type [4:0]	Byte 0, Bit 0-4	Spolu s *Fmt* určuje typ transakcie v TLP (viď tabuľka č.4)
Fmt [1:0] *(Format)*	Byte 0, Bit 5-6	Určuje veľkosť hlavičky, informuje či sú súčasťou paketu dáta a doplňuje typ transakcie (viď tabuľka č.4). 00_b = hlavička o veľkosti 3DW, TLP bez dát 01_b = hlavička o veľkosti 4DW, TLP bez dát 10_b = hlavička o veľkosti 3DW,TLP s dátami 11_b = hlavička o veľkosti 4DW,TLP s dátami
First DW **Byte Enables**	Byte 7, Bit 0-3	Bitová mapa platnosti prvého dátového bytu paketu. Hodnoty: Bit 3 = 1, byte 3 v prvom DW je platný Bit 2 = 1, byte 2 v prvom DW je platný Bit 1 = 1, byte 1 v prvom DW je platný Bit 0 = 1, byte 0 v prvom DW je platný
Last DW **Byte Enables**	Byte 7, Bit 4-7	Bitová mapa platnosti posledného dátového bytu paketu. Hodnoty: Bit 3 = 1, byte 3 v poslednom DW je platný Bit 2 = 1, byte 2 v poslednom DW je platný Bit 1 = 1, byte 1 v poslednom DW je platný Bit 0 = 1, byte 0 v poslednom DW je platný

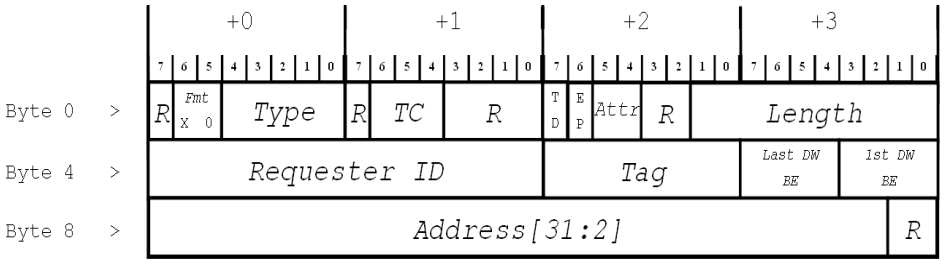
Obrázok 6: Generické položky hlavičky TLP (nastaviť znamená zapísať 1)

TLP	Fmt[1:0]	Type[4:0]
Memory Read Request (MRd)	00 = 3DW, no data 01 = 4DW, no data	0 0000
Memory Read Lock Request (MRdLk)	00 = 3DW, no data 01 = 4DW, no data	0 0001
Memory Write Request (MWr)	10 = 3DW, w/ data 11 = 4DW, w/ data	0 0000
IO Read Request (IORd)	00 = 3DW, no data	00010
IO Write Request (IOWr)	10 = 3DW, w/ data	0 0010
Config Type 0 Read Request (CfgRd0)	00 = 3DW, no data	0 0100
Config Type 0 Write Request (CfgWr0)	10 = 3DW, w/ data	0 0100
Config Type 1 Read Request (CfgRd1)	00 = 3DW, no data	0 0101
Config Type 1 Write Request (CfgWr1)	10 = 3DW, w/ data	0 0101
Message Request (Msg)	01 = 4DW, no data	1 0rr*
Message Request W/Data (MsgD)	11 = 4DW, w/ data	1 0rr*
Completion (Cpl)	00 = 3DW, no data	0 1010
Completion W/Data (CplD)	10 = 3DW, w/ data	0 1010
Completion-Locked (CplLk)	00 = 3DW, no data	0 1011
Completion W/Data (CplDLk)	10 = 3DW, w/ data	0 1011

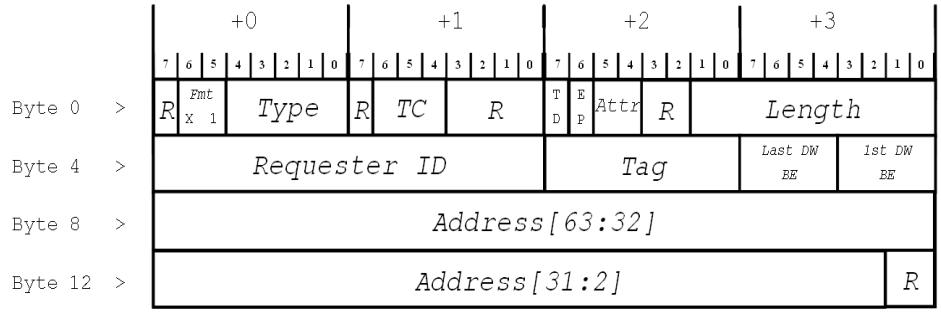
Obrázok 7: Veľkosť hlavičky paketu a konkrétny typ TLP

Pamäťové transakcie

PCI Expres pamäťové transakcie zahŕňajú dve triedy: Read Request/Completion a Write Request. Na Obrázku 9 a Obrázku 10 je vidieť rozdielne použitie týchto transakcií s 3 DW a 4 DW hlavičkou. TLP s 3DW hlavičkou (32 bitová adresa) musí byť použitý striktne pre pamäťové operácie zasahujúce do 4 GB pamäťového priestoru. 4 DW hlavičky sú naopak použité u transakcií presahujúcich 4 GB (je použitá 64 bitová adresa).



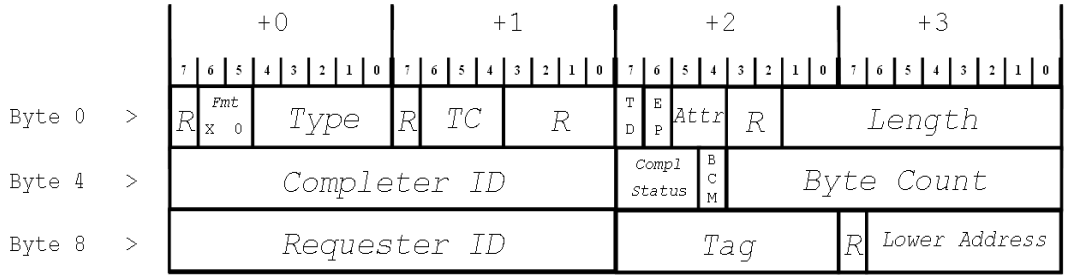
Obrázok 8: Formát hlavičky pamäťovej 3DW transakcie (do 4GB)



Obrázok 8: Formát hlavičky pamäťovej 4DW transakcie (nad 4GB)

Completion transakcie

Mnoho polí hlavičky TLP v transakcii typu Competition musí mať rovnaké hodnoty ako mala požiadavka. Sú to položky Traffic Class, Attributes a pôvodné Requester ID, ktoré je použité pre smerovanie odpovede. Formát hlavičky paketu je na Obrázku 12:

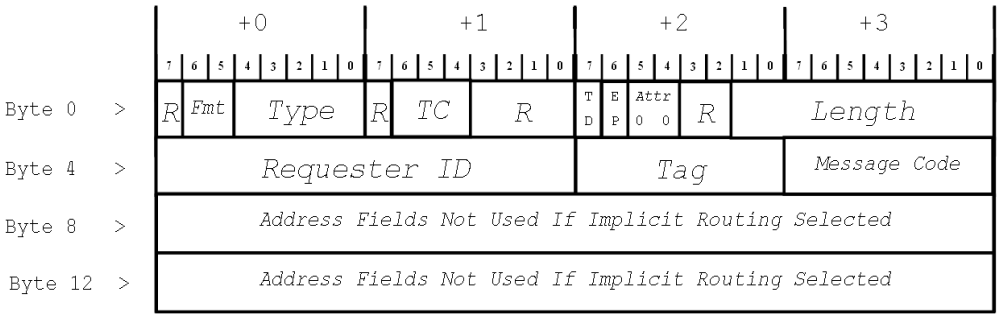


Obrázok 9: Formát hlavičky completion/dokončovacej transakcie

Ako je na tomto obrázku vidieť, pribudlo mnoho doposiaľ nedefinovaných položiek. Prvou z nich je 3 bitový Compl. Status (byte 6, bit 5-7), ktorý informuje o stave výsledku žiadosti. Jedinou správnou návratovou hodnotou je 000_b (SC - Successful Completion). Pre ostatné kódy viď odbornejšia literatúra. Ďalšou novou položkou je Byte Count (byte 7, bit 0-7 a byte 6, bit 0-3), definujúci počet zostávajúcich bytov do dokončenia celej odpovede na požiadavku. Odpoveď na ľubovoľnú Non-Posted transakciu totiž nemusí prísť v jednej samostatnej transakcii. Políčko BCM (byte 6, bit 4) je nastavované iba od PCI-X zariadení a indikuje, že predchádzajúca položka počtu zostávajúcich bytov reflektuje prvé prijaté dáta, a nie počet zostávajúcich, ešte neprenesených bytov transakcie. Posledným ešte nespomenutým políčkom je Lower Address (byte 11, bit 0-6), ktoré udáva spodných 7 bitov adresy dát navrátených čítaním. Hodnota je používaná pre zistenie adresy ďalšej odpovede.

Message transakcie

Transakcie typu Message nahradili v novej PCI Express zbernici mnoho z prerušovacích, chybových a signálov riadenia spotreby u skorších zberníc. Všetky transakcie tohto typu používajú 4 DW formát hlavičky. Ako bude vysvetlené, smerované sú na základe adresy, ID alebo implicitne. Od toho sa odvíja i význam jednotlivých položiek zobrazených na Obrázku 13:



Obrázok 10: Formát hlavičky message transakcie (správa)

Spodné 3 bity políčka typu paketu definuje spôsob smerovania. Toto môže byť smerovanie do Root Complex (000_b), smerovanie na základe adresy (001_b), smerovanie podľa ID (010_b), Broadcast (011_b), lokálna správa končiaca u prijímača (100_b) alebo zber a smerovanie k Root Complexu (101_b). Ostatné hodnoty sú rezervované pre budúce využitie. Je jasné, že položky adresy sú využité iba u smerovania na základe adresy.

Úplne novým políčkom tu je 8 bitové Message Code (byte 7, bit 0-7). Existujú správy typu Unlock Message ($0000\ 0000_b$), Power Mgmt Message ($0001\ xxxx_b$), INTx Message (0010 0xxxb) definujíce prerušenia ($0010\ 0000_b$ vyvolá INTa, $0010\ 0001_b$ vyvolá INTb apod.), Error Message ($0011\ 00xx_b$) informujúce o chybách, Hot Plug Message ($0100\ xxxx_b$), Slot Power Message ($0101\ 0000_b$), Vendor Type 0 Message ($0111\ 111x_b$) a Vendor Type 1 Message ($0111\ 1111_b$).

Ďalšie funkcie transakčnej vrstvy

Riadenie toku (Flow Control)

Zaistenie kvality služieb (QoS)

Komunikácia s kartou osadenou FPGA

Je možná po jednotlivých, 32 bitových slovách alebo pomocou DMA prenosu z pamäti FPGA (16kB zložená z BRAM) do systémovej pamäte RAM a opačne. DMA je možné zahájiť vhodným nastavením komponenty v FPGA, ktorá DMA prenos vyvolá.

Adresový priestor na FPGA

Adresa	Význam pri čítaní	Význam pri zápise
0x00000000	Globálna adresa RAM (low)	Globálna adresa RAM (low)
0x00000004	Globálna adresa RAM (high)	Globálna adresa RAM (high)
0x00000008	Lokálna adresa pamäti FPGA (100000)	Lokálna adresa pamäti FPGA (100000)
0x0000000C	Dĺžka generovanej transakcie (0-4095)	Dĺžka generovanej transakcie (0-4095)
0x00000014	Kontrolný register	Kontrolný register
0x00000018	Počet tikov do ukončenia DMA (low)	Počet DMA požiadavok (low)
0x0000001C	Počet tikov do ukončenia DMA (high)	Počet DMA požiadavok (high)
0x00100000	Počiatočná adresa pamäte FPGA	Počiatočná adresa pamäte FPGA

Na ostatné adresy nešahať!!!

Kontrolný register

0 bit → štart DMA prenosov zahájime nastavením tohto bitu do 1. Celý prenos (dĺžka transakcie x počet DMA požiadaviek) sa korektne ukončila pokiaľ je bit vynulovaný.

1 bit → typ požadovaného prenosu. Nastavenie do 1 znamená DMA prenos **FPGA → RAM**, naopak nastavenie do 0 znamená DMA prenos v smere **RAM → FPGA.**

ostatné bity → nedefinované

Nástroj csbus

Nástroj csbus umožňuje prístup ku zbernici a vnútorným prvkom vytvoreného dizajnu v FPGA. Adresy sa zadávajú priamo a sú transparentné voči adresovému priestoru pod správou operačného systému.

Zápis:

csbus <address> <value>

Čítanie:

csbus <address>

Postup práce

1. Naštudovať pozorne teoretickú časť úlohy, poznatky budú využité pri analýze komunikácie.

2. SW k analyzátoru Agilent je dostupný na stanici s OS Windows (nástroj Protocol Analyzer). K počítaču s kartou obsahujúcou FPGA je možné pripojiť sa pomocou putty. Postačí načítať uložené sedenie „itp“ (IP adresa: 147.229.177.246, login: itp, heslo: js3mitp)

3. **Zachytenie ľubovoľnej aktivity na zbernici:**
   a. spustiť Protocol analyzer
   b. v okne „Select type of connection“ ponechať všetky nastavenia bezo zmeny a odkliknúť „Start“
   c. v okne „Port Selection“ zvoliť port 101 (stačí sledovať LED na analyzátore a určiť správny port na základe pripojeného kábla), následne „OK“
   d. v menu „View“ povoliť zobrazenie okna „Hardware status“, spustiť zachytávanie – tlačítko „Start hardware capturing“, následne zastaviť

4. **Zachytenie Read a Write transakcie na zbernici:**
   a. pomocou tlačítka „T“ zobrazíte okno „Trigger Setup“
   b. pokúste sa nájsť vhodné nastavenie triggeru vytvorením stavového automatu tak, aby ste boli schopní rozumne zachytávať transakcie (zamyslite sa nad tým, čo potrebujete zachytávať a na akej vrstve, vyskúšajte si prácu s triggerom... malá pomoc: postačia 2 stavy), spustite zachytávanie
   c. generujte zápisovú transakciu na adresu 0 nástrojom csbus
   d. analyzujte zachytené dáta – všímajte si označenie typu transakcií, adresu, identifikáciu komunikujúcich strán – Requester a Completer
   e. podobne generujte čítaciu transakciu z rovnakej adresy a analyzujte zachytené dáta – porovnajte s predchádzajúcim prípadom!
   f. naštudujte manuálovú stránku nástroja lspci a na základe analyzovanej komunikácie (ID komunikujúcich strán, adresa, určte zariadenia, ktorých sa komunikácia týkala)

      analyzujte výpis programu: lspci –v –n
      zorientujte sa v súbore /usr/share/hwdata/pci.ids
      (cat /usr/share/hwdata/pci.ids | less)

5. **DMA prenosy a analýza komunikácie:**
   a. v domovskom adresári sa presuňte do zložky bin: cd ~/bin
   b. spustite zachytávanie na analyzátore, spustite nástroj na generovanie DMA prenosov: ./throughput_test –rl 1 (zahájí prenos 4096B bloku dát z FPGA do pamäte), všímajte si výpis pomocných informácií po spustení programu a nechajte si zobraziť nápovedu
   c. analyzujte komunikáciu – všímajte si inicializačné nastavenia týkajúce sa zápisu dát do registrov podľa adresového priestoru FPGA a samotný prenos dát

6. **Burst Transfers:**
   a. spustite zachytávanie a odsledujte komunikáciu pre zápisové a čítacie transakcie ./burst_test –[w|r]
   b. všímajte si rozdiely v spôsobe komunikácie na zbernici medzi týmito typmi transakcií

7. **Generovanie DMA prenosu:**
    a. na základe doterajších znalostí vytvorte skript, ktorým zahájite DMA prenos dvoch 4096B blokov z FPGA do RAM – správnou inicializáciou registrov v FPGA pomocou csbus
    b. na určenie adresy, ktorú ovládač alokoval zariadeniu môžete využiť nástroj burst_test, ktorý vypíše počiatočnú adresu
    c. nepokúšajte sa zapisovať mimo pridelený adresový priestor – viedlo by to k havárii systému!