

PCT-7303C/E

programátorská příručka

historie dokumentu:		
datum	verze	změny
24.1.2011	01.2012	výchozí verze
6.4.2012	04.2012	první finální verze
9.5.2012	05.2012	doplněna kapitola 11. Servisní rozhraní
25.9.2012	09.2012	oprava formálních chyb, drobná zpřesnění
6.12.2012	12.2012	oprava BAR1/F1 v odstavci 2.2 (změna I/O na MEM); zpřesněný popis FIFO_CLR v odstavci 7.3; rozšířený popis mapování registrů v odstavci 11.13
2.1.2013	01.2013	oprava několika překlepů, změna BoardID na CardID
19.3.2013	03.2013	oprava popisu kontrolního součtu v 11. kapitole, změna na kontrolní číslo
	09.2013	blokové schéma karty, verze firmware (reset), cardID
		XSTR reg přejmenovány na CNTXSTR, INIT DOUT
		FIFONoSmpIStbReg
30.9.2015	09.2015	drobná zpřesnění
10.12.2015	12.2015	oprava odstavce 6.4

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.

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 Rozdělení registrů
- 3.3 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 CNTCtrlReg (WR)
- 4.5 Registr CNTXSTREnReg (WR)
- 4.6 Registr CNTXSTRStatusReg (RD)
- 4.7 Registr CNTXSTRClrReg (WR)
- 4.8 Registr CNT0SetReg (WR)
- 4.9 Registr CNT0RngReg (WR)
- 4.10 Registr CNT0StrReg (RD)
- 4.11 Registr CNT0XStrReg (RD)
- 4.12 Registr CNT0CWReg (WR)
- 4.13 Registr CNT0StatReg (RD)
- 4.14 Schéma registrové struktury

5. Registry pro obsluhu komparátorů

- 5.1 Úvod
- 5.2 Funkce komparátorů
- 5.3 Registr CMPEnReg (WR)
- 5.4 Registr CMPStatReg (RD)
- 5.5 Registr CMPClrReg (WR)
- 5.6 Registr CNT0CMP1Reg (WR)
- 5.7 Registr CNT0CMP2Reg (WR)
- 5.8 Registr CNT1CMP1Reg (WR)
- 5.9 Registr CNT1CMP2Reg (WR)
- 5.10 Registr CNT2CMP1Reg (WR)
- 5.11 Registr CNT2CMP2Reg (WR)
- 5.12 Schéma registrové struktury

6. Registry pro detekci minima/maxima hodnoty čítačů

- 6.1 Úvod
- 6.2 Funkce detektorů minima/maxima
- 6.3 Registr CNTMinMaxEnReg (WR)
- 6.4 Registr CNTMinMaxCtrlReg (WR)
- 6.5 Registr CNT0MinReg (RD)
- 6.6 Registr CNT0MaxReg (RD)
- 6.7 Registr CNT1MinReg (RD)
- 6.8 Registr CNT1MaxReg (RD)
- 6.9 Registr CNT2MinReg (RD)
- 6.10 Registr CNT2MaxReg (RD)

7. Registry pro obsluhu FIFO paměti

- 7.1 Úvod
- 7.2 Funkce FIFO paměti
- 7.3 Registr ScanCtrlReg (WR)
- 7.4 FIFOStatusReg (RD)
- 7.5 ScanDINPH (WR)
- 7.6 ScanDINRE (WR)
- 7.7 ScanDINFE (WR)
- 7.8 ScanTimer (WR)
- 7.9 FIFONoSmpIStbReg (WR)
- 7.10 FIFONoSmpIReg (RD)
- 7.11 FIFOCNT0Reg (RD), FIFOCNT1Reg (RD), FIFOCNT2Reg (RD), FIFOTimeStamp (RD), FIFODINReg (RD)

8. Registry pro obsluhu přerušení

- 8.1 Úvod
- 8.2 Funkce obvodů pro obsluhu přerušení
- 8.3 Registr INTEnReg (WR)
- 8.4 Registr IRQCfgReg (WR)
- 8.5 Registr IRQStatusReg (RD)
- 8.6 Registr IRQClrReg (WR)
- 8.7 Registr TimerReg (WR, RD)
- 8.8 Registr CMPIRQReg (WR)
- 8.9 Registr FIFOIRQReg (WR)
- 8.10 Schéma registrové struktury

9. Registry digitálních vstupů/výstupů a RT digitálních výstupů

- 9.1 Úvod
- 9.2 Funkce digitálních portů
- 9.3 Registr DINReg (RD)
- 9.4 Registr DOUTReg (WR)
- 9.5 Registr RTDOUTReg (WR)
- 9.6 Registr RTDOUTCfgReg (WR)

10. Pomocné diagnostické registry

- 10.1 Úvod
- 10.2 Registr FreeRunCNTReg (RD)
- 10.3 Registr FreeRunCNTStbReg (WR)
- 10.4 Registr CardIDReg (RD)
- 10.5 Registr FPGATypeReg (RD)
- 10.6 Registr FPGAVerReg (RD)

11. Servisní rozhraní

- 11.1 Úvod
- 11.2 Zapojení UARTů
- 11.3 Struktura povelů
- 11.4 Přehled povelů
- 11.5 Povel identifikace
- 11.6 Povel restart mikropočítače
- 11.7 Povel rekonfigurace FPGA
- 11.8 Povel čtení flash paměti
- 11.9 Povel zápis flash paměti
- 11.10 Povel zápis flash paměti se zpětným čtením
- 11.11 Rozdělení flash paměti
- 11.12 Propojení UARTu s mikropočítačem
- 11.13 Popis programové obsluhy UARTu

1. Základní informace

1.1 Úvod

Tato programátorská příručka navazuje na uživatelskou příručku karet PCT-7303C/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_DAO01).

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 systém Linux).

1.2 Verze firmware

Aktuální verze firmware v době vydání příručky:

mikropočítač - firmware:	PCIloader v.1.15	
FPGA - typ firmware:	17	(reprezentováno hodnotou 17 _H)
FPGA - verze firmware:	0.7	(reprezentováno hodnotou 07 _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-7303C/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-7303C/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 lokální 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 ÷00 _H	Vendor ID	1760 _H (tzn. VID TEDIA)	1760 _H (tzn. VID TEDIA)
03 _H ÷02 _H	Device ID	0210 _H (PCT-7303C) 0212 _H (PCT-7303E)	0211 _H (PCT-7303C) 0213 _H (PCT-7303E)
0B _H ÷09 _H	Class Code	118000 _H (tzn. PCI třída "other data acquisition controller")	118000 _H (tzn. PCI třída "other data acquisition controller")
13 _H ÷10 _H	BAR0	UART0 (I/O, 8B, adresu přidělí BIOS)	lokální sběrnice (I/O, 256B, adresu přidělí BIOS)
17 _H ÷14 _H	BAR1	UART1 (I/O, 8B, adresu přidělí BIOS)	lokální sběrnice (MEM, 4kB, adresu přidělí BIOS)
1B _H ÷18 _H	BAR2	nevyužito	konfigurační registry řadiče (I/O, 32B, adresu přidělí BIOS)
1F _H ÷1C _H	BAR3	nevyužito	konfigurační registry řadiče (MEM, 4kB, adresu přidělí BIOS)
23 _H ÷20 _H	BAR4	konfigurační registry řadiče (I/O, 32B, adresu přidělí BIOS)	nevyužito
27 _H ÷24 _H	BAR5	UART0 + UART1 + konfig. registry (MEM, 4kB, adresu přidělí BIOS)	nevyužito
2D _H ÷2C _H	Subsystem Vendor ID	1760 _H (tzn. VID TEDIA)	1760 _H (tzn. VID TEDIA)
2F _H ÷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 přerušeni

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 nízkourovňový 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 mikro počítačem karty (mikro počítač 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 funkčním registrům karty ...

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

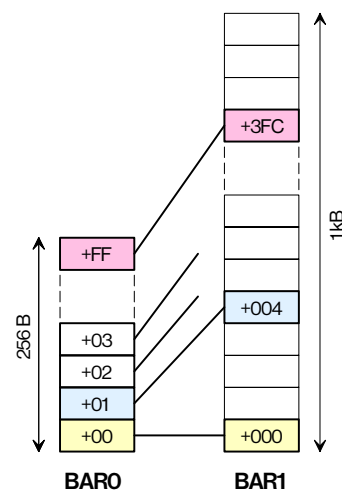
Odlišuje se nějak mapování funkčních registrů v BAR0 a BAR1 PCI funkce F1 karty ...

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	DOUReg	(zpětné čtení)
+108/104/100		FIFOCNT0Reg
+118/114/110		FIFOCNT1Reg
+128/124/120		FIFOCNT2Reg
+138/134/130		FIFOTimeStamp (24 bitů @ 1 MHz)
+140		FIFODINReg
+150	FIFONoSmpIStrbReg	FIFONoSmpIReg (0..1024)
+154	FIFOIRQReg	
+160	ScanCtrlReg	FIFOStatusReg
+164	ScanDINPH	(zpětné čtení)
+168	ScanDINRE	(zpětné čtení)
+16C	ScanDINFE	(zpětné čtení)
+178/174/170	ScanTimer (24 bitů @ 25 MHz pro FIFO)	(zpětné čtení)
+180	IRQCfgReg	IRQStatusReg
+184	IRQClrReg	
+18C	INTEnReg	(zpětné čtení)
+190	CNTXSTREnReg	CNTXSTRStatusReg
+194	CNTXSTRClrReg	
+208/204/200	CNT0SetReg	CNT0StrReg
+218/214/210	CNT0RngReg	CNT0XStrReg
+228/224/220	CNT0CMP1Reg	CNT0MinReg
+238/234/230	CNT0CMP2Reg	CNT0MaxReg
+270	CNT0CWReg	CNT0StatReg
+288/284/280	CNT1SetReg	CNT1StrReg
+298/294/290	CNT1RngReg	CNT1XStrReg
+2A8/2A4/2A0	CNT1CMP1Reg	CNT1MinReg
+2B8/2B4/2B0	CNT1CMP2Reg	CNT1MaxReg
+2F0	CNT1CWReg	CNT1StatReg
+308/304/300	CNT2SetReg	CNT2StrReg
+318/314/310	CNT2RngReg	CNT2XStrReg
+328/324/320	CNT2CMP1Reg	CNT2MinReg
+338/334/330	CNT2CMP2Reg	CNT2MaxReg
+370	CNT2CWReg	CNT2StatReg
+380	CNTEnReg	
+384	CNTCtrlReg	
+388	CNTMinMaxEnReg	(zpětné čtení)
+38C	CNTMinMaxCtrlReg	
+390	CMPEnReg	CMPStatReg
+394	CMPClrReg	
+398	CMPIRQReg	
+3A0	RTDOUReg	
+3A4	RTDOUTCfgReg	
+3EC/3E8/3E4/3E0	FreeRunCNTStrbReg	FreeRunCNTReg (32 bitů @ 100 kHz)
+3F0	TimerReg	TimerReg
+3F4		CardIDReg
+3F8		FPGATypeReg
+3FC		FPGAVerReg

Upozornění: Registry *CNT0RngReg*, *CNT1RngReg* a *CNT2RngReg* jsou po startu karty (resp. konfiguraci FPGA) nastaveny na hodnotu $FFFFFF_H$, všechny ostatní registry jsou po startu karty vynulovány. Při spuštění programu však nelze na tento stav spoléhat, jelikož registry mohou být nastaveny na odlišné hodnoty předchozím programem.

3.2 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)
modrá	registry související s komparátory hodnoty čítačů
oranžová	registry související s detekcí minima/maxima hodnoty čítače
červená	registry související s FIFO
žlutá	registry související s přerušením systému
šedá	registry digitálních vstupů/výstupů 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).

3.3 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 ...

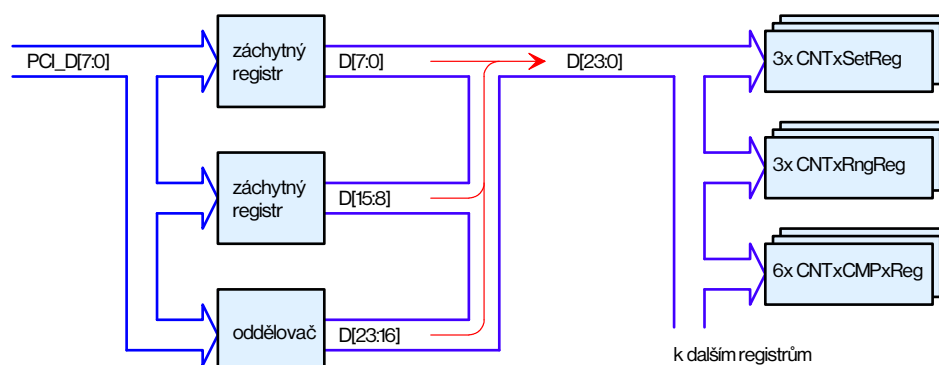
- osmibitový registr s nejnižší adresou obsahuje nejnižší bity vícebytového registru a osmibitový registr s nejvyšší adresou obsahuje nejvyšší bity vícebytového registru
- čtení i zápis osmibitových registrů musí probíhat v pořadí od registru s nejnižší adresou po registr s nejvyšší adresou
- musí být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

Pro příklad postup čtení a výpočtu 24bitové hodnoty registru CNT0StrReg:

- čtení B_0 = obsah registru na adrese $+200_H$
 - čtení B_1 = obsah registru na adrese $+204_H$
 - čtení B_2 = obsah registru na adrese $+208_H$
- výpočet $\text{výsledná hodnota} = 65536 * B_2 + 256 * B_1 + B_0$

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 CNT0SetReg jde o adresu $+200_H$), v druhém cyklu se zapisuje do záchytného registru s významem D[15:8] (pro CNT0SetReg jde o adresu $+204_H$) a v třetím cyklu při zápisu dat s významem D[23:16] (pro CNT0SetReg jde o adresu $+208_H$) se do odpovídajícího funkčního registru přenáší všech 24 bitů dat.



Poznámka: Obrázek zachycuje stav aktuální verze firmwaru. V budoucích verzích firmwaru mohou být funkční registry vybaveny vlastními záchytnými registry, avšak nelze se spoléhat na uchování obsahu záchytných registrů při postupném zápisu do různých funkčních registrů a vždy zapisovat/číst data celé bitové šířky.

4. Registry pro obsluhu čítačů

4.1 Úvod

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

Registry lze rozdělit na skupinu společnou všem čítačům

CNTEnReg	povoluje čítání, resp. nulování externími signály
CNTCtrlReg	umožňuje programovat hodnotu čítačů, resp. zachytit aktuální hodnotu čítačů do registrů
CNTXSTREnReg	aktivuje obvody pro zachycení hodnoty čítačů do vyrovnávacích registrů signálem EXT-IN
CNTXSTRStatusReg	umožňuje zjistit stav příznaků zachycení hodnoty čítačů externím signálem
CNTXSTRClrReg	umožňuje nulovat registr CNTXSTRStatusReg

a skupinu registrů implementovaných pro každý čítač samostatně (registry mají názvy CNT0..., CNT1... nebo CNT2...)

CNT0SetReg	obsahuje hodnotu programovanou do čítače CNT0
CNT0RngReg	definuje rozsah čítání čítače CNT0 (0÷1 až 0÷16.777.215)
CNT0StrReg	zpřístupňuje hodnotu čítače CNT0 zachycenou softwarovým povelům
CNT0XStrReg	zpřístupňuje hodnotu čítače CNT0 zachycenou externím signálem
CNT0CWReg	konfiguruje pracovní režim čítače CNT0
CNT0StatReg	zpřístupňuje stavový registr čítače CNT0 (stav řídicího signálů, příznak chyby vstupních signálů)
CNT1...	analogická sada šesti registrů pro čítač CNT1
CNT2...	analogická sada šesti registrů pro čítač CNT2

4.2 Funkce čítačů

V dále uvedených odstavcích budou popsány postupy typických programových obsluh čítačů.

Konfigurace čítačů (popsáno pro čítač CNT0)

Ke konfiguraci jsou určeny registry CNT0CWReg a CNT0RngReg umožňující nastavit režim čítače a rozsah čítání. Následně je možné nastavit hodnotu čítače pomocí CNT0SetReg (resp. také CNTCtrlReg).

Vhodné je ověřit stav registru CNT0StatReg (příznak chyby) a případně jej vynulovat pomocí registru CNT0CWReg.

Spuštění čítačů (popsáno pro čítač CNT0)

Ke spuštění a zastavení čítačů (tzn. reakce na vstupní signály) je určen registr CNTEnReg.

Softwarové čtení čítačů (popsáno pro čítač CNT0)

Čtení čítačů probíhá ve dvou fázích; v první je hodnota čítače (resp. vybraných čítačů) zachycena do vyrovnávacích registrů pomocí CNTCtrlReg a v druhé fázi je zachycená hodnota čtena pomocí CNT0StrReg.

Čtení hodnot zachycených externím signálem (popsáno pro čítač CNT0)

Tento režim je určen pro aplikace, ve kterých je potřeba synchronizovat čtení čítače (resp. zachycení aktuální hodnoty) vnější událostí, v případě PCT-7303C/E sestupnou hranou signálu EXT-IN, a reagovat na událost v reálném čase.

Nejprve je potřeba povolit detekci události pomocí registru CNTXSTREnReg, následně softwarovým pollingem (registr CNTXSTRStatusReg) nebo s podporou přerušení vyčkat na příchod události a nakonec přečíst hodnotu čítače (resp. hodnoty čítačů, v tomto režimu jsou zachyceny vždy hodnoty všech tří čítačů) pomocí CNT0XStrReg. Po přečtení je nutné povolit další detekci události nulováním příznaku v registru CNTXSTRStatusReg pomocí CNTXSTRClrReg.

Čtení hodnot záznamem do paměti FIFO

Tento režim je určen pro aplikace, ve kterých je potřeba zaznamenávat polohu v nastaveném časovém intervalu (frekvence zápisu do FIFO je pak řízena časovačem generujícím signál s frekvencí od 1,5 Hz do desítek kHz) nebo změnou stavu digitálního vstupního portu. Reakce v reálném čase je možná při detekci zaplnění FIFO jedním vzorkem.

Záznam do FIFO je společný pro všechny čítače a kromě jejich hodnoty je zaznamenán i stav vstupního digitálního portu a časová značka.

Poznámka: Všechny tři způsoby čtení jsou nezávislé a mohou pracovat současně. Samozřejmě však sdílejí společnou konfiguraci čítačů a jejich spuštění/zastavování. Programová obsluha paměti FIFO je popsána v samostatné kapitole.

4.3 Registr CNTEnReg (WR)

Tento registr slouží ke spuštění a zastavení čítačů, resp. k povolení nulování externím signálem.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	EN_R2	EN_R1	EN_R0	RSRV	EN_AB2	EN_AB1	EN_AB0

EN_AB0	povolení čítání CNT0 0 čítač je zastaven 1 čítač zpracovává signály CNT0_A a CNT0_B
EN_AB1	povolení čítání CNT1 0 čítač je zastaven 1 čítač zpracovává signály CNT1_A a CNT1_B
EN_AB2	povolení čítání CNT2 0 čítač je zastaven 1 čítač zpracovává signály CNT2_A a CNT2_B
EN_R0	povolení nulování CNT0 0 čítač ignoruje signál CNT0_R 1 čítač zpracovává signál CNT0_R (aktivní úroveň signálu je konfigurovatelná CNT0CWReg)
EN_R1	povolení nulování CNT1 0 čítač ignoruje signál CNT1_R 1 čítač zpracovává signál CNT1_R (aktivní úroveň signálu je konfigurovatelná CNT1CWReg)
EN_R2	povolení nulování CNT2 0 čítač ignoruje signál CNT2_R 1 čítač zpracovává signál CNT2_R (aktivní úroveň signálu je konfigurovatelná CNT2CWReg)
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

4.4 Registr CNTCtrlReg (WR)

Tento registr slouží k softwarovému zachycení aktuální hodnoty čítačů do záchytných registrů a rovněž nastavení hodnoty čítačů podle předvolených hodnot. Všechny bity lze používat současně, čítače umožňují současné čtení i nastavení.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	SET_2	SET_1	SET_0	RSRV	STR_2	STR_1	STR_0

STR_0	přenos aktuální hodnoty čítače CNT0 do registru CNT0StrReg 0 bez významu, stav CNT0StrReg není modifikován 1 data přenesena do registru (generuje krátký puls, následný zápis 0 není vyžadován)
STR_1	přenos aktuální hodnoty čítače CNT1 do registru CNT1StrReg 0 bez významu, stav CNT1StrReg není modifikován 1 data přenesena do registru (generuje krátký puls, následný zápis 0 není vyžadován)
STR_2	přenos aktuální hodnoty čítače CNT2 do registru CNT2StrReg 0 bez významu, stav CNT2StrReg není modifikován 1 data přenesena do registru (generuje krátký puls, následný zápis 0 není vyžadován)
SET_0	přenos aktuální hodnoty z registru CNT0SetReg do čítače CNT0 0 bez významu, stav čítače není modifikován 1 data přenesena do čítače (generuje krátký puls, následný zápis 0 není vyžadován)
SET_1	přenos aktuální hodnoty z registru CNT1SetReg do čítače CNT1 0 bez významu, stav čítače není modifikován 1 data přenesena do čítače (generuje krátký puls, následný zápis 0 není vyžadován)
SET_2	přenos aktuální hodnoty z registru CNT2SetReg do čítače CNT2 0 bez významu, stav čítače není modifikován 1 data přenesena do čítače (generuje krátký puls, následný zápis 0 není vyžadován)
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

4.5 Registr CNTXSTREnReg (WR)

Tento registr slouží k aktivaci obvodů pro zachycení hodnoty čítačů do vyrovnávacích registrů externím signálem EXT-IN (detekovanou událostí je sestupná hrana signálu, tzn. přechod signálu z úrovně H do L).

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	XSTR	RSRV					

XSTR aktivace zachycení hodnot čítačů do registrů CNT0XStrReg, CNT1XStrReg a CNT2XStrReg
 0 obvody pro zachycení stavu čítačů neaktivní
 1 obvody pro zachycení stavu čítačů aktivovány

RSRV rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

4.6 Registr CNTXSTRStatusReg (RD)

Tento registr slouží k detekci zachycení stavu čítačů do vyrovnávacích registrů (viz popis CNTXSTREnReg).

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	XSTR	RSRV					

XSTR signalizace zachycení hodnot čítačů do registrů CNT0XStrReg, CNT1XStrReg a CNT2XStrReg
 0 nové hodnoty nejsou zachyceny, čeká se na příchod události
 1 nové hodnoty čítačů byly zachyceny

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

4.7 Registr CNTXSTRClrReg (WR)

Tento registr slouží k nulování příznaku v CNTXSTRStatusReg, tzn. k novému povolení detekce události na signálu EXT-IN.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	XSTR	RSRV					

XSTR aktivace zachycení hodnoty čítačů externím signálem do CNT0XStrReg, CNT1XStrReg a CNT2XStrReg
 0 bez významu, stav příznaku zachován
 1 příznak vynulován (generuje krátký puls, následný zápis 0 není vyžadován)

RSRV rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

4.8 Registr CNT0SetReg (WR)

Tento registr plní funkci 24bitového vyrovnávacího datového registru pro zápis dat do čítače CNT0, data jsou do čítače CNT0 přenesena pomocí registru CNTCtrlReg. Pro čítače CNT1 a CNT2 jsou implementovány analogické registry.

Je-li do čítače CNT0 zapsána hodnota mimo rozsah 0÷CNT0RngReg, pracuje čítač v plném 24bitovém rozsahu až do okamžiku, kdy hodnota čítače vstoupí do zvoleného rozsahu 0÷CNT0RngReg.

4.9 Registr CNT0RngReg (WR)

Tento 24bitový registr je určen pro nastavení rozsahu čítání čítače CNT0, platné hodnoty jsou 1 až 16.777.215 (čítač CNT0 pak pracuje v rozsahu 0÷CNT0RngReg). Pro čítače CNT1 a CNT2 jsou implementovány analogické registry.

Je-li do CNT0RngReg zapsána taková hodnota, že aktuální stav čítače CNT0 je mimo rozsah 0÷CNT0RngReg, pracuje čítač v plném 24bitovém rozsahu až do okamžiku, kdy hodnota čítače vstoupí do zvoleného rozsahu 0÷CNT0RngReg (zpracováním vstupních signálů nebo naprogramováním hodnoty).

4.10 Registr CNT0StrReg (RD)

Tento registr plní funkci 24bitového vyrovnávacího datového registru pro softwarové čtení dat z čítače CNT0, data jsou z čítače CNT0 přenesena pomocí registru CNTCtrlReg. Pro čítače CNT1 a CNT2 jsou implementovány analogické registry.

4.11 Registr CNT0XStrReg (RD)

Tento registr plní funkci 24bitového vyrovnávacího datového registru pro zachycení aktuální hodnoty čítače CNT0 externím signálem (viz popis registrů CNTXSTREnReg, CNTXSTRStatusReg a CNTXSTRClrReg). Pro čítače CNT1 a CNT2 jsou implementovány analogické registry.

4.12 Registr CNT0CWReg (WR)

Tento registr slouží ke konfiguraci čítače CNT0, pro čítače CNT1 a CNT2 jsou implementovány analogické registry.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	CNT_MODE			ERR	RSRV	LPF	R_CFG

R_CFG	volba polarity nulovacího pulsu (signál CNT0_R) 0 čítač nulován úrovní L vstupního signálu 1 čítač nulován úrovní H vstupního signálu
LPF	aktivace dolnopropustního filtru enkodéru vstupních signálů 0 filtr vypnut 1 filtr aktivován
ERR	nulování příznaku ERR v CNT0StatReg 0 bez významu, stav příznaku zachován 1 příznak vynulován (generuje krátký puls, následný zápis 0 není vyžadován)
CNT_MODE	volba pracovního režimu čítače 000 kvadrurní enkodér, režim X1 001 kvadrurní enkodér, režim X2 010 kvadrurní enkodér, režim X4 011 rezerva 100 režim "up/down" 101 režim "count/dir" 110 režim "count/gate" 111 rezerva
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

Poznámka: Pracovní režimy jsou podrobně popsány v uživatelské příručce.

4.13 Registr CNT0StatReg (RD)

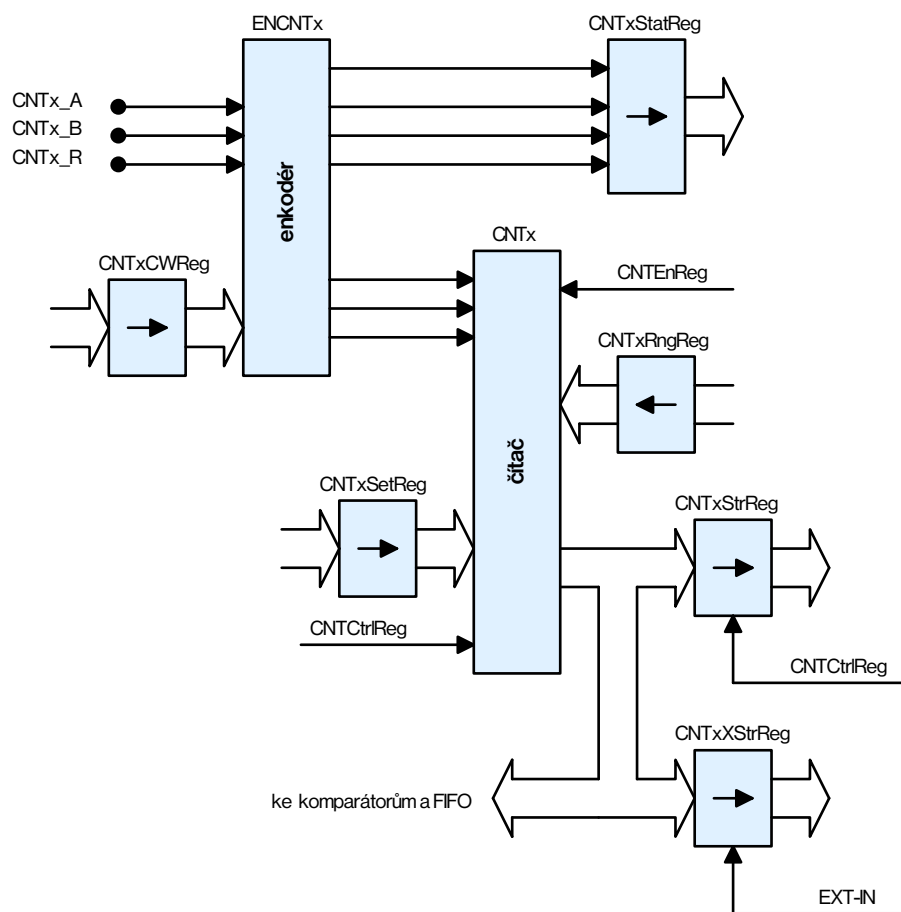
Tento registr slouží ke čtení stavových příznaků čítače CNT0, pro čítače CNT1 a CNT2 jsou implementovány analogické registry.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV				ERR	CNT0_R	CNT0_B	CNT0_A

CNT0_A	aktuální stav signálu CNT0_A
CNT0_B	aktuální stav signálu CNT0_B
CNT0_R	aktuální stav signálu CNT0_R
ERR	chybový příznak signalizující "přeskočení" fáze kvadrurního signálu v režimech X1, X2 a X4 nebo detekovaný současný stav signálů CNT0_A=L a CNT0_B=L v režimu "up/down" 0 od posledního nulování příznaku nebyla detekována chyba 1 od posledního nulování příznaku byla detekována chyba
RSRV	rezerva (z důvodu dopředné kompatibility je doporučeno hodnoty ignorovat)

4.14 Schéma registrové struktury

Na níže uvedeném obrázku jsou pro lepší porozumění zakresleny registry související s čítači (zobrazuje jeden čítač; registry CNTx jsou implementovány pro každý čítač samostatně, registry CNT jsou společné všem třem čítačů a umožňují jejich synchronní řízení).



5. Registry pro obsluhu komparátorů

5.1 Úvod

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

Registry lze rozdělit na skupinu společnou všem komparátorům (resp. čítačům)

CMPEnReg	aktivuje/deaktivuje záchytné registry komparátorů
CMPSatReg	příznaky záchytných registrů komparátorů
CMPClrReg	nuluje záchytné registry komparátorů

a skupinu registrů implementovaných pro každý komparátor (resp. čítač) samostatně (registry mají názvy CNT0..., CNT1... nebo CNT2...)

CNT0CMP1Reg	24bitová hodnota 1. komparátoru čítače CNT0
CNT0CMP2Reg	24bitová hodnota 2. komparátoru čítače CNT0
CNT1...	analogická sada dvou registrů pro čítač CNT1
CNT2...	analogická sada dvou registrů pro čítač CNT2

5.2 Funkce komparátorů

Každý čítač má k dispozici dva nezávislé komparátory trvale porovnávající aktuální stav čítače s hodnotou přednastavenou v registru CNTxCMPxReg. V případě dosažení shody (a současně aktivace pomocí CMPEnReg) je záchytný registr nastaven na hodnotu 1 a na této hodnotě setrvává až do vynulování registrem CMPClrReg. Stav všech šest záchytných registrů lze zjistit pomocí registru CMPSatReg.

Všechny tři registry (CMPEnReg, CMPSatReg a CMPClrReg) jsou bitově orientovány a umožňují tedy nezávisle konfigurovat každý komparátor.

Nastavení záchytných registrů může být detekováno jednak softwarově, využito pro vyvolání přerušení systému nebo obsah záchytných registrů směřován na RT digitální výstupy (viz popis v samostatných kapitolách).

5.3 Registr CMPEnReg (WR)

Tento registr slouží k aktivaci/deaktivaci záchytných registrů všech šesti komparátorů.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	EN_CMP22	EN_CMP12	EN_CMP02	RSRV	EN_CMP21	EN_CMP11	EN_CMP01

EN_CMP01	aktivuje/deaktivuje záchytný registr 1. komparátoru čítače CNT0 0 záchytný registr je blokován 1 záchytný registr je aktivován
EN_CMP11	aktivuje/deaktivuje záchytný registr 1. komparátoru čítače CNT1 0 záchytný registr je blokován 1 záchytný registr je aktivován
EN_CMP21	aktivuje/deaktivuje záchytný registr 1. komparátoru čítače CNT2 0 záchytný registr je blokován 1 záchytný registr je aktivován
EN_CMP02	aktivuje/deaktivuje záchytný registr 2. komparátoru čítače CNT0 0 záchytný registr je blokován 1 záchytný registr je aktivován
EN_CMP12	aktivuje/deaktivuje záchytný registr 2. komparátoru čítače CNT1 0 záchytný registr je blokován 1 záchytný registr je aktivován
EN_CMP22	aktivuje/deaktivuje záchytný registr 2. komparátoru čítače CNT2 0 záchytný registr je blokován 1 záchytný registr je aktivován
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

5.4 Registr CMPStatReg (RD)

Tento registr slouží k zjištění stavu záchytných registrů všech šesti komparátorů povolovaných registrem CMPEnReg.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	CMP_22	CMP_12	CMP_02	RSRV	CMP_21	CMP_11	CMP_01

CMP01	stav záchytného registru 1. komparátoru čítače CNT0
0	registr není nastaven, tzn. od posledního nulování nenastala shoda CNT0 s CNT0CMP1Reg
1	registr je nastaven, tzn. od posledního nulování byla detekována shoda CNT0 s CNT0CMP1Reg
CMP11	stav záchytného registru 1. komparátoru čítače CNT1
0	registr není nastaven, tzn. od posledního nulování nenastala shoda CNT1 s CNT1CMP1Reg
1	registr je nastaven, tzn. od posledního nulování byla detekována shoda CNT1 s CNT1CMP1Reg
CMP21	stav záchytného registru 1. komparátoru čítače CNT2
0	registr není nastaven, tzn. od posledního nulování nenastala shoda CNT2 s CNT2CMP1Reg
1	registr je nastaven, tzn. od posledního nulování byla detekována shoda CNT2 s CNT2CMP1Reg
CMP02	stav záchytného registru 2. komparátoru čítače CNT0
0	registr není nastaven, tzn. od posledního nulování nenastala shoda CNT0 s CNT0CMP2Reg
1	registr je nastaven, tzn. od posledního nulování byla detekována shoda CNT0 s CNT0CMP2Reg
CMP12	stav záchytného registru 2. komparátoru čítače CNT1
0	registr není nastaven, tzn. od posledního nulování nenastala shoda CNT1 s CNT1CMP2Reg
1	registr je nastaven, tzn. od posledního nulování byla detekována shoda CNT1 s CNT1CMP2Reg
CMP22	stav záchytného registru 2. komparátoru čítače CNT2
0	registr není nastaven, tzn. od posledního nulování nenastala shoda CNT2 s CNT2CMP2Reg
1	registr je nastaven, tzn. od posledního nulování byla detekována shoda CNT2 s CNT2CMP2Reg
RSRV	rezerva (z důvodu dopředné kompatibility je doporučeno hodnoty ignorovat)

5.5 Registr CMPClrReg (WR)

Tento registr slouží k nulování záchytných registrů všech šesti komparátorů povolovaných registrem CMPEnReg.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	CL_CMP22	CL_CMP12	CL_CMP02	RSRV	CL_CMP21	CL_CMP11	CL_CMP01

CL_CMP01	nuluje záchytný registr 1. komparátoru čítače CNT0
0	bez významu, stav záchytného registru není modifikován
1	záchytný registr je vynulován (generuje krátký puls, následný zápis 0 není vyžadován)
CL_CMP11	nuluje záchytný registr 1. komparátoru čítače CNT1
0	bez významu, stav záchytného registru není modifikován
1	záchytný registr je vynulován (generuje krátký puls, následný zápis 0 není vyžadován)
CL_CMP21	nuluje záchytný registr 1. komparátoru čítače CNT2
0	bez významu, stav záchytného registru není modifikován
1	záchytný registr je vynulován (generuje krátký puls, následný zápis 0 není vyžadován)
CL_CMP02	nuluje záchytný registr 2. komparátoru čítače CNT0
0	bez významu, stav záchytného registru není modifikován
1	záchytný registr je vynulován (generuje krátký puls, následný zápis 0 není vyžadován)
CL_CMP12	nuluje záchytný registr 2. komparátoru čítače CNT1
0	bez významu, stav záchytného registru není modifikován
1	záchytný registr je vynulován (generuje krátký puls, následný zápis 0 není vyžadován)
CL_CMP22	nuluje záchytný registr 2. komparátoru čítače CNT2
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)

5.6 Registr CNT0CMP1Reg (WR)

Tento 24bitový registr definuje práh prvního komparátoru čítače CNT0. Při dosažení ekvivalence obsahu čítače s tímto registrem může být nastaven odpovídající příznak v registru CMPStatReg (viz CMPEnReg) a případně i vyvoláno přerušení (viz CMPIRQReg) nebo nastaven "real-time" digitální výstup (viz RTDOCfgReg).

5.7 Registr CNT0CMP2Reg (WR)

Tento 24bitový registr definuje práh druhého komparátoru čítače CNT0. Význam je analogický registru CNT0CMP1Reg.

5.8 Registr CNT1CMP1Reg (WR)

Tento 24bitový registr definuje práh prvního komparátoru čítače CNT1. Význam je analogický registru CNT0CMP1Reg.

5.9 Registr CNT1CMP2Reg (WR)

Tento 24bitový registr definuje práh druhého komparátoru čítače CNT1. Význam je analogický registru CNT0CMP1Reg.

5.10 Registr CNT2CMP1Reg (WR)

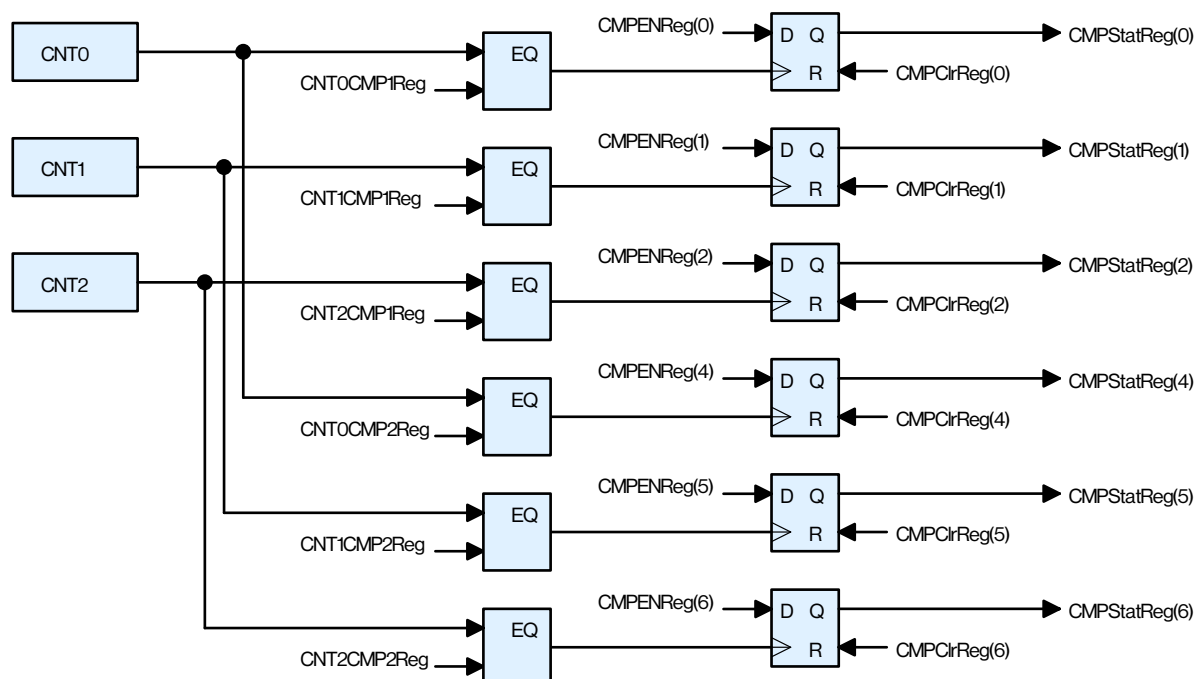
Tento 24bitový registr definuje práh prvního komparátoru čítače CNT2. Význam je analogický registru CNT0CMP1Reg.

5.11 Registr CNT2CMP2Reg (WR)

Tento 24bitový registr definuje práh druhého komparátoru čítače CNT2. Význam je analogický registru CNT0CMP1Reg.

5.12 Schéma registrové struktury

Na níže uvedeném obrázku jsou pro lepší porozumění souvislostí zakresleny registry související s komparátory.



6. Registry pro detekci minima/maxima hodnoty čítačů

6.1 Úvod

V následujících odstavcích budou popsány registry související s obvodem pro detekci minima/maxima čítačů (dále také detektor minima/maxima), viz přehled v 3. kapitole.

Registry lze rozdělit na skupinu společnou všem detektorům

- CNTMinMaxEnReg aktivuje/deaktivuje detektory minima/maxima
- CNTMinMaxCtrlReg umožňuje zachytit aktuální hodnotu detektorů do registrů

a skupinu registrů implementovaných pro každý detektor (resp. čítač) samostatně (registry mají názvy CNT0..., CNT1... nebo CNT2...)

- CNT0MinReg 24bitová hodnota záchytného registru detektoru minima čítače CNT0
- CNT0MaxReg 24bitová hodnota záchytného registru detektoru maxima čítače CNT0
- CNT1... analogická sada dvou registrů pro čítač CNT1
- CNT2... analogická sada dvou registrů pro čítač CNT2

6.2 Funkce detektorů minima/maxima

Každý čítač má k dispozici nezávisle pracující detektor minimální hodnoty a detektor maximální hodnoty.

Všech šest detektorů je ovládáno dvojicí registrů CNTMinMaxEnReg a CNTMinMaxCtrlReg.

Registr CNTMinMaxEnReg slouží k aktivaci, resp. k restartu detektorů. Po dobu, kdy je odpovídající bit tohoto registru nastaven na hodnotu 0, detektor kopíruje aktuální hodnotu čítače do interního pracovního registru detektoru. Po nastavení odpovídajícího bitu registru CNTMinMaxEnReg na hodnotu 1 detektor začíná vyhodnocovat aktuální hodnotu čítače a průběžně aktualizuje hodnotu interního pracovního registru podle změn čítače (detektor maxima aktualizuje svůj vnitřní registr každou vyšší hodnotou čítače, detektor minima naopak aktualizuje svůj vnitřní registr každou nižší hodnotou čítače).

Pro čtení interních pracovních registrů jsou určeny záchytné registry společně řízené registrem CNTMinMaxCtrlReg.

Detektory minima/maxima neumožňují vyvolat přerušení a nejsou ani jiným způsobem propojeny s dalšími obvody karty.

6.3 Registr CNTMinMaxEnReg (WR)

Tento registr slouží k aktivaci/deaktivaci funkce všech šesti detektorů minima/maxima.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	EN_MAX2	EN_MAX1	EN_MAX0	RSRV	EN_MIN2	EN_MIN1	EN_MIN0

EN_MIN0	aktivuje/deaktivuje detektor minima čítače CNT0 0 funkce detektoru je blokována 1 detektor je aktivován, přechodem 0=>1 je detektor restartován
EN_MIN1	aktivuje/deaktivuje detektor minima čítače CNT1 0 funkce detektoru je blokována 1 detektor je aktivován, přechodem 0=>1 je detektor restartován
EN_MIN2	aktivuje/deaktivuje detektor minima čítače CNT2 0 funkce detektoru je blokována 1 detektor je aktivován, přechodem 0=>1 je detektor restartován
EN_MAX0	aktivuje/deaktivuje detektor maxima čítače CNT0 0 funkce detektoru je blokována 1 detektor je aktivován, přechodem 0=>1 je detektor restartován
EN_MAX1	aktivuje/deaktivuje detektor maxima čítače CNT1 0 funkce detektoru je blokována 1 detektor je aktivován, přechodem 0=>1 je detektor restartován
EN_MAX2	aktivuje/deaktivuje detektor maxima čítače CNT2 0 funkce detektoru je blokována 1 detektor je aktivován, přechodem 0=>1 je detektor restartován
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

6.4 Registr CNTMinMaxCtrlReg (WR)

Tento registr slouží k softwarovému zachycení aktuální hodnoty interních pracovních registrů detektorů minima nebo maxima do záchytných registrů.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	STR_MAX2	STR_MAX1	STR_MAX0	RSRV	STR_MIN2	STR_MIN1	STR_MIN0

STR_MIN0	přenos aktuální hodnoty interního registru detektoru minima CNT0 do registru CNT0MinReg 0 bez významu, stav CNT0MinReg není modifikován 1 data přenesena do registru (generuje krátký puls, následný zápis 0 není vyžadován)
STR_MIN1	přenos aktuální hodnoty interního registru detektoru minima CNT1 do registru CNT1MinReg 0 bez významu, stav CNT1MinReg není modifikován 1 data přenesena do registru (generuje krátký puls, následný zápis 0 není vyžadován)
STR_MIN2	přenos aktuální hodnoty interního registru detektoru minima CNT2 do registru CNT2MinReg 0 bez významu, stav CNT2MinReg není modifikován 1 data přenesena do registru (generuje krátký puls, následný zápis 0 není vyžadován)
STR_MAX0	přenos aktuální hodnoty interního registru detektoru maxima CNT0 do registru CNT0MaxReg 0 bez významu, stav CNT0MaxReg není modifikován 1 data přenesena do registru (generuje krátký puls, následný zápis 0 není vyžadován)
STR_MAX1	přenos aktuální hodnoty interního registru detektoru maxima CNT1 do registru CNT1MaxReg 0 bez významu, stav CNT1MaxReg není modifikován 1 data přenesena do registru (generuje krátký puls, následný zápis 0 není vyžadován)
STR_MAX2	přenos aktuální hodnoty interního registru detektoru maxima CNT2 do registru CNT2MaxReg 0 bez významu, stav CNT2MaxReg není modifikován 1 data přenesena do registru (generuje krátký puls, následný zápis 0 není vyžadován)
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

6.5 Registr CNT0MinReg (RD)

Tento 24bitový registr obsahuje hodnotu detektoru minima čítače CNT0 zachycenou pomocí registru CNTMinMaxCtrlReg.

6.6 Registr CNT0MaxReg (RD)

Tento 24bitový registr obsahuje hodnotu detektoru maxima čítače CNT0 zachycenou pomocí registru CNTMinMaxCtrlReg.

6.7 Registr CNT1MinReg (RD)

Tento 24bitový registr obsahuje hodnotu detektoru minima čítače CNT1 zachycenou pomocí registru CNTMinMaxCtrlReg.

6.8 Registr CNT1MaxReg (RD)

Tento 24bitový registr obsahuje hodnotu detektoru maxima čítače CNT1 zachycenou pomocí registru CNTMinMaxCtrlReg.

6.9 Registr CNT2MinReg (RD)

Tento 24bitový registr obsahuje hodnotu detektoru minima čítače CNT2 zachycenou pomocí registru CNTMinMaxCtrlReg.

6.10 Registr CNT2MaxReg (RD)

Tento 24bitový registr obsahuje hodnotu detektoru maxima čítače CNT2 zachycenou pomocí registru CNTMinMaxCtrlReg.

7. Registry pro obsluhu FIFO paměti

7.1 Úvod

V následujících odstavcích budou popsány registry související s FIFO pamětí, resp. obvody pro detekci události a časovačem pro řízení zápisu do FIFO paměti, viz přehled v 3. kapitole.

Registry lze rozdělit na skupinu určenou pro řízení zápisu do FIFO paměti

ScanCtrlReg	základní řídicí registr zápisů do FIFO paměti
ScanDINPH	detektor změny na digitálním vstupním portu (filtr zákmitů)
ScanDINRE	detektor změny na digitálním vstupním portu (povolení detekce náběžné hrany)
ScanDINFE	detektor změny na digitálním vstupním portu (povolení detekce sestupné hrany)
ScanTimer	časovač pro periodické zápisy do FIFO (dělička 24 bitů se vstupní frekvencí 25 MHz)

a skupinu související se čtením FIFO paměti

FIFONoSmpIStbReg	slouží k zachycení aktuálního zaplnění FIFO paměti (tzn. počtu vzorků uložený ve FIFO)
FIFONoSmpIReg	zachycená hodnota zaplnění FIFO paměti (tzn. počtu vzorků uložený ve FIFO)
FIFOStatusReg	stavový registr
FIFOCNT0Reg	data čtená z FIFO paměti, hodnota čítače CNT0 v okamžik zápisu do FIFO
FIFOCNT1Reg	data čtená z FIFO paměti, hodnota čítače CNT1 v okamžik zápisu do FIFO
FIFOCNT2Reg	data čtená z FIFO paměti, hodnota čítače CNT2 v okamžik zápisu do FIFO
FIFOTimeStamp	data čtená z FIFO paměti, hodnota časové značky uložené spolu s čítači
FIFODINReg	data čtená z FIFO paměti, hodnota digitálního vstupního portu v okamžik zápisu do FIFO

7.2 Funkce FIFO paměti

FIFO je zjednodušeně řečeno fronta dat "první dovnitř - první ven".

Na straně zápisu je funkce řízená registry umožňujícími

- periodický zápis hodnot frekvencí definovanou 24bitovou děličkou se vstupní frekvencí 25 MHz
- zápis řízený změnou stavu vstupního digitálního portu, resp. změnou vybraných bitů

Každá událost popsaná v předešlém odstavci uloží do FIFO paměti jeden záznam složený z hodnot všech čítačů, časové značky (definuje čas detekované události s krokem 1 μ s) a stavu digitálního vstupního portu.

Kapacita FIFO umožňuje uložit 1024 záznamů, v případě pokusu o zápis 1025. záznamu dojde k přetečení paměti a ukončení zápisů.

Počítač musí vyhodnocovat stav zaplnění FIFO (softwarovým pollingem nebo s podporou přerušení) a data z FIFO paměti vyčítat. Není nutné vyčítat všechna data záznamu, nezbytné je však číst registr FIFODINReg; jeho přečtením dojde k odstranění záznamu z FIFO paměti a tedy i uvolnění části paměti pro další zapisované záznamy.

Příklad postupu periodického zápisu frekvencí 1 kHz

Výchozím stavem je nastavení obvodů po zápisu hodnoty 80_H do registru ScanCtrlReg; tímto zápisem dojde k zastavení případného probíhajícího záznamu a současně vynulování obsahu FIFO paměti.

- v prvním kroku je potřeba nastavit frekvenci 1 kHz (zápisem hodnoty 25000 do registru ScanTimer).
- v druhém kroku je potřeba odstartovat záznam (zápisem hodnoty 02_H nebo 82_H do registru ScanCtrlReg); od tohoto okamžiku je spuštěn z nulového stavu generátor časové značky a současně zahájen záznam do FIFO se zvolenou periodou 1 ms (k prvnímu zápisu dojde 1 ms po zápisu do registru ScanCtrlReg)
- následně se musí vyhodnocovat zaplnění FIFO paměti a vyčítat data; program může načíst stav vybraných čítačů a časové značky (v případě periodického zápisu časovačem nemá valný význam), vždy však musí (jako poslední) načíst stav registru FIFODINReg
- k ukončení záznamu dojde zápisem hodnoty 00_H (popř. 80_H) do registru ScanCtrlReg nebo přetečením FIFO paměti

Zápis do FIFO změnami digitálního vstupního portu

Postup je v tomto případě analogický předešlému s tím rozdílem, že namísto časovače je konfigurován detektor změn na digitálním vstupním portu a záznam je startován zápisem hodnoty 01_H nebo 81_H do registru ScanCtrlReg. Čtením časové značky lze v tomto případě určit čas záznamu do FIFO, tzn. čas detekované změny vstupních signálů.

Karta umožňuje navíc současně aktivovat periodický záznam i záznam změnou vstupních signálů.

7.3 Registr ScanCtrlReg (WR)

Tento registr slouží k zahájení zápisů do FIFO, nulování obsahu FIFO a nulování příznaku přetečení (bit D7 registru FIFOStatusReg). Současně se zahájením záznamu (tzn. jeden nebo oba bity D1/D0 jsou nastaveny na hodnotu 1) dojde ke spuštění generátoru časové značky z nulového stavu.

K zastavení zápisů do FIFO dojde vynulováním obou bitů D1/D0 tohoto registru, případně přetečením FIFO paměti (je signalizováno bitem D7 registru FIFOStatusReg).

D7	D6	D5	D4	D3	D2	D1	D0
FIFO_CLR	RSRV	RSRV	RSRV	RSRV	RSRV	RUN_TIM	RUN_DIN

RUN_DIN	aktivuje detektor změn na digitálním vstupním portu a zápis do FIFO vybranými událostmi 0 detektor blokován 1 detektor je aktivní
RUN_TIM	aktivuje časovač pro periodický zápis do FIFO 0 časovač je blokován 1 časovač generuje zápisové pulsy do FIFO
FIFO_CLR	nuluje obsah FIFO a příznak přetečení (viz FIFOStatusReg) 0 bez významu, stav FIFO a příznak není modifikován 1 vynuluje FIFO a příznak (generuje krátký puls, následný zápis 0 není vyžadován)
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

7.4 FIFOStatusReg (RD)

Registr poskytuje informaci o stavu FIFO paměti, resp. probíhajícím záznamu do FIFO.

Dva nejnižší bity signalizují aktuální stav registru ScanCtrlReg, nejvyšší bit slouží k signalizaci přetečení FIFO paměti (při přetečení je ukončen zápis dat do FIFO, doposud zapsaná data jsou platná a lze je načíst).

D7	D6	D5	D4	D3	D2	D1	D0
FIFO_OVFL	RSRV	RSRV	RSRV	RSRV	RSRV	RUN_TIM	RUN_DIN

RUN_DIN	aktuální stav stejnojmenného bitu registru ScanCtrlReg
RUN_TIM	aktuální stav stejnojmenného bitu registru ScanCtrlReg
FIFO_OVFL	signalizuje přetečení FIFO 0 nedošlo k přetečení FIFO 1 došlo k přetečení FIFO (tzn. pokus o zápis 1025. vzorku dat)
RSRV	rezerva (z důvodu dopředné kompatibility je doporučeno hodnoty ignorovat)

Poznámka: Při přetečení je ukončen zápis dat do FIFO (identicky, jako by byly programem vynulovány D0/D1 bity ScanCtrlReg), doposud zapsaná data ve FIFO jsou platná a lze je načíst.

7.5 ScanDINPH (WR)

Registr slouží k povolení filtrace digitálních vstupů (blokuje změny kratší než 300 ns).

Význam má pouze nejnižší bit registru; nastavením hodnoty 1 se povoluje filtrace všech osmi signálů digitálního portu, hodnotou 0 se filtrace vypíná. Každý signál je filtrovaný samostatně. Z důvodu dopředné kompatibility je pro všechny ostatní bity doporučena hodnota 0.

7.6 ScanDINRE (WR)

Registr slouží k povolení detekce náběžných hran na osmibitovém digitálním vstupním portu.

Každý bit registru hodnotou 1 povoluje detekci náběžné hrany na odpovídajícím signálu digitálního portu, hodnotou 0 se detekce blokuje.

7.7 ScanDINFE (WR)

Registr slouží k povolení detekce sestupných hran na osmibitovém digitálním vstupním portu.

Každý bit registru hodnotou 1 povoluje detekci sestupné hrany na odpovídajícím signálu digitálního portu, hodnotou 0 se detekce blokuje.

7.8 ScanTimer (WR)

Registr slouží k nastavení časovače frekvence pro periodické zápisy do FIFO (viz ScanCtrlReg , bit RUN_TIM).

Časovač pracuje jako dělička se vstupní frekvencí 25 MHz s nastavitelným poměrem 1:N, kde N je 24bitové číslo zapsané do registru, nejnižší frekvence je tedy $25.000.000 : 16.777.215 = 1,49$ Hz.

Poznámka: *Nejvyšší reálně použitelná frekvence je limitována dobou obsluhy FIFO paměti; při kapacitě FIFO 1024 záznamů a zvolené mezní době reakce 5 ms na příznak zaplnění 50% (záleží na způsobu detekce příznaku, na operačním systému a řadě dalších podmínek) lze pracovat s frekvencí do 100 kHz.*

Poznámka: *Výstup časovače lze směřovat na digitální výstup RTDOUT3 (viz popis v příslušné kapitole); na výstupu je generován signál se střídou 1:1 (k zápisu do FIFO dochází sestupnou hranou tohoto signálu). Tento signál je generován kdykoliv je bit RUN_TIM registru ScanCtrlReg nastaven do úrovně 1 a časovač nastaven na platnou frekvenci (tzn. dělicí poměr 2 a vyšší); signál generátoru lze tedy využít pro obecné aplikace i v případě, kdy záznam do FIFO není používán (obsah FIFO tedy nemusí být čten, pouze je potřeba blokovat přerušení odvozené od zaplnění FIFO).*

7.9 FIFONoSmpIStbReg (WR)

Registr slouží k zachycení aktuálního stavu zaplnění FIFO paměti do registru FIFONoSmpIReg, obsah zapisovaných dat je nevýznamný.

7.10 FIFONoSmpIReg (RD)

Registr poskytuje informaci o stavu zaplnění FIFO paměti v rozsahu 0 až 1024; obsah tohoto registru je aktualizován zápisem do registru FIFONoSmpIStbReg.

7.11 FIFOCNT0Reg (RD), FIFOCNT1Reg (RD), FIFOCNT2Reg (RD), FIFOTimeStamp (RD), FIFODINReg (RD)

Registry slouží pro přístup k datům uložených ve FIFO paměti.

Pro zjištění počtu vzorků uložených v paměti je určen registr FIFONoSmpIReg, vzorkem se rozumí sada hodnot všech tří čítačů, časové značky (čítač 24 bitů se vstupní frekvencí 1 MHz spuštěný z nulového stavu v okamžik startu záznamu jedním nebo oběma bity D0/D1 ScanCtrlReg) a stavu digitálního vstupu uložená do FIFO v okamžik detekce.

Obsah všech tří registrů FIFOCNTxReg a FIFOTimeStampReg lze číst v libovolném pořadí, mohou být čteny opakovaně a případně nemusí být čteny vůbec. Čtení registru FIFODINReg je naopak povinné, protože jeho čtením dojde k odstranění vzorku z FIFO paměti (registr tedy musí být čten jako poslední).

8. Registry pro obsluhu přerušení

8.1 Úvod

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

Registry lze rozdělit na skupinu vyhrazenou pro konfiguraci obvodů přerušení

INTEnReg	aktivace obvodů pro řízení signálu přerušení PCI sběrnice (INTA)
IRQCfgReg	povolení zdrojů přerušení
IRQStatusReg	příznaky zdrojů přerušení
IRQClrReg	nulování příznaků přerušení
CMPIRQReg	povolení přerušení od komparátorů
FIFOIRQReg	nastavení prahu pro vyvolání přerušení
TimerReg	generátor časových značek pro periodické vyvolávání přerušení

a skupinu registrů souvisejících s funkcemi popsanými v předešlých kapitolách

CMPEnReg	viz kapitola s popisem obsluhy komparátorů
CMPStatReg	viz kapitola s popisem obsluhy komparátorů
CMPClrReg	viz kapitola s popisem obsluhy komparátorů
CNTXSTREnReg	viz kapitola s popisem obsluhy čítačů
CNTXSTRStatusReg	viz kapitola s popisem obsluhy čítačů
CNTXSTRClrReg	viz kapitola s popisem obsluhy čítačů

8.2 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.

Zachycení hodnot čítačů

Umožňuje vyvolat přerušení zachycením hodnoty čítačů do vyrovnávacích registrů signálem EXT-IN.

Obvody komparátorů čítačů

Umožňuje vyvolat přerušení nastavením příznaků komparátorů hodnot čítačů.

Obvody pro řízení FIFO

Umožňuje vyvolat přerušení při definovaném zaplnění FIFO paměti.

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.

Totéž platí pro související registry, programová obsluha přerušení musí např. vyčistit obsah FIFO pod nastavenou prahovou úroveň, vynulovat příznaky komparátorů (registr CMPClrReg) apod.

8.3 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)

8.4 Registr IRQCfgReg (WR)

Tento registr slouží k povolení základních zdrojů přerušení.

D7	D6	D5	D4	D3	D2	D1	D0
FIFO	XSTR	CMP	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
CMP	povoluje vyvolání přerušení odvozené od komparátorů čítačů (tzn. hodnoty registru CMPStatReg) 0 záchytný registr navázaný na registr CMPStatReg je blokován 1 funkce záchytného registru je aktivována; přerušení je vyvoláno přechodem logickému součinu stavu registrů CMPStatReg & CMPIRQReg z nulové hodnoty do nenulové
XSTR	povoluje vyvolání přerušení odvozené od zachycení hodnoty čítačů do vyrovnávacích registrů signálem EXT-IN (tzn. hodnoty registru CNTXSTRStatusReg) 0 záchytný registr navázaný na registr CNTXSTRStatusReg je blokován 1 funkce záchytného registru je aktivována; přerušení je vyvoláno přechodem stavu registru CNTXSTRStatusReg z nulové hodnoty do nenulové
FIFO	povoluje vyvolání přerušení odvozené od dosažení prahu zaplnění FIFO daty 0 záchytný registr navázaný na detekci zaplnění FIFO je blokován 1 funkce záchytného registru je aktivována; přerušení je vyvoláno dosažením nastaveného zaplnění FIFO paměti (viz FIFOIRQReg)
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

8.5 Registr IRQStatusReg (RD)

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

D7	D6	D5	D4	D3	D2	D1	D0
FIFO	XSTR	CMP	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
CMP	stav záchytného registru navázaného na komparátory čítačů 0 registr není nastaven, tzn. od posledního nulování nedošlo k detekci shody komparátorem 1 registr je nastaven, tzn. od posledního nulování došlo k detekci shody komparátorem
XSTR	stav záchytného registru navázaného na obvody zachycení hodnot čítačů signálem EXT-IN 0 registr není nastaven, tzn. od posledního nulování nedošlo k zachycení hodnot čítačů 1 registr je nastaven, tzn. od posledního nulování došlo k zachycení hodnot čítačů
FIFO	stav záchytného registru navázaného na obvody zaplnění FIFO paměti 0 registr není nastaven, tzn. od posledního nulování nedošlo k zaplnění FIFO paměti 1 registr je nastaven, tzn. od posledního nulování došlo k zaplnění FIFO paměti
RSRV	rezerva (z důvodu dopředné kompatibility je doporučeno hodnoty ignorovat)

8.6 Registr IRQClrReg (WR)

Tento registr slouží nulování záchytných registrů povolovaných registrem IRQCfgReg.

D7	D6	D5	D4	D3	D2	D1	D0
FIFO	XSTR	CMP	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)
CMP	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)
XSTR	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)
FIFO	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)

8.7 Registr TimerReg (WR, RD)

Tento registr slouží k ovládní 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.

8.8 Registr CMPIRQReg (WR)

Tento registr slouží k povolení vyvolání přerušení komparátory, resp. obsahem registru CMPStatReg.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	IRQ_CMP22	IRQ_CMP12	IRQ_CMP02	RSRV	IRQ_CMP21	IRQ_CMP11	IRQ_CMP01

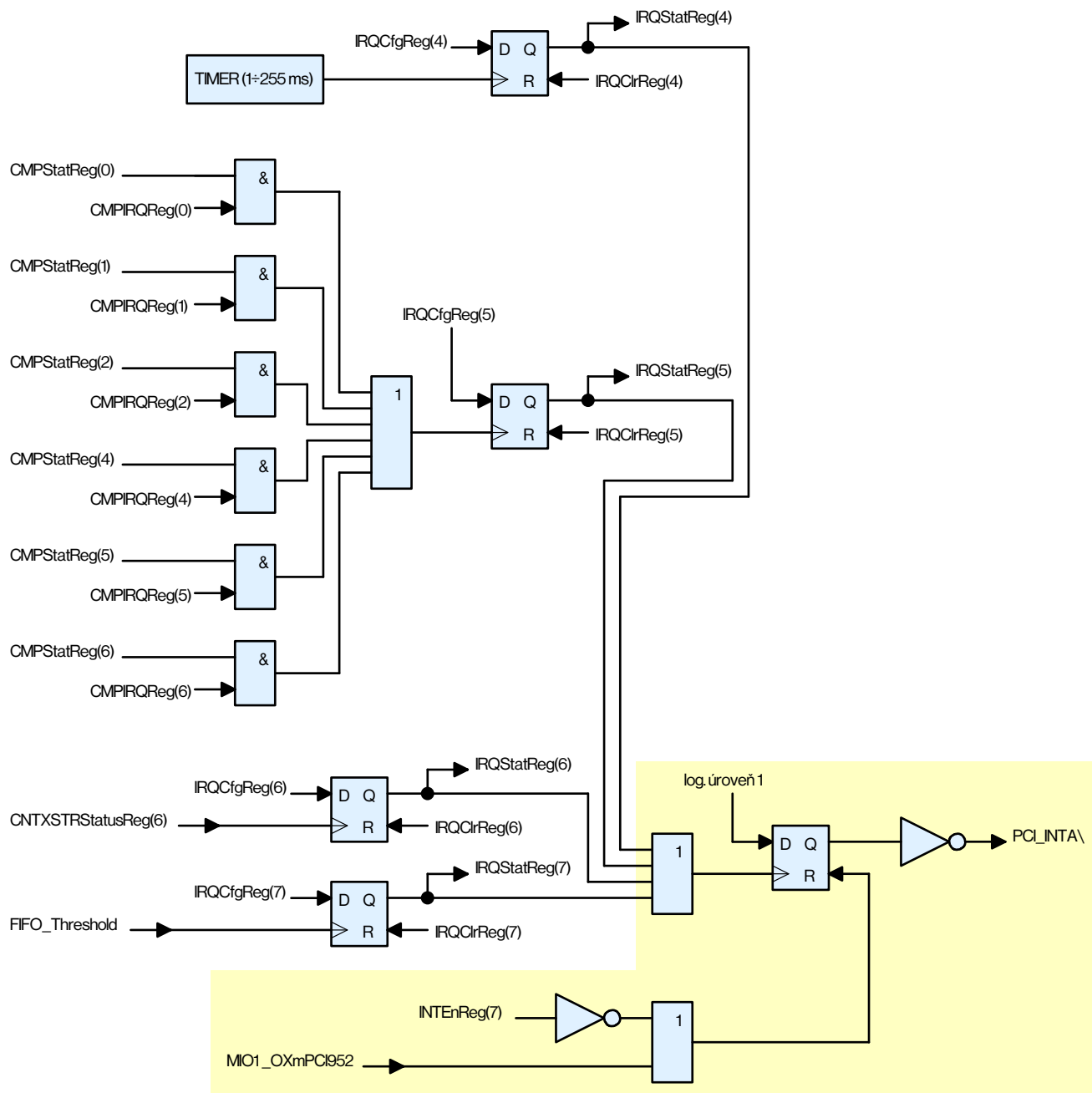
IRQ_CMP01	povoluje vyvolání přerušení odvozené od záchytného registru 1. komparátoru čítače CNT0
0	nastavení záchytného registru v IRQCfgReg blokováno
1	nastavení záchytného registru v IRQCfgReg povoleno
IRQ_CMP11	povoluje vyvolání přerušení odvozené od záchytného registru 1. komparátoru čítače CNT1
0	nastavení záchytného registru v IRQCfgReg blokováno
1	nastavení záchytného registru v IRQCfgReg povoleno
IRQ_CMP21	povoluje vyvolání přerušení odvozené od záchytného registru 1. komparátoru čítače CNT2
0	nastavení záchytného registru v IRQCfgReg blokováno
1	nastavení záchytného registru v IRQCfgReg povoleno
IRQ_CMP02	povoluje vyvolání přerušení odvozené od záchytného registru 2. komparátoru čítače CNT0
0	nastavení záchytného registru v IRQCfgReg blokováno
1	nastavení záchytného registru v IRQCfgReg povoleno
IRQ_CMP12	povoluje vyvolání přerušení odvozené od záchytného registru 2. komparátoru čítače CNT1
0	nastavení záchytného registru v IRQCfgReg blokováno
1	nastavení záchytného registru v IRQCfgReg povoleno
IRQ_CMP22	povoluje vyvolání přerušení odvozené od záchytného registru 2. komparátoru čítače CNT2
0	nastavení záchytného registru v IRQCfgReg blokováno
1	nastavení záchytného registru v IRQCfgReg povoleno
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

8.9 Registr FIFOIRQReg (WR)

Registr FIFOIRQReg umožňuje nastavit práh zaplnění FIFO, při jehož dosažení dojde k nastavení příznaku zaplnění a vyvolání přerušení. Zapisovaná hodnota je v rozsahu $0 \div 10$ a představuje práh 2^N (tzn. v rozsahu $1 \div 1024$).

8.10 Schéma registrové struktury

Na níže uvedeném obrázku je pro lepší porozumění souvislosti 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.

9. Registry digitálních vstupů/výstupů a RT digitálních výstupů

9.1 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

a konfigurační registry

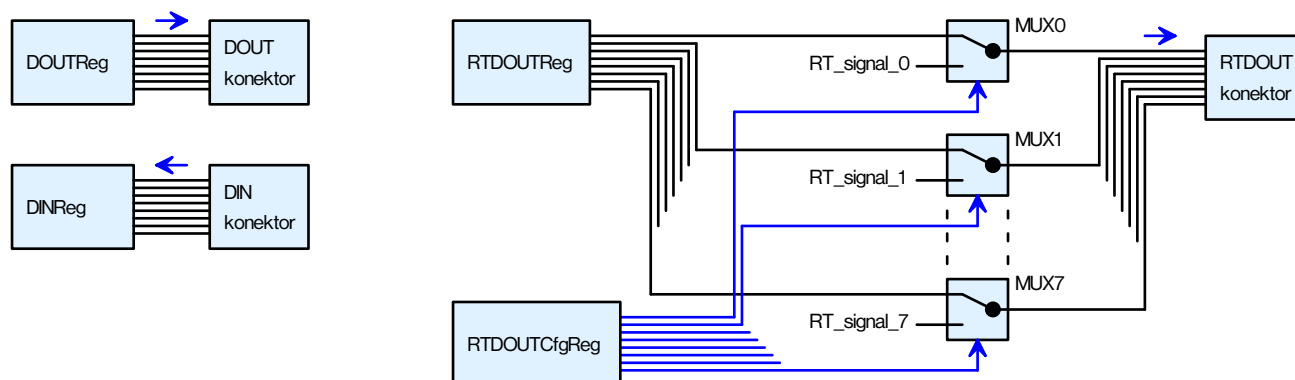
RTDOUTCfgReg	konfigurační registr RT výstupního portu
--------------	--

9.2 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).

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.



9.3 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).

9.4 Registr DOUTReg (WR)

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

9.5 Registr RTDOUTReg (WR)

Tento registr slouží k ovládní 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.

9.6 Registr RTDOUTCfgReg (WR)

Tento registr slouží k přepínání alternativních signálů na konektor RT digitálních výstupů (viz obrázek výše).

D7	D6	D5	D4	D3	D2	D1	D0
RSRV	RTCfg6	RTCfg5	RTCfg4	RTCfg3	RTCfg2	RTCfg1	RTCfg0

RTCfg0	konfigurační registr RT digitálního výstupu, signál 0 0 na výstup je směrován obsah RTDOUTReg, bit 0 1 na výstup je směrován obsah CMPStatReg, bit 0
RTCfg1	konfigurační registr RT digitálního výstupu, signál 1 0 na výstup je směrován obsah RTDOUTReg, bit 1 1 na výstup je směrován obsah CMPStatReg, bit 1
RTCfg2	konfigurační registr RT digitálního výstupu, signál 2 0 na výstup je směrován obsah RTDOUTReg, bit 2 1 na výstup je směrován obsah CMPStatReg, bit 2
RTCfg3	konfigurační registr RT digitálního výstupu, signál 3 0 na výstup je směrován obsah RTDOUTReg, bit 3 1 na výstup je směrován výstup časovače ScanTimer (pulsy se střídou 1:1; k zápisu do FIFO dochází sestupnou hranou tohoto signálu) (další informace viz kapitola s popisem obsluhy FIFO)
RTCfg4	konfigurační registr RT digitálního výstupu, signál 4 0 na výstup je směrován obsah RTDOUTReg, bit 4 1 na výstup je směrován obsah CMPStatReg, bit 4
RTCfg5	konfigurační registr RT digitálního výstupu, signál 5 0 na výstup je směrován obsah RTDOUTReg, bit 5 1 na výstup je směrován obsah CMPStatReg, bit 5
RTCfg6	konfigurační registr RT digitálního výstupu, signál 6 0 na výstup je směrován obsah RTDOUTReg, bit 6 1 na výstup je směrován obsah CMPStatReg, bit 6
RSRV	rezerva (z důvodu dopředné kompatibility je doporučena hodnota 0)

10. Pomocné diagnostické registry

10.1 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.

10.2 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.

10.3 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.

10.4 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.

10.5 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-7303C/E je uvedena v 1. kapitole.

10.6 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-7303C/E je uvedena v 1. kapitole.

11. Servisní rozhraní

11.1 Úvod

V následujících odstavcích bude popsán komunikační protokol servisního rozhraní.

Ačkoliv servisní rozhraní je primárně určeno pro programování flash paměti, má implementována 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.

11.2 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.

11.3 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:

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

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".

11.4 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	zapiše 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	zapiše 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 nereguluje na dotazy až do vykonání obsluhy povelu). Všechny ostatní povely jsou nejprve vykonány a až následně je vyslána odpověď.

11.5 Povel identifikace

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

struktura dotazu	struktura odpovědi
{ 0 0 0 0 }	[0 0 I... I S S]
{ = start_značka 0 0 = povel 0 0 = kontrolní číslo (uvedena přímo hodnota) } = stop_značka	[= start_značka 0 0 = povel I...I = 192 ASCII HEX znaků S S = 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-7303C          typ karty
173033001         výrobní číslo karty (shodné se štítkem na kartě)
PCIloader v.1.15  identifikace firmware mikropočítače
FPGA-17-0.7      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.

11.6 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

11.7 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

11.8 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
{ 0APPPPN }	[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

11.9 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 }	[0BPPPPVSS]
{ = 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

11.10 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
{ 0 C P P P D ... D N N }	[0 C P P P D ... D S S]
{ = start_značka 0 C = povel P P P P = 4 ASCII HEX znaky (16bitové číslo stránky) D...D = 512 ASCII HEX znaků (256 B dat stránky) S S = kontrolní číslo } = stop_značka	[= start_značka 0 C = povel P P P P = 4 ASCII HEX znaky (16bitové číslo stránky) D...D = 512 ASCII HEX znaků (256 B dat stránky) S S = kontrolní číslo] = stop_značka

11.11 Rozdělení flash paměti

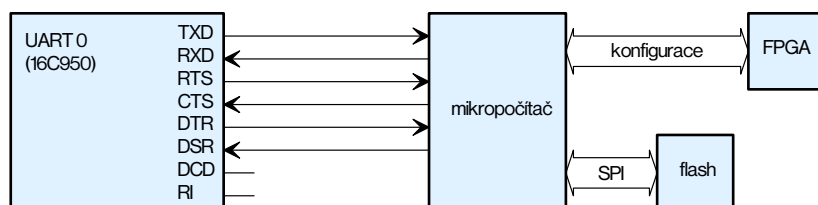
Flash má kapacitu 512 kB rozdělenou do 2048 stránek velikosti 256 B.

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)

11.12 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)

11.13 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 přerušování 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 básová 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	úroveň 1 signalizuje přítomnost dat v přijímacím zásobníku (tzn. lze 1x číst znak z adresy ADDR+0)
OverR_Err	úroveň 1 signalizuje přetečení zásobníku přijímaných dat (fatální chyba přenosu)
Par_Err	úroveň 1 signalizuje chybu parity přijímaných dat (fatální chyba přenosu)
Fram_Err	úroveň 1 signalizuje chybu stopbitu přijímaných dat (fatální chyba přenosu)
RX_break	úroveň 1 signalizuje chybu "break" přijímaných dat (fatální chyba přenosu)
THR_empty	úroveň 1 signalizuje prázdný vysílací zásobník (tzn. lze 1x zapsat znak na adresu ADDR+0)
TX_empty	úroveň 1 signalizuje prázdný vysílací zásobník a navíc i vyslaný poslední znak
Data_Err	úroveň 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 sběrnic 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 nadřazeným systémem rozhraním RS-485, RS-422, RS-232 nebo LAN
- protokoly AlBus-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 sběrnic 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 typů mikropočítačových desek
- speciální elektronika vybavená rozhraním a technologiemi, se kterými již máme zkušenosti
 - počítačové sběrnic PCI, PCI Express, ISA, USB, ethernet, ...
 - rutinní vyžívání hradlových polí FPGA a CPLD Altera
 - mikropočítače Cortex M3/M4, výkonné mikropočítače řady ˆ51
 - 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>



Member of PCI Special Interest Group

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>