PCT-7424C/E

programátorská příručka

historie dokumentu:		
datum	verze	změny
6.4.2012	04.2012	výchozí verze
14.5.2012	05.2012	první finální verze (firmware FPGA verze 1.3)
25.9.2012	09.2012	rozšíření firmware FPGA verze 1.4
6.12.2012	12.2012	oprava BAR1/F1 v odstavci 2.2 (změna I/O na MEM); rozšířený popis mapování registrů v odstavci 8.13
2.1.2013	01.2013	oprava několika překlepů, změna BoardID na CardID
19.3.2013	03.2013	oprava popisu kontrolního součtu v 8. kapitole, změna na kontrolní číslo
30.9.2015	09.2015	drobná zpřesnění

Výhrada odpovědnosti, autorských práv, ochranných známek a názvů:

Ačkoliv byla tato programátorská příručka vytvořena s maximální pečlivostí, nelze vyloučit, že obsahuje chyby. Domníváte-li se, že jsou některé údaje uvedeny nesprávně, neúplně nebo nepřesně, prosíme, informujte technickou podporu.

Pro případ typografických nebo obsahových chyb si TEDIA® vyhrazuje právo kdykoliv provést opravy nebo zpřesnění publikovaných informací. Právě tak produkty popsané v programátorské příručce mohou být kdykoliv revidovány se záměrem zlepšení technických parametrů nebo dosažení lepších užitných vlastností. Doporučujeme proto před každým užitím této příručky ověřit, zda není k dispozici vydání nové.

TEDIA® nezodpovídá za žádné škody vzniklé užitím této programátorské příručky nebo informací v příručce obsažených.

Programátorská příručka a její součásti jsou autorským dílem chráněným ustanovením zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

Všechna jména a názvy použité v textu mohou být chráněnými známkami nebo obchodními názvy výrobků příslušných firem.

© 1994+2015 TEDIA® spol. s r. o.

OBSAH

1. Základní informace
1.1 Úvod
1.2 Verze firmware
1.3 Technická podpora

2. PCI řadič
2.1 Úvod
2.2 Prostor PCI konfiguračních registrů
2.3 Mapování funkčních registrů

3. Funkční registry
3.1 Přehled registrů
3.2 Změny ve firmware verze 1.4
3.3 Rozdělení registrů
3.4 Registry s délkou větší než 8 bitů

4. Registry pro obsluhu čítačů
4.1 Úvod
4.2 Funkce čítačů
4.3 Registr CNTEnReg (WR)
4.4 Registr CNTClrReg (WR)
4.5 Registr CNTCWReg (WR)
4.6 Registr CNTDataReg (RD)

5. Registry pro obsluhu přerušení
5.1 Úvod
5.2 Funkce obvodů pro obsluhu přerušení
5.3 Registr INTEnReg (WR)
5.4 Registr IRQCfgReg (WR)
5.5 Registr IRQStatusReg (RD)
5.6 Registr IRQClrReg (WR)
5.7 Registr IRQEXTINReg (RD)
5.8 Registr TimerReg (WR, RD)
5.9 Schéma registrové struktury

6. Registry digitálních vstupů/výstupů a RT digitálních výstupů
6.1 Úvod
6.2 Funkce digitálních portů
6.3 Registr DINReg (RD)
6.4 Registry CNTDINReg (RD)
6.5 Registr DOUTReg (WR)
6.6 Registr RTDOUTReg (WR)
6.7 Registr RTDOUTCfgReg (WR)

7. Pomocné diagnostické registry
7.1 Úvod
7.2 Registr FreeRunCNTReg (RD)
7.3 Registr FreeRunCNTStrbReg (WR)
7.4 Registr CardIDReg (RD)
7.5 Registr FPGATypeReg (RD)
7.6 Registr FPGAVerReg (RD)

8. Servisní rozhraní

8.1 Úvod
8.2 Zapojení UARTů
8.3 Struktura povelů
8.4 Přehled povelů
8.5 Povel identifikace
8.6 Povel restart mikropočítače
8.7 Povel rekonfigurace FPGA
8.8 Povel čtení flash paměti
8.9 Povel zápis flash paměti
8.10 Povel zápis flash paměti se zpětným čtením
8.11 Rozdělení flash paměti
8.12 Propojení UARTu s mikropočítačem
8.13 Popis programové obsluhy UARTu

1. Základní informace

1.1 Úvod

Tato programátorská příručka navazuje na uživatelskou příručku karet PCT-7424C/E obsahující ...

- základní technické údaje,
- popis instalace
- a popis zapojení konektorů.

Uživatelská příručka tedy obsahuje informace postačující běžnému uživateli k nasazení karty ve spolupráci s hotovými aplikačními programy, případně k vytváření vlastních programů nad ovladači s abstraktním API (v případě Windows se jedná například o ovladač TEDIA_DAQ01).

Oproti tomu programátorská příručka obsahuje ...

- popis použitého PCI řadiče,
- popis všech funkčních registrů karty
- a popis programování na úrovni registrů.

Programátorská příručka tedy umožňuje programování nad systémovým ovladačem s API nabízejícím přímý přístup k registrům (v případě Windows se jedná o ovladač tedia_ox952), tzn. vytváření speciálních programů nebo vlastních ovladačů (například pro různé SCADA systémy nebo pro operační sytém Linux).

1.2 Verze firmware

Aktuální verze firmware v době vydání příručky:

mikropočítač - firmware:	PCIloader v.1.15	
FPGA - typ firmware:	18	(reprezentováno hodnotou 18_H)
FPGA - verze firmware:	1.4	(reprezentováno hodnotou 14_H)

Verze firmware mikropočítače nemá pro uživatele žádný zásadní význam (ovlivňuje zpravidla jen programy pro update firmware FPGA).

Typ firmware FPGA je kontrolní číslo přidělené standardnímu firmware PCT-7424C/E. Odlišné číslo představuje buď konfiguraci nesprávným firmware (například určeným pro jinou kartu) nebo zakázkovým firmware.

Verze firmware FPGA je pro uživatele poměrně významnou informací, jelikož firmware FPGA může uživatel aktualizovat samostatně.

Poznámka: *Funkce popsané v této příručce vycházejí z uvedených verzí firmware.*

1.3 Technická podpora

V případě nejasností se lze obrátit na technickou podporu výrobce:

adresa:	TEDIA spol. s r. o., Zábělská 12, 312 11 Plzeň, Česká republika
URL:	http://www.tedia.cz/podpora
e-mail:	aktuální informace najdete na adresách http://www.tedia.cz/kontakty http://www.tedia.cz/podpora
telefon:	+420 373730421 (základní číslo) +420 373730426 (technická podpora)

Doporučujeme seznámit se s užitečnými pravidly pro kontaktování technické podpory (viz výše uvedená URL).

Poznámka: *Ačkoliv byla tato programátorská příručka vytvořena s maximální pečlivostí, nelze vyloučit, že obsahuje chyby. Domníváte-li se, že jsou některé údaje uvedeny nesprávně, neúplně nebo nepřesně, prosíme, informujte technickou podporu.*

2. PCI řadič

2.1 Úvod

Karty PCT-7424C/E jsou osazeny PCI řadičem typu OXuPCI952 kompatibilním s PCI Local BUS Specification Revision 3.0.

Použitý řadič je z pohledu PCI specifikace dvojfunkční, zjednodušeně řečeno - pracuje jako dvě PCI zařízení umístěná v jediném slotu.

První PCI funkce (dále označena F0) obsahuje dva UARTy; první z nich je vyhrazen pro komunikaci s mikropočítačem karty, druhý zůstává nevyužit.

Druhá PCI funkce (dále označena F1) mapuje obvody lokání sběrnice, na kterou jsou připojeny všechny periferní obvody (digitální vstupy/výstupy, čítače apod.).

2.2 Prostor PCI konfiguračních registrů

V následující tabulce jsou vybrané registry z prostoru PCI konfiguračních registrů. Jelikož je PCI řadič dvojfunkční, tabulka obsahuje dvě sady registrů.

adresa	jméno registru	PCI funkce F0	PCI funkce F1
$01_H \div 00_H$	Vendor ID	1760_H (tzn. VID TEDIA)	1760_H (tzn. VID TEDIA)
$03_H \div 02_H$	Device ID	0214_H (PCT-7424C) 0216_H (PCT-7424E)	0215_H (PCT-7424C) 0217_H (PCT-7424E)
$0B_H \div 09_H$	Class Code	118000_H (tzn. PCI třída "other data acquisition controller")	118000_H (tzn. PCI třída "other data acquisition controller")
$13_H \div 10_H$	BAR0	UART0 (I/O, 8B, adresu přidělí BIOS)	lokální sběrnice (I/O, 256B, adresu přidělí BIOS)
$17_H \div 14_H$	BAR1	UART1 (I/O, 8B, adresu přidělí BIOS)	lokální sběrnice (MEM, 4kB, adresu přidělí BIOS)
$1B_H \div 18_H$	BAR2	nevyužito	konfigurační registry řadiče (I/O, 32B, adresu přidělí BIOS)
$1F_H \div 1C_H$	BAR3	nevyužito	konfigurační registry řadiče (MEM, 4kB, adresu přidělí BIOS)
$23_H \div 20_H$	BAR4	konfigurační registry řadiče (I/O, 32B, adresu přidělí BIOS)	nevyužito
$27_H \div 24_H$	BAR5	UART0 + UART1 + konfig. registry (MEM, 4kB, adresu přidělí BIOS)	nevyužito
$2D_H \div 2C_H$	Subsystem Vendor ID	1760_H (tzn. VID TEDIA)	1760_H (tzn. VID TEDIA)
$2F_H \div 2E_H$	Subsystem ID	0001_H	0001_H
$3C_H$	Interrupt Line	číslo IRQ kanálu (přidělí BIOS)	číslo IRQ kanálu (přidělí BIOS)
$3D_H$	Interrupt Pin	01_H (INTA)	01_H (INTA)

K čemu jsou určeny výše popsané PCI konfigurační registry ...

- Vendor ID a Device ID jsou určeny pro 100% identifikaci karty (resp. každé PCI funkce samostatně) v systému (v případě nejednoznačnosti mohou být navíc použity Subsystem Vendor ID a Subsystem ID, popř. Class Code)
- BARx jsou určeny pro zjištění přidělených prostředků, tzn. počáteční adresu bloků funkčních registrů karty
- Interrupt Line je určen pro zjištění aktuálního propojení INT signálu karty s logickým IRQ kanálem přerušení

2.3 Mapování funkčních registrů

V následující odstavcích je řešena problematika mapování funkčních registrů.

Proč jsou funkční registry mapovány duplicitně v I/O a MEM prostoru (např. PCI funkce 1, BAR0 a BAR1), jaký je mezi oběma prostory funkční rozdíl ...

Všechny registry karty jsou mapované duplicitně v obou prostorech a jejich funkční význam je identický.

Existuje zřejmě jediný důvod, proč pro přístup k funkčním registrům používat I/O prostor - jednodušší podpora v operačních systémech, resp. vývojových prostředcích, které neumožňují jednoduché 32bitové adresování MEM prostoru (např. MS-DOS). Ve všech ostatních případech je výhodnější používat MEM prostor.

Z tohoto důvodu i nízkoúrovňový ovladač pro Windows tedia_ox952 mapuje výhradně MEM prostor.

K čemu jsou určeny registry PCI funkce F0 (tzn. UARTy a konfigurační registry řadiče) ...

UART0 je vyhrazen pro komunikaci s mikropočítačem karty (mikropočítač obsluhuje servisní funkce karty, například programování firmware FPGA do flash paměti, konfiguraci FPGA atd.). UART1 je na kartě nezapojen.

Konfigurační registry řadiče obsahují životně důležité informace, například pro časování lokální sběrnice nebo mapování zdrojů přerušení. Jejich přepsání způsobí zpravidla nefunkčnost karty, případně celého počítače. Jedinou nápravou je pak restart počítače (obsah registrů je obnovován z EEPROM paměti při každém startu počítače).

K čemu jsou konfigurační registry řadiče PCI funkce F1 (tzn. BAR2 a BAR3) ...

Tyto registry mají identickou funkci jako registry BAR5 PCI funkce F0.

Jakým způsobem má ovladač, případně program přistupovat k funčním registrům karty ...

Výhradně prostřednictvím BAR0 (I/O prostor) nebo BAR1 (MEM prostor) PCI funkce F1.

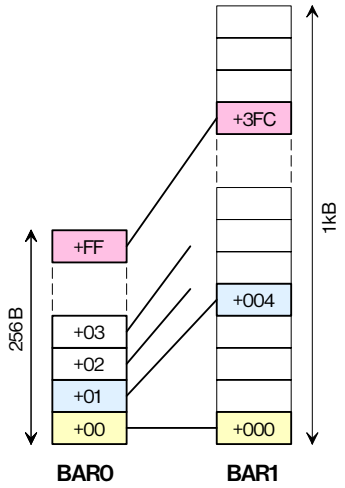
Odlišuje se nějak mapování funčních registrů v BAR0 a BAR1 PCI funkce F1 karty ...

Ačkoliv význam funkčních registrů mapovaných v I/O a MEM prostoru je identický, mapování je odlišné.

Pro všechny registrové bloky mapované v I/O prostoru (tzn. i BAR0 PCI funkce F1) předává BARx registr 16bitovou adresu prvního registru bloku a všechny další registry jsou mapovány s rozestupem +1. Prostor s velikostí 256B tak může mapovat 256 funkčních registrů. K funkčním registrům mapovaným v prostoru BAR0 PCI funkce F1 lze přistupovat výhradně s daty velikosti 1 byte.

Pro všechny registrové bloky mapované v MEM prostoru (tzn. i BAR1 PCI funkce F1) předává BARx registr 32bitovou adresu prvního registru bloku a všechny další jsou mapovány s rozestupem +4 (první registr je na adrese BARx+0, druhý registr na adrese BARx+4, atd.). Blok 256 funkčních registrů tak zabírá 1kB paměti. Funkčně lze přistupovat výhradně na adresy BARx+0, BARx+4, atd., přenášená data mohou mít velikost byte, word (platných je nejnižších osm bitů z 16) nebo doubleword (platných je nejnižších osm bitů z 32).

Funkční registry popisované v následujících kapitolách jsou specifikovány jako ofset od adresy předané registrem BAR1 PCI funkce F1. V případě použití I/O prostoru (tzn. BAR0 PCI funkce F1) stačí ofset vydělit čtyřmi. Viz obrázek vpravo.



3. Funkční registry

3.1 Přehled registrů

Následující tabulka uvádí přehled funkčních registrů implementovaných v aktuální verzi firmware (viz 1. kapitola).

ofset F1/BAR1	význam pro operaci zápis	význam pro operaci čtení
+000		DINReg
+004	DOUTReg	(zpětné čtení)
+180	IRQCfgReg	IRQStatusReg
+184	IRQClrReg	
+188	IRQEXTINReg	
+18C	INTEnReg	(zpětné čtení)
+20C/208/204/200	CNTEnReg (24 bitů)	CNTDataReg (32 bitů)
+218/214/210	CNTClrReg	
+220	CNTCWReg	
+3A0	RTDOUTReg	
+3A4	RTDOUTCfgReg	
+3B0		CNTDINReg [07..00]
+3B4		CNTDINReg [15..08]
+3B8		CNTDINReg [23..16]
+3EC/3E8/3E4/3E0	FreeRunCNTStrbReg	FreeRunCNTReg (32 bitů @ 100 kHz)
+3F0	TimerReg	TimerReg
+3F4		CardIDReg
+3F8		FPGATypeReg
+3FC		FPGAVerReg

3.2 Změny ve firmware verze 1.4

Ve firmware verze 1.4 byly doplněny tyto funkce:

- doplněna podpora digitálního vstupu EXT-IN v registrech IRQCfgReg, IRQClrReg, IRQStatusReg a IRQEXTINReg
- doplněna možnost přímého čtení stavu vstupů čítačů, tzn. registry CNTDINReg [07..00], CNTDINReg [15..08] a CNTDINReg [23..16]

3.3 Rozdělení registrů

Registry lze podle významu rozdělit do několika skupin, v předešlé tabulce jsou skupiny registrů vyznačeny barvou s následujícím významem:

bílá	registry čítačů (datové i konfigurační)
žlutá	registry související s přerušením systému
šedá	registry digitálních vstupů/výstupů, vstupů čítačů a RT digitálních výstupů
zelená	pomocné diagnostické registry

Uvedené skupiny registrů budou popsány v následujících kapitolách.

Upozornění: *Všechny neobsazené adresy jsou rezervovány a program nesmí na tyto adresy přistupovat (tzn. nesmí na ně zapisovat ani z nich číst).*

3.4 Registry s délkou větší než 8 bitů

Karta obsahuje řadu vícebytových registrů (tzn. jejichž délka je větší než 8 bitů) a jsou tedy složeny ze dvou až čtyř osmibitových registrů. Pro všechny registry platí ...

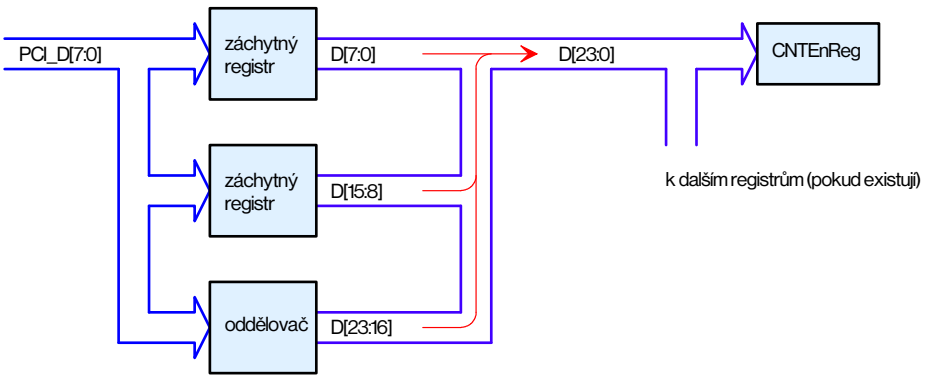
- osmibitový registr s nejnižší adresou obsahuje nejnižší bity vícebytového registru a osmibitový registr s nejvyšší adresou obsahuje nejvyšší bity vícebytového registru
- čtení i zápis osmibitových registrů musí probíhat v pořadí od registru s nejnižší adresou po registr s nejvyšší adresou
- musí být vždy čteny všechny osmibitové registry příslušné jednomu vícebytovému registru

Pro příklad postup čtení a výpočtu 32bitové hodnoty registru FreeRunCNTReg:

1. čtení	B0 = obsah registru na adrese $+3E0_H$
2. čtení	B1 = obsah registru na adrese $+3E4_H$
3. čtení	B2 = obsah registru na adrese $+3E8_H$
4. čtení	B3 = obsah registru na adrese $+3EC_H$
výpočet	výsledná hodnota = 16777216 * B3 + 65536 * B2 + 256 * B1 + B0

V případě zápisu do vícebytových registrů je potřeba vzít v úvahu, že záchytné registry nižších byte mohou být (ale nemusí) společné pro všechny vícebytové registry a je tedy potřeba vždy zapsat všechny byte a až následně přistupovat k jinému registru (je nutné ošetřit zejména v případech, kdy k registrům přistupuje více souběžných procesů).

Viz obrázek níže zobrazující řešení zápisu do 24bitových funkčních registrů; v prvním cyklu se zapisuje do záchytného registru s významem D[7:0] (např. pro funkční registr CNTEnReg jde o adresu $+200_H$), v druhém cyklu se zapisuje do záchytného registru s významem D[15:8] (pro CNTEnReg jde o adresu $+204_H$) a v třetím cyklu při zápisu dat s významem D[23:16] (pro CNTEnReg jde o adresu $+208_H$) se do odpovídajícího funkčního registru přenáší všech 24 bitů dat.



Poznámka: *Obrázek zachycuje stav aktuální verze firmware. V budoucích verzích firmware mohou být funkční registry vybaveny vlastními záchytnými registry a nelze se tedy spoléhat na uchování obsahu záchytných registrů při postupném zápisu do různých funkčních registrů.*

4. Registry pro obsluhu čítačů

4.1 Úvod

V následujících odstavcích budou popsány registry související s čítači, viz přehled v 3. kapitole.

Přehled registrů:

CNTEnReg	24bitový registr, povoluje čítání jednotlivých čítačů
CNTClrReg	24bitový registr, slouží k nulování jednotlivých čítačů (24bitový registr)
CNTCWReg	8bitový registr, slouží k zachycení aktuálního stavu vybraného čítače do CNTDataReg
CNTDataReg	32bitový registr, slouží pro přenos dat z čítačů

4.2 Funkce čítačů

Základní registrový interface karty nabízí 24 čítačů s možností ...

- spuštění a zastavení čítačů (tzn. povolení/blokování reakce na vstupní signál), povel lze provést synchronně pro více čítačů
- nulování stavu, povel lze provést synchronně pro více čítačů
- čtení (stav čítačů je zachycován a čten postupně)

Čítače PCT-7424C reagují na sestupnou hranu TTL signálu (tzn. přechod signálu z úrovně H do L), čítače PCT-7424E reagují na náběžnou hranu 24V signálu (tzn. přechod signálu z úrovně 0V do 24V).

Poznámka: *Připravované verze firmware umožní nastavení libovolné hodnoty a synchronní čtení více čítačů. Uvažováno je rovněž s podporou záznamu do FIFO paměti.*

4.3 Registr CNTEnReg (WR)

Tento registr slouží ke spuštění a zastavení čítačů, resp. povolení/blokování reakce na vstupní signál.

Registr je 24bitový a vyžaduje zápis do všech tří osmibitových registrů v pořadí nejnižší adresou počínaje; zápisem do registru s nejvyšší adresou je obsah všech 24bitů přenesen do řídicích obvodů čítačů.

Každý z 24 čítačů má přidělen jeden bit 24bitového registru CNTEnReg; bit D0 registru s nejnižší adresou řídí čítač CNT0, bit D7 registru s nejvyšší adresou řídí čítač CNT23.

Je-li odpovídající bit registru nastaven na hodnotu 0, čítač je blokován, tzn. nereaguje na vstupní signál. Je-li odpovídající bit registru nastaven na hodnotu 1, čítač zpracovává vstupní signál a s každým vstupním pulsem inkrementuje svůj stav.

Počáteční hodnota registru je nulová a čítání všech čítačů je tedy blokováno.

4.4 Registr CNTClrReg (WR)

Tento registr slouží k nulování čítačů.

Registr je 24bitový, na rozdíl však od CNTEnReg nevyžaduje zápis do všech tří osmibitových registrů v pořadí nejnižší adresou počínaje; zapisovat lze do každého z registru nezávisle a zapsaný osmibitový obsah je ihned intepretován řídicími obvodů čítačů.

Poznámka: *Z důvodu dopředné kompatibility je doporučeno zapisovat vždy do všech tří registrů v pořadí nejnižší adresou počínaje.*

Každý z 24 čítačů má přidělen jeden bit 24bitového registru CNTClrReg; bit D0 registru s nejnižší adresou řídí čítač CNT0, bit D7 registru s nejvyšší adresou řídí čítač CNT23.

Je-li odpovídající bit registru nastaven na hodnotu 0, stav čítače není modifikován. Je-li odpovídající bit registru nastaven na hodnotu 1, čítač je vynulován.

Registr má funkci automatického nulování a zápis hodnoty 1 tak nevyžaduje následný zápis hodnoty 0; registr je nastaven na hodnotu 1 na dobu 60 ns po zápisu (po tuto dobu není zpracováván příslušný vstupní signál) a následně vynulován.

4.5 Registr CNTCWReg (WR)

Tento registr slouží k zachycení aktuálního stavu vybraného čítače do CNTDataReg.

Význam zapisovaných dat je následující:

0	do CNTDataReg je zachycen aktuální stav čítače CNT0
1	do CNTDataReg je zachycen aktuální stav čítače CNT1
. . .	. . . . .
23	do CNTDataReg je zachycen aktuální stav čítače CNT23
128	do CNTDataReg je zachycen aktuální stav vstupních signálů čítačů (nejnižších 24 bitů CNTDataReg přenáší stav vstupů, nejvyšších 8 bitů je trvale nulových)

Ostatní hodnoty CNTCWReg jsou vyhrazeny pro případná rozšíření.

4.6 Registr CNTDataReg (RD)

Tento registr slouží ke čtení zachyceného stavu čítačů, resp. stavu vstupů čítačů (viz popis CNTCWReg).

5. Registry pro obsluhu přerušení

5.1 Úvod

V následujících odstavcích budou popsány registry související s obvody pro přerušení systému, viz přehled v 3. kapitole.

Přehled registrů:

INTEnReg	aktivace obvodů pro řízení signálu přerušení PCI sběrnice (INTA)
IRQCfgReg	povolení zdrojů přerušení
IRQStatusReg	příznaky zdrojů přerušení
IRQClrReg	nulování příznaků přerušení
IRQEXTINReg	aktuální stav signálu na digitálním vstupu EXT-IN
TimerReg	generátor časových značek pro periodické vyvolávání přerušení

5.2 Funkce obvodů pro obsluhu přerušení

Obvody pro obsluhu přerušení umožňují vyvolat přerušení systému jedním ze zdrojů, případně zvolenou kombinací zdrojů přerušení. Karta disponuje těmito zdroji přerušení:

Generátor časových značek

Umožňuje vyvolávat přerušení se zvolenou časovou periodou v rozsahu 1÷255 ms.

Digitální vstup EXT-IN

Umožňuje vyvolávat přerušení sestupnou hranou na signálu EXT-IN.

Pro správnou obsluhu je potřeba vzít v úvahu, že přerušení systému je vyvoláno první detekovanou událostí, tzn. přechodem obsahu IRQStatusReg registru z nulové hodnoty do nenulové. Pro další vyvolání přerušení je tedy nezbytné, aby programová obsluha přerušení zpracovala všechny požadavky detekované od vyvolání přerušení do vykonání této obsluhy a vynulovala všechny nastavené příznaky.

5.3 Registr INTEnReg (WR)

Tento registr slouží k aktivaci obvodů pro řízení signálu přerušení PCI sběrnice a současně k nulování záchytného registru generujícího signál PCI sběrnice INTA.

D7	D6	D5	D4	D3	D2	D1	D0
INTEN	RSRV						

INTEN	aktivace obvodů pro řízení INTA 0 záchytný registr generující řídicí signál INTA je trvale nulován 1 funkce záchytného registru je aktivována, tzn. karta může vyvolat přerušení systému
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

5.4 Registr IRQCfgReg (WR)

Tento registr slouží k povolení základních zdrojů přerušení.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	EXT-IN	RSRV	TIM	RSRV			

TIM	povoluje vyvolání přerušení odvozené od generátoru časových značek 0 záchytný registr navázaný na generátor časových značek je blokován 1 funkce záchytného registru je aktivována
EXT-IN	povoluje vyvolání přerušení odvozené od sestupné hrany digitálního vstupu EXT-IN 0 záchytný registr navázaný na digitální vstup EXT-IN je blokován 1 funkce záchytného registru je aktivována
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

5.5 Registr IRQStatusReg (RD)

Tento registr slouží k zjištění stavu záchytných registrů povolovaných registrem IRQCfgReg.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	EXT-IN	RSRV	TIM	RSRV			

TIM stav záchytného registru navázaného na generátor časových značek
0 registr není nastaven, tzn. od posledního nulování nedošlo k vygenerování časové značky
1 registr je nastaven, tzn. od posledního nulování došlo k vygenerování časové značky

EXT-IN stav záchytného registru navázaného na na digitální vstup EXT-IN
0 registr není nastaven, tzn. od posledního nulování nedošlo k sestupné hraně na EXT-IN
1 registr je nastaven, tzn. od posledního nulování došlo k k sestupné hraně na EXT-IN

RSRV rezerva (z důvodu dopředné kompatibility je doporučeno hodnoty ignorovat)

5.6 Registr IRQClrReg (WR)

Tento registr slouží nulování záchytných registrů povolovaných registrem IRQCfgReg.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	EXT-IN	RSRV	TIM	RSRV			

TIM nuluje stejnojmenný záchytný registr v IRQStatusReg
0 bez významu, stav záchytného registru není modifikován
1 záchytný registr je vynulován (generuje krátký puls, následný zápis 0 není vyžadován)

EXT-IN nuluje stejnojmenný záchytný registr v IRQStatusReg
0 bez významu, stav záchytného registru není modifikován
1 záchytný registr je vynulován (generuje krátký puls, následný zápis 0 není vyžadován)

RSRV rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

5.7 Registr IRQEXTINReg (RD)

Tento registr slouží k zjištění aktuálního stavu signálu na digitálním vstupu EXT-IN.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	EXT-IN	RSRV					

EXT-IN aktuální stav signálu na digitálním vstupu EXT-IN

RSRV rezerva (z důvodu dopředné kompatibility je doporučeno hodnoty ignorovat)

5.8 Registr TimerReg (WR, RD)

Tento registr slouží k ovládání generátoru časových značek určeného pro periodické vyvolávání přerušení.

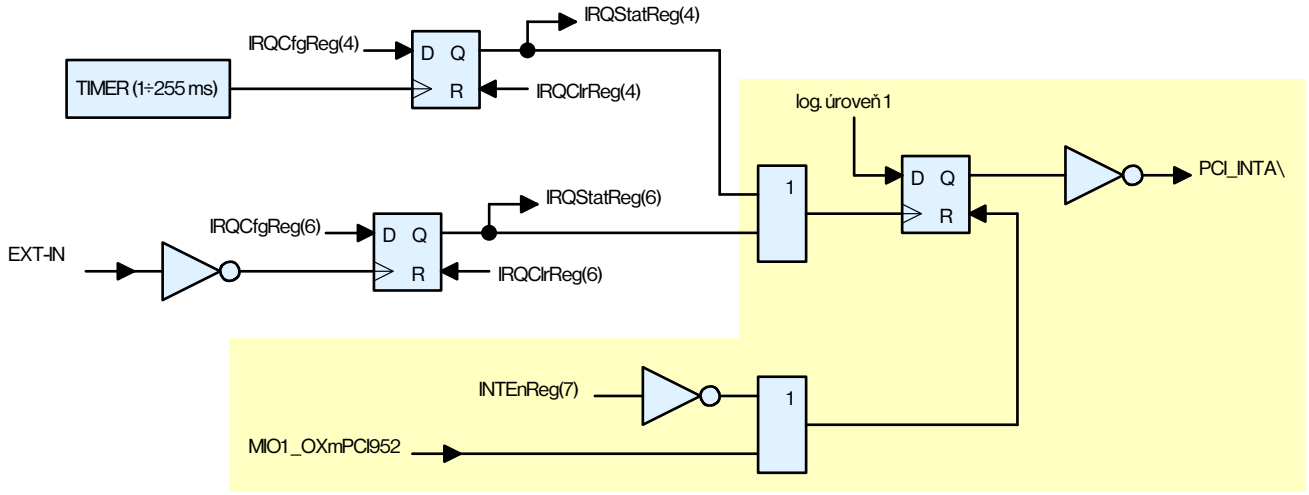
Počáteční hodnota registru je nulová a generátor časových značek je blokován. Zápisem nenulové hodnoty je generátor odstartován, perioda je definovaná zapsanou hodnotou v milisekundách. Zápisem nulové hodnoty je generátor zastaven.

Registr má význam i pro čtení (je čtena aktuální hodnota čítače inkrementovaného od nuly každou milisekundu až do zadané hodnoty zmenšené o jedna).

Pro příklad, zápisem hodnoty 100 bude první přerušení vyvoláno 100 ms po zápisu do registru a následně vždy po dalších 100 ms. Čtením budou získávány hodnoty 0, 1, ... , 98, 99, 0, 1, ..., přerušení je vyvoláno v okamžik přechodu z 99 na 0.

5.9 Schéma registrové struktury

Na níže uvedeném obrázku je pro lepší porozumění souvislostí zakresleno schéma registrové struktury.



Žlutě vyznačené obvody musejí být řízeny v rámci ISR, ostatní mohou být obsluhovány v rámci uživatelského programu, aplikačního ovladače (zpravidla DLL v případě Windows) nebo také v rámci ISR.

Signál PCI řadiče MIO1_OXmPCI952 je řešen identicky u všech karet TEDIA s tímto PCI řadičem a umožňuje unifikovat ISR obsluhu karet. Signál je po startu karty v logické úrovni 0 a neblokuje tedy funkci signálu INTEnReg(7), pro vytváření specifického uživatelského ovladače lze tedy využít i řízení pomocí INTEnReg(7).

Signál MIO1_OXmPCI952 využívá systémový ovladač tedia_ox952 pro Windows. V rámci aplikačního ovladače je potřeba při konfiguraci zdrojů přerušení nastavit registr INTEnReg(7) do logické úrovně 1 a při ukončení požadavku o podporu přerušení (zpravidla při ukončování programu) pak registr nastavit zpět do logické úrovně 0; vlastní deaktivaci signálu PCI_INTA\ pomocí signálu MIO1_OXmPCI952 provádí ISR v kernel části ovladače automaticky.

6. Registry digitálních vstupů/výstupů a RT digitálních výstupů

6.1 Úvod

V následujících odstavcích budou popsány registry související s digitálními vstupy a výstupy, viz přehled v 3. kapitole.

Registry lze dále rozdělit na skupinu datových registrů

DINReg	registr vstupního portu
DOUTReg	registr výstupního portu
RTDOUTReg	registr RT výstupního portu s možností přesměrování na interní signály karty
CNTDINReg	tři registry pro čtení vstupních signálů čítačů

a konfigurační registry

RTDOUTCfgReg	konfigurační registr RT výstupního portu

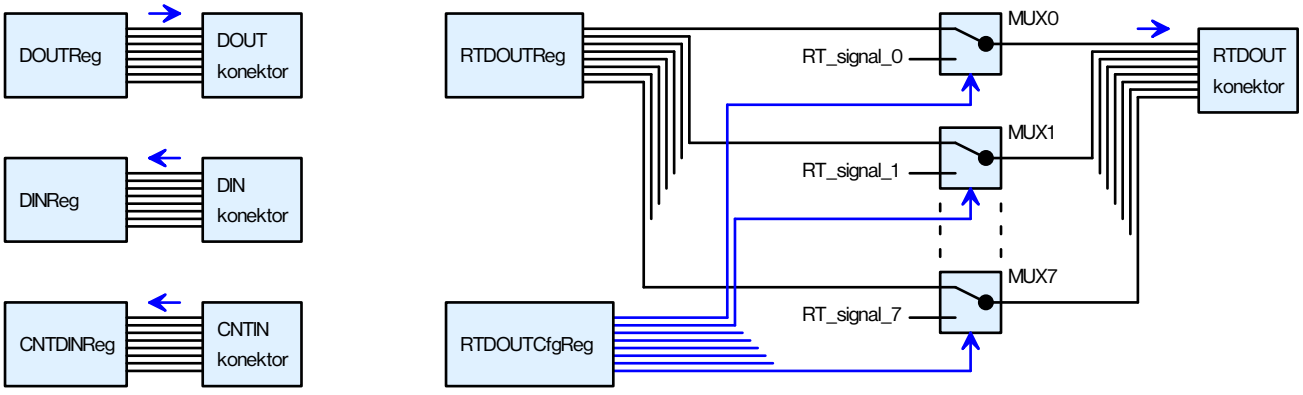
6.2 Funkce digitálních portů

Digitální porty jsou nejjednodušším I/O portem karty. Aktuální stavy vstupů lze zjistit čtením jediného registru (bez vlivu na ostatní funkce karty), výstupy lze ovládat zápisem do registru (rovněž bez vlivu na ostatní funkce karty).

Mezi tyto registry byly zařazeny i tři registry CNTDINReg určené pro čtení vstupních signálů čítačů, jelikož jejich význam je zcela analogický pro aplikace, kdy mají vstupy čítačů sloužit jako běžné digitální vstupy.

RT digitální výstupy se od standardních výstupů odlišují sdílením fyzického výstupu signálem digitálního výstupního portu a vnitřním signálem karty.

Další podrobnosti jsou patrné z obrázku níže.



6.3 Registr DINReg (RD)

Tento registr slouží ke čtení stavu vstupního digitálního portu, každý bit registru zpřístupňuje jeden signál osmibitového digitálního portu (nejnižší bit DIN0, nejvyšší bit DIN7).

6.4 Registry CNTDINReg (RD)

Tyto tři registry slouží ke čtení stavu vstupních signálů čítačů (tzn. vstupy čítačů tedy pracují současně fungují jako běžné digitální vstupní porty).

CNTDINReg [07..00]	zpřístupňuje vstupy čítačů CNT0 (nejnižší bit registru) až CNT7 (nejvyšší bit registru)
CNTDINReg [15..08]	zpřístupňuje vstupy čítačů CNT8 (nejnižší bit registru) až CNT15 (nejvyšší bit registru)
CNTDINReg [23..16]	zpřístupňuje vstupy čítačů CNT16 (nejnižší bit registru) až CNT23 (nejvyšší bit registru)

Poznámka: *Karta umožňuje rovněž synchronní čtení všech 24 vstupů čítačů, viz popis registru CNTCWReg.*

6.5 Registr DOUTReg (WR)

Tento registr slouží k ovládání stavu výstupního digitálního portu, každý bit registru zpřístupňuje jeden signál osmibitového digitálního portu (nejnižší bit DOUT0, nejvyšší bit DOUT7).

6.6 Registr RTDOUTReg (WR)

Tento registr slouží k ovládání stavu RT výstupního digitálního portu, každý bit registru zpřístupňuje (za předpokladu nevyužití alternativního RT signálu) jeden signál osmibitového digitálního portu (nejnižší bit RT-DOUT0, nejvyšší bit RT-DOUT7).

Alternativní RT signály jsou aktivovány pomocí registru RTDOUTCfgReg pro každý bit, resp. každý signál samostatně.

6.7 Registr RTDOUTCfgReg (WR)

Tento registr nebyl doposud implementován, žádné alternativní signály tedy nelze aktivovat.

7. Pomocné diagnostické registry

7.1 Úvod

V následujících odstavcích budou popsány pomocné diagnostické registry, viz přehled v 3. kapitole.

Přehled registrů:

FreeRunCNTReg	registr pro čtení stavu volnoběžného čítače 32 bitů @ 100 kHz
FreeRunCNTStrbReg	registr pro zachycení stavu volnoběžného čítače do FreeRunCNTReg
CardIDReg	registr pro čtení stavu DIP spínače (umožňuje identifikovat až 4 karty stejného typu)
FPGATypeReg	konstanta označující typ firmware FPGA (standardní, zakázkový apod.)
FPGAVerReg	konstanta označující verzi firmware FPGA

Diagnostické registry jsou určeny zpravidla pro servisní funkce, může je však používat každý program pro jednoduchou detekci verze firmware FPGA, doby běhu firmware FPGA, jako přesnou časovou značku využívanou v programu.

Diagnostické registry jsou implementovány ve všech nových kartách s FPGA, resp. nových verzí firmware.

7.2 Registr FreeRunCNTReg (RD)

Tento registr zpřístupňuje hodnotu 32bitového čítače inkrementovaného frekvencí 100 kHz (k přetečení tedy dochází každých cca 12 hodin) z nulového stavu od okamžiku konfigurace FPGA provedené po zapnutí počítače nebo po softwarovém spuštění konfigurace. Čítač nelze zastavit ani jeho hodnotu přepsat, lze jej výhradně číst.

Čítač má význam přesné časové značky pro obecné užití.

7.3 Registr FreeRunCNTStrbReg (WR)

Tento registr slouží k zachycení aktuálního stavu volnoběžného čítače do registru FreeRunCNTReg.

K zachycení stavu čítače do FreeRunCNTReg dojde zápisem do FreeRunCNTStrbReg, obsah dat je nevýznamný.

7.4 Registr CardIDReg (RD)

Tento registr zpřístupňuje stav dvousegmentového DIP spínače a umožňuje tak identifikovat až 4 karty stejného typu instalované v systému. Data jsou přenášena na nejnižších dvou bitech, horních šest bitů je trvale nulových.

7.5 Registr FPGATypeReg (RD)

Tento registr zpřístupňuje konstantu označující typ firmware FPGA v rozsahu 0 až 255.

Stejná hodnota je obsažena v identifikačních datech přenášených servisním rozhraním.

Poznámka: *Hodnota typu standardního firmware karet PCT-7424C/E je uvedena v 1. kapitole.*

7.6 Registr FPGAVerReg (RD)

Tento registr zpřístupňuje konstantu označující verzi firmware FPGA.

Hodnota registru v rozsahu 0 až 255 definuje verzi v rozsahu 0.0 do F.F (tzn. je zobrazena v HEX tvaru s desetinnou čárkou). Stejná hodnota je obsažena v identifikačních datech přenášených servisním rozhraním.

Poznámka: *Hodnota aktuální verze standardního firmware karet PCT-7424C/E je uvedena v 1. kapitole.*

8. Servisní rozhraní

8.1 Úvod

V následujících odstavcích bude popsán komunikační protokol servisního rozhraní.

Ačkoliv servisní rozhraní je primárně určeno pro programování flash paměti, má implementovánu jednu uživatelsky použitelnou funkci pro zjištění identifikačních a stavových údajů karty (výrobní číslo, verze firmware, stav konfigurace FPGA, "CardID" číslo definované DIP spínačem umožňující identifikovat v systému až 4 karty stejného typu). Ostatní funkce by však měly být použity jen v odůvodněných případech a výhradně servisními programy.

8.2 Zapojení UARTů

Řadič OXuPCI952 obsahuje dva UARTy typu 16C950 číslované jako UART0 a UART1 (viz popis v 2. kapitole), pro servisní funkce je použit UART0 (je propojen s mikropočítačem karty), UART1 je nezapojen.

Komunikace s mikropočítačem probíhá z pohledu konfigurace pro standardní oscilátor 1,8432 MHz rychlostí 57,6 kBd, ve skutečnosti však přenosy probíhají výrazně rychleji (je použit oscilátor s frekvencí 25 MHz, data jsou tedy přenášena rychlostí 781,25 kBd).

Programovat lze s obecnou znalostí registrů 16C950 (lze tedy volit FIFO režim libovolné kapacity) bez použití přerušení a s podmínkou aktivace automatického RTS/CTS režimu (brání ztrátě dat při příjmu; mikropočítač pozastaví vysílání v případě zaplnění FIFO paměti až do jejího uvolnění). Přenos probíhá s délkou znaku 8 bitů, 1 stopbitem a bez parity.

8.3 Struktura povelů

Komunikuje se metodou dotaz/odpověď.

Struktura dotazu:

start značka	tělo dotazu			stop značka
	povel	data	kontrolní číslo	
ASCII znak	1 byte	N byte	1 byte	ASCII znak
{	<--------- část kódovaná do 2*(1+N+1) ASCII HEX znaků --------->			}

Struktura odpovědi:

start značka	tělo odpovědi			stop značka
	povel	data	kontrolní číslo	
ASCII znak	1 byte	N byte	1 byte	ASCII znak
[	<--------- část kódovaná do 2*(1+N+1) ASCII HEX znaků --------->			]

Kontrolní číslo

Kontrolní číslo je stanoveno jako doplněk do hodnoty 100_H k součtu modulo 256 částí povel a data.

Příklad výpočtu kontrolního čísla:

Povel	= $1A_H$
Data	= 46_H $7D_H$ $F1_H$
Aritmetický součet	= $1A_H + 46_H + 7D_H + F1_H = 1CE_H$
Součet modulo 256	= $1A_H + 46_H + 7D_H + F1_H = CE_H$
Kontrolní číslo	= 32_H (tzn. platí $CE_H + 32_H = 100_H$)
Zpráva jako dotaz	= {1A467DF032}
Zpráva jako odpověď	= [1A467DF032]

Kódování do ASCII HEX znaků

Každý byte je rozložen do dvou ASCII HEX znaků, například číslo s hodnotou $3A_H$ je kódováno do znaků "3" a "A".

Řazení vícebytových čísel

Vícebytová čísla jsou přenášena nejnižším bytem počínaje, například 16bitové číslo s hodnotou 1234_H je kódováno do posloupnosti znaků "3", "4", "1" a "2".

8.4 Přehled povelů

Dále uvedená tabulka obsahuje přehled implementovaných povelů.

název povelu	povel	data dotazu		data odpovědi		popis
identifikace	00_H	0 B	žádná nejsou	96 B	ASCII text	vrací název zařízení, verzi firmware apod.
restart mikropočítače	01_H	0 B	žádná nejsou	0 B	žádná nejsou	restartuje řídicí mikropočítač a následuje kompletní konfigurace FPGA
rekonfigurace FPGA	02_H	0 B	žádná nejsou	0 B	žádná nejsou	vyvolá novou konfiguraci FPGA
čtení flash	$0A_H$	2 B	číslo stránky	2 B	číslo stránky	čte vybranou stránku paměti flash (flash s kapacitou 512 kB je dělena do stránek o velikosti 256 B; stránky jsou číslovány od 0 do 2047)
				256 B	data stránky	
zápis flash	$0B_H$	2 B	číslo stránky	2 B	číslo stránky	zapíše vybranou stránku paměti flash status = 0: verifikace proběhla v pořádku status = 1: zápis se nepodařil (tzn. on-board verifikace)
		256 B	data stránky	1 B	status verifikace	
zápis flash a následné zpětné čtení	$0C_H$	2 B	číslo stránky	2 B	číslo stránky	zapíše vybranou stránku paměti flash a následně ji přečte (tzn. umožní verifikace v systému)
		256 B	data stránky	256 B	data stránky	

Doba trvání povelů **restart** a **rekonfigurace** nepřekračuje v případě úspěšného prvního pokusu o konfiguraci FPGA jednu sekundu. V případě neúspěšného prvního pokusu o konfiguraci jsou provedeny ještě druhý, případně i třetí pokus a až poté je provádění povelu ukončeno; doba trvání tří neúspěšných pokusů může dosáhnout až 20 sekund.

Doba trvání všech ostatních povelů nepřekračuje 200 milisekund a odpověď je vyslána po provedení povelu.

Upozornění: *Je potřeba vzít v úvahu, že odpověď povelů **restart** a **rekonfigurace** je vyslána ihned po přijetí dotazu a až následně je povel prováděn (tzn. mikropočítač nereguje na dotazy až do vykonání obsluhy povelu). Všechny ostatní povely jsou nejprve vykonány a až následně je vyslána odpověď.*

8.5 Povel identifikace

Povel identifikace je obecně použitelným povelem umožňujícím zjistit řadu informací o kartě bez ohledu na verzi firmware FPGA, resp. funkčnost FPGA.

struktura dotazu	struktura odpovědi
`{0000}`	`[00I…ISS]`
`{` = start_značka `00` = povel `00` = kontrolní číslo (uvedena přímo hodnota) `}` = stop_značka	`[` = start_značka `00` = povel `I…I` = 192 ASCII HEX znaků `SS` = kontrolní číslo `]` = stop_značka

Zatímco dotaz neobsahuje žádná data, v odpovědi je přenášeno 96 ASCII znaků obsahově strukturovaných do 6 řádek (nejsou odděleny CR-LF ani jinak).

Příklad výpisu identifikačních ASCII znaků:

`PCT-7424C`	typ karty
`174243001`	výrobní číslo karty (shodné se štítkem na kartě)
`PCIloader v.1.15`	identifikace firmware mikropočítače
`FPGA-18-1.4`	identifikace firmware FPGA (typ a verze, obě konstanty zobrazeny v HEX tvaru)
`FPGA Status: 1`	0 = chyba konfigurace FPGA, 1 = FPGA obsahuje základní firmware
`CardID: 3`	číslo 0÷3 nastavené DIP spínači
`1234567890123456`	(vyznačení pozice 16 znaků v předešlých řádkách)

Poznámka: *Podpora této funkce je implementovaná v administračním programu ovladače TEDIA_DAQ01.*

8.6 Povel restart mikropočítače

Povel slouží k restartování řídicího mikropočítače karty (jsou znovu inicializovány všechny vnitřní registry mikropočítače do výchozího stavu) následovaného konfigurací FPGA.

Povel má smysl výhradně pro servisní programy.

struktura dotazu	struktura odpovědi
{01FF}	[01FF]
{ = start_značka 01 = povel FF = kontrolní číslo (uvedena přímo hodnota) } = stop_značka	[= start_značka 01 = povel FF = kontrolní číslo (uvedena přímo hodnota)] = stop_značka

8.7 Povel rekonfigurace FPGA

Povel slouží k provedení nové konfigurace FPGA mikropočítačem karty (použito např. po update firmware FPGA).

Povel má smysl výhradně pro servisní programy.

struktura dotazu	struktura odpovědi
{02FE}	[02FE]
{ = start_značka 02 = povel FE = kontrolní číslo (uvedena přímo hodnota) } = stop_značka	[= start_značka 02 = povel FE = kontrolní číslo (uvedena přímo hodnota)] = stop_značka

8.8 Povel čtení flash paměti

Povel slouží k přečtení obsahu vybrané stránky paměti flash (použito např. pro update firmware FPGA).

Povel má smysl výhradně pro servisní programy.

struktura dotazu	struktura odpovědi
{0APPPPNN}	[0APPPPD…DSS]
{ = start_značka 0A = povel PPPP = 4 ASCII HEX znaky (16bitové číslo stránky) SS = kontrolní číslo } = stop_značka	[= start_značka 0A = povel PPPP = 4 ASCII HEX znaky (16bitové číslo stránky) D…D = 512 ASCII HEX znaků (256 B dat stránky) SS = kontrolní číslo] = stop_značka

8.9 Povel zápis flash paměti

Povel slouží k zápisu obsahu vybrané stránky paměti flash (použito např. pro update firmware FPGA).

Povel má smysl výhradně pro servisní programy.

struktura dotazu	struktura odpovědi
{0BPPPPD…DNN}	[0BPPPPVVSS]
{ = start_značka 0B = povel PPPP = 4 ASCII HEX znaky (16bitové číslo stránky) D…D = 512 ASCII HEX znaků (256 B dat stránky) SS = kontrolní číslo } = stop_značka	[= start_značka 0B = povel PPPP = 4 ASCII HEX znaky (16bitové číslo stránky) VV = výsledek verifikace zápisu (0=o.k., 1=chyba) SS = kontrolní číslo] = stop_značka

8.10 Povel zápis flash paměti se zpětným čtením

Povel slouží k zápisu a následně čtení obsahu vybrané stránky paměti flash (použito např. pro update firmware FPGA).

Povel má smysl výhradně pro servisní programy.

struktura dotazu	struktura odpovědi
{0CPPPPD…DNN}	[0CPPPPD…DSS]
{ = start_značka 0C = povel PPPP = 4 ASCII HEX znaky (16bitové číslo stránky) D…D = 512 ASCII HEX znaků (256 B dat stránky) SS = kontrolní číslo } = stop_značka	[= start_značka 0C = povel PPPP = 4 ASCII HEX znaky (16bitové číslo stránky) D…D = 512 ASCII HEX znaků (256 B dat stránky) SS = kontrolní číslo] = stop_značka

8.11 Rozdělení flash paměti

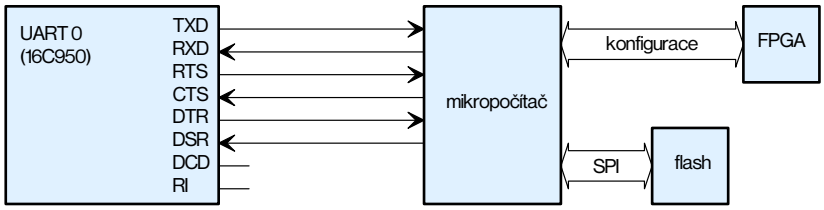
Flash má kapacitu 512 kB rozdělenou do 2048 stránek velikosti 256 B.

Prvních část paměti velikosti 64 kB (tzn. 256 stránek) je blokována proti zápisu (obsahuje tedy read–only konstanty), zbývající část je přístupná pro čtení i pro zápis.

stránka flash	význam
0	identifikační data karty
1÷255	read-only konstanty (například konfigurace karty z výroby)
256÷511	obecná data (například kalibrační konstanty)
512÷2047	konfigurační data FPGA (tzn. firmware FPGA)

8.12 Propojení UARTu s mikropočítačem

Na obrázku níže je zakresleno zjednodušené schéma propojení UARTu s mikropočítačem karty.



Význam signálů:

TXD	přenos dat z PC do mikropočítače (tzn. dotaz)
RXD	přenos dat z mikropočítače do PC (tzn. odpověď)
RTS	signalizuje mikropočítači, že FIFO ve směru do PC je plné a musí pozastavit vysílání odpovědi (UART musí být konfigurován do automatického RTS/CTS režimu)
CTS	nevyužito
DTR	rezerva
DSR	rezerva
DCD	nezapojeno (resp. signál nastaven do neaktivní úrovně)
RI	nezapojeno (resp. signál nastaven do neaktivní úrovně)

8.13 Popis programové obsluhy UARTu

Komunikace servisním rozhraním vyžaduje inicializaci UARTu podle následujícího postupu:

```
{ 16C950 registry }
WR_REG[ADDR+3] = 00H      { LCR: ujištění, že poslední zápis nebyl $BF }
WR_REG[ADDR+7] = 0CH      { SPR: index CSR registru }
WR_REG[ADDR+5] = 00H      { CSR: reset všech registrů s výjimkou CKS !!! }
WR_REG[ADDR+7] = 00H      { SPR: index ACR registru }
WR_REG[ADDR+5] = 20H      { ACR: ´950 trigger level enable }
WR_REG[ADDR+7] = 06H      { SPR: index FCL registru }
WR_REG[ADDR+5] = 60D      { FCL: autom. flow control lower trigger level }
WR_REG[ADDR+7] = 07H      { SPR: index FCH registru }
WR_REG[ADDR+5] = 120D     { FCH: autom. flow control higher trigger level }

{ 16C650 registry }
WR_REG[ADDR+3] = BFH      { LCR: přepnutí na rozšiřující ´650 registry }
WR_REG[ADDR+2] = 50H      { EFR: povolení RTS flow control + enhance mode}

{ standardní registry }
WR_REG[ADDR+3] = 80H      { LCR: nastavení DLAB=1, tzn. přístup k děličce }
WR_REG[ADDR+0] = 02H      { DLL: dolní byte děličky rychlosti :2 }
WR_REG[ADDR+1] = 00H      { DLM: horní byte děličky rychlosti :2 }
WR_REG[ADDR+3] = 03H      { LCR: 8 bitů, bez parity, 1 stopbit }
WR_REG[ADDR+1] = 00H      { IER: všechna přerušení zakázána }
WR_REG[ADDR+2] = 01H      { FCR: nastavení FIFO režimu }
WR_REG[ADDR+4] = 02H      { MCR: DTR=0, RTS=1 }
```

Každý řádek představuje zápis do jednoho registru, resp. zápis na adresu [ADDR+OFFSET], kde ADDR je bázová adresa UARTu 0 (tzn. adresa F0/BAR0, resp. F0/BAR5) a OFFSET je adresa registru v rámci osmi adres UARTu. Hodnota za rovnítkem představuje zapisovanou hodnotu.

Offsety adres v rozsahu 0÷7 jsou uvedeny pro I/O prostor mapovaný BAR0/F0, v případě použití MEM prostoru mapovaného BAR5/F0 je offset 4x větší, např. registr ADDR+1 je mapován současně na adresu BAR0/F0+1 BAR5/F0+4.

Po provedení inicializace lze vysílat zprávu zápisem do registru na adrese [ADDR+0] za podmínky volného místa v zásobníku UARTu (viz popis registru LSR na adrese [ADDR+5], bity D5 a D6).

Po vyslání zprávy lze přijmout odpověď čtením registru na adrese [ADDR+0] za podmínky přítomných dat v zásobníku UARTu (viz popis registru LSR na adrese [ADDR+5], bity D0÷D4, D7).

Popis registru LSR na adrese ADDR+5:

D7	D6	D5	D4	D3	D2	D1	D0
Data_Err	TX_empty	THR_empty	RX_break	Fram_Err	Par_Err	OverR_Err	RX_Rdy

RX_Rdy	úrovní 1 signalizuje přítomnost dat v přijímacím zásobníku (tzn. lze 1x číst znak z adresy ADDR+0)
OverR_Err	úrovní 1 signalizuje přetečení zásobníku přijímaných dat (fatální chyba přenosu)
Par_Err	úrovní 1 signalizuje chybu parity přijímaných dat (fatální chyba přenosu)
Fram_Err	úrovní 1 signalizuje chybu stopbitu přijímaných dat (fatální chyba přenosu)
RX_break	úrovní 1 signalizuje chybu "break" přijímaných dat (fatální chyba přenosu)
THR_empty	úrovní 1 signalizuje prázdný vysílací zásobník (tzn. lze 1x zapsat znak na adresu ADDR+0)
TX_empty	úrovní 1 signalizuje prázdný vysílací zásobník a navíc i vyslaný poslední znak
Data_Err	úrovní 1 signalizuje alespoň jednu z chyb parity, stopbitu a "break" přijímaných dat (viz předešlé bity)

Několik slov o TEDIA® spol. s r. o.

TEDIA® spol. s r. o. je ryze českým výrobcem měřicí a průmyslové elektroniky s vlastním vývojovým, výrobním a servisním zázemím s tradicí od roku 1994.

Společnost je od roku 2002 **řádným členem PCI Special Interest Group**, organizace odpovědné za standardizaci a vývoj sběrnice PCI a PCI Express.

Aktuální nabídka výrobků a služeb

Komponenty pro PC systémy

- multifunkční karty pro laboratorní a průmyslové aplikace
  (A/D převodníky, D/A převodníky, digitální porty, IRC a registrační čítače, ...)
- dceřiné desky pro úpravu signálů (izolační zesilovače, multiplexery, výkonové výstupy, ...)
- podpora sběrnic ISA, PC/104, PCI a PCI Express

Komponenty pro distribuované systémy - stavebnice modulů MicroUnit serie

- kompletní sortiment inteligentních I/O modulů (analogové vstupy a výstupy, digitální porty, čítače, ...)
- komunikace s nadřízeným systémem rozhraním RS-485, RS-422, RS-232 nebo LAN
- protokoly AIBus-2 (firemní komunikační protokol) a standardní Modbus RTU
- průmyslové provedení v pouzdru pro montáž na lištu DIN 35 mm
- speciální typy v zakázkovém provedení nebo LCD moduly pro montáž na čelní panel rozváděče

Komponenty pro komunikaci v průmyslovém prostředí

- komunikační karty pro sběrnice ISA, PC/104, PCI a PCI Express
- jedno, dvou a čtyřportové karty s řadiči UART s podporou rozhraní RS-232, RS-422 a RS-485
- konvertory a repeatery komunikačních rozhraní RS-232, RS-422 a RS-485 v pouzdru pro montáž na lištu DIN 35 mm
- miniaturní izolované USB konvertory pro rozhraní RS-232 a RS-485

Komponenty pro mobilní měřicí systémy

- multifunkční moduly pro nasazení v laboratořích a zkušebnách
- USB nebo LAN konektivita

Software

- vývoj univerzálních Windows ovladačů a ovladačů pro Control Web ovladačů pro naše výrobky
- jako oem partner dodáváme všechny komponenty systém Control Web
- dodáváme systém ScopeWin pro laboratorní nebo průmyslová měření a analýzu dat

Zakázkový vývoj a výroba elektroniky

- zkušenosti s vývojem více než stovky typů zásuvných PC karet a téměř 400 typy mikropočítačových desek
- speciální elektronika vybavená rozhraními a technologiemi, se kterými již máme zkušenosti
  - počítačové sběrnice PCI, PCI Express, ISA, USB, ethernet, ...
  - rutinní vyžívání hradlových polí FPGA a CPLD Altera
  - mikropočítače Cortex M3/M4, výkonné mikropočítače řady ´51
  - zpracování analogových signálů z technologických čidel (termočlánky, odporová čidla, tenzometry, inkrementální snímač, LVDT, ...)
  - aplikace A/D převodníků s vysokým rozlišením nad 20 bitů nebo vzorkovací frekvencí desítek MHz
  - zpracování a analýza signálů v reálném čase pomocí FPGA (FIR, DFT, ...)
- výroba zajištěna vlastním osazovacím automatem a pájecí pecí pracující na principu nasycených par
- zkušební laboratoř vybavená mj. klimatizační komorou pro zkoušky chladem, suchým teplem a vlhkým teplem podle norem ČSN EN 60068
- další informace viz http://www.tedia.cz/vyvoj

© 1994÷2015 TEDIA® spol. s r. o.

Vývoj, výroba, obchod, servis, technická podpora:

adresa:	TEDIA® spol. s r. o. Zábělská 12 31211 Plzeň Česká republika
internet:	http://www.tedia.cz http://www.pci.cz
telefon:	+420 373730421 (základní číslo) +420 373730426 (technická podpora)
fax:	+420 373730420
e-mail:	aktuální informace najdete na adresách http://www.tedia.cz/kontakty http://www.tedia.cz/podpora