

PCA-7428CL/CS/CE

programátorská příručka

historie dokumentu:		
datum	verze	změny
28.6.2013	06.2013	výchozí verze
16.9.2013	09.2013	pracovní verze
2.10.2013	10.2013	finální verze
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.

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. Blokové schéma vnitřních obvodů karty

- 3.1 Úvod
- 3.2 Schéma
- 3.3 Popis schématu

4. Funkční registry

- 4.1 Přehled registrů
- 4.2 Rozdělení registrů
- 4.3 Registry s délkou větší než 8 bitů

5. Registry digitálních vstupů a výstupů

- 5.1 Úvod
- 5.2 Funkce digitálních portů
- 5.3 Registr DINReg (RD)
- 5.4 Registr DINEExtReg (RD)
- 5.5 Registr DOUTReg (WR)

6. Registry analogových výstupů

- 6.1 Úvod
- 6.2 Funkce analogových výstupů
- 6.3 Registr DAC0Reg (WR)
- 6.4 Registr DAC1Reg (WR)
- 6.5 Registr DACRangeReg (RD)

7. Registry pro obsluhu čítačů kompatibilních s PCA-7428A

- 7.1 Úvod
- 7.2 Funkce XCNT čítačů
- 7.3 XCNTEnReg (WR)
- 7.4 XCNTCtrlReg (WR)
- 7.5 Registr XCNT0SetReg (WR)
- 7.6 Registr XCNT1SetReg (WR)

8. Registry pro obsluhu programovatelných čítačů

- 8.1 Úvod
- 8.2 Funkce čítačů
- 8.3 Registrová struktura
- 8.4 Registr CNTSelReg (WR)
- 8.5 Registr CNTEnReg (WR)
- 8.6 Registr CNTCtrlReg (WR)
- 8.7 Registr CNTXSTREnReg (WR)
- 8.8 Registr CNTXSTRStatusReg (RD)
- 8.9 Registr CNTXSTRClrReg (WR)
- 8.10 Registr CNT0CWReg (WR, CNTSelReg=0000)
- 8.11 Registr CNT0StatReg (RD, CNTSelReg=0000)
- 8.12 Registr CNT0SetReg (WR)
- 8.13 Registr CNT0RngReg (WR, CNTSelReg=0001)
- 8.14 Registr CNT0StrReg (RD)
- 8.15 Registr CNT0XStrReg (RD, CNTSelReg=0001)
- 8.16 Schéma registrové struktury čítače

9. Registry pro obsluhu scanovacích obvodů a FIFO pamětí

- 9.1 Úvod
- 9.2 Funkce scanovacích obvodů
- 9.3 Princip a funkce FIFO paměti
- 9.4 Registr CWReg (WR)
- 9.5 StatusReg (RD)
- 9.6 SWTrigReg (WR)
- 9.7 SWTrigStatusReg (RD)
- 9.8 SWFIFODataReg (RD)
- 9.9 FIFONoSmpStrbReg (WR)
- 9.10 FIFONoSmpReg (RD)
- 9.11 FIFODataReg (RD)
- 9.12 ScanAdrReg (WR)
- 9.13 ScanDataReg (WR/RD)
- 9.14 Struktura scanovacích parametrů
- 9.15 Programová obsluha scanovacích obvodů
- 9.16 Porovnání pracovních režimů

10. Registry pro obsluhu přerušování

- 10.1 Úvod
- 10.2 Funkce obvodů pro obsluhu přerušování
- 10.3 Registr INTEnReg (WR)
- 10.4 Registr IRQCfgReg (WR)
- 10.5 Registr IRQStatusReg (RD)
- 10.6 Registr IRQClrReg (WR)
- 10.7 Registr TimerReg (WR, RD)
- 10.8 Registr FIFOIRQReg (WR)
- 10.9 Schéma registrové struktury

11. Pomocné diagnostické registry

- 11.1 Úvod
- 11.2 Registr FreeRunCNTReg (RD)
- 11.3 Registr FreeRunCNTStrbReg (WR)
- 11.4 Registr CardIDReg (RD)
- 11.5 Registr FPGATypeReg (RD)
- 11.6 Registr FPGAVerReg (RD)
- 11.7 Registr ResetReg (WR)
- 11.8 Rozhraní bloku inicializačních dat
- 11.9 Registr CalibAdrReg (WR, RD)
- 11.10 Registr CalibDataReg (WR, RD)
- 11.11 Registr CalibCtrlReg (WR)
- 11.12 Registr CalibStatReg (RD)
- 11.13 Blok s kalibračními a inicializačními konstantami
- 11.14 Formát konstant a kalibrace A/D převodníku
- 11.15 Formát konstant a kalibrace D/A převodníků

12. Servisní rozhraní

- 12.1 Úvod
- 12.2 Zapojení UARTů
- 12.3 Struktura povelů
- 12.4 Přehled povelů
- 12.5 Povel identifikace
- 12.6 Povel restart mikropočítače
- 12.7 Povel rekonfigurace FPGA
- 12.8 Povel čtení flash paměti
- 12.9 Povel zápis flash paměti
- 12.10 Povel zápis flash paměti se zpětným čtením
- 12.11 Rozdělení flash paměti
- 12.12 Propojení UARTu s mikropočítačem
- 12.13 Popis programové obsluhy UARTu

1. Základní informace

1.1 Úvod

Tato programátorská příručka navazuje na uživatelskou příručku karet PCA-7428CL/CS/CE (dále společně označovány jako PCA-7428C) 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 systém Linux).

1.2 Verze firmware

Aktuální verze firmware v době vydání příručky:

mikropočítač - firmware:	PCIloader v.1.20	
FPGA - typ firmware:	29	(reprezentováno hodnotou 1D _H)
FPGA - verze firmware:	1.6	(reprezentováno hodnotou 10 _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 PCA-7428C. 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 PCA-7428C 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 mikroprocesorem 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	0240 _H (PCA-7428CL) 0242 _H (PCA-7428CS) 0244 _H (PCA-7428CE)	0241 _H (PCA-7428CL) 0243 _H (PCA-7428CS) 0245 _H (PCA-7428CE)
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, 8 B, adresu přidělí BIOS)	lokální sběrnice (I/O, 256 B, adresu přidělí BIOS)
17 _H ÷14 _H	BAR1	UART1 (I/O, 8 B, adresu přidělí BIOS)	lokální sběrnice (MEM, 4 kB, adresu přidělí BIOS)
1B _H ÷18 _H	BAR2	nevyužito	konfigurační registry řadiče (I/O, 32 B, adresu přidělí BIOS)
1F _H ÷1C _H	BAR3	nevyužito	konfigurační registry řadiče (MEM, 4 kB, 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 následujících 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 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 funkčním registrům karty ...

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

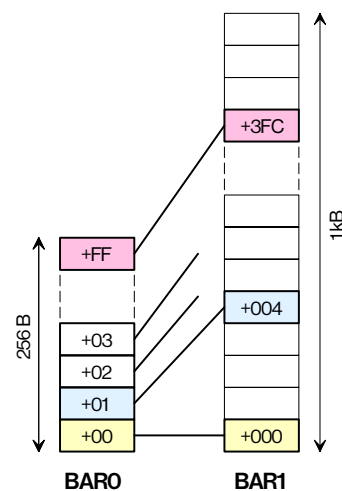
Odlišuje se nějak mapování funkčních registrů v BAR0 a BAR1 PCI funkce F1 karty ...

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 256 B 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 1 kB 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.



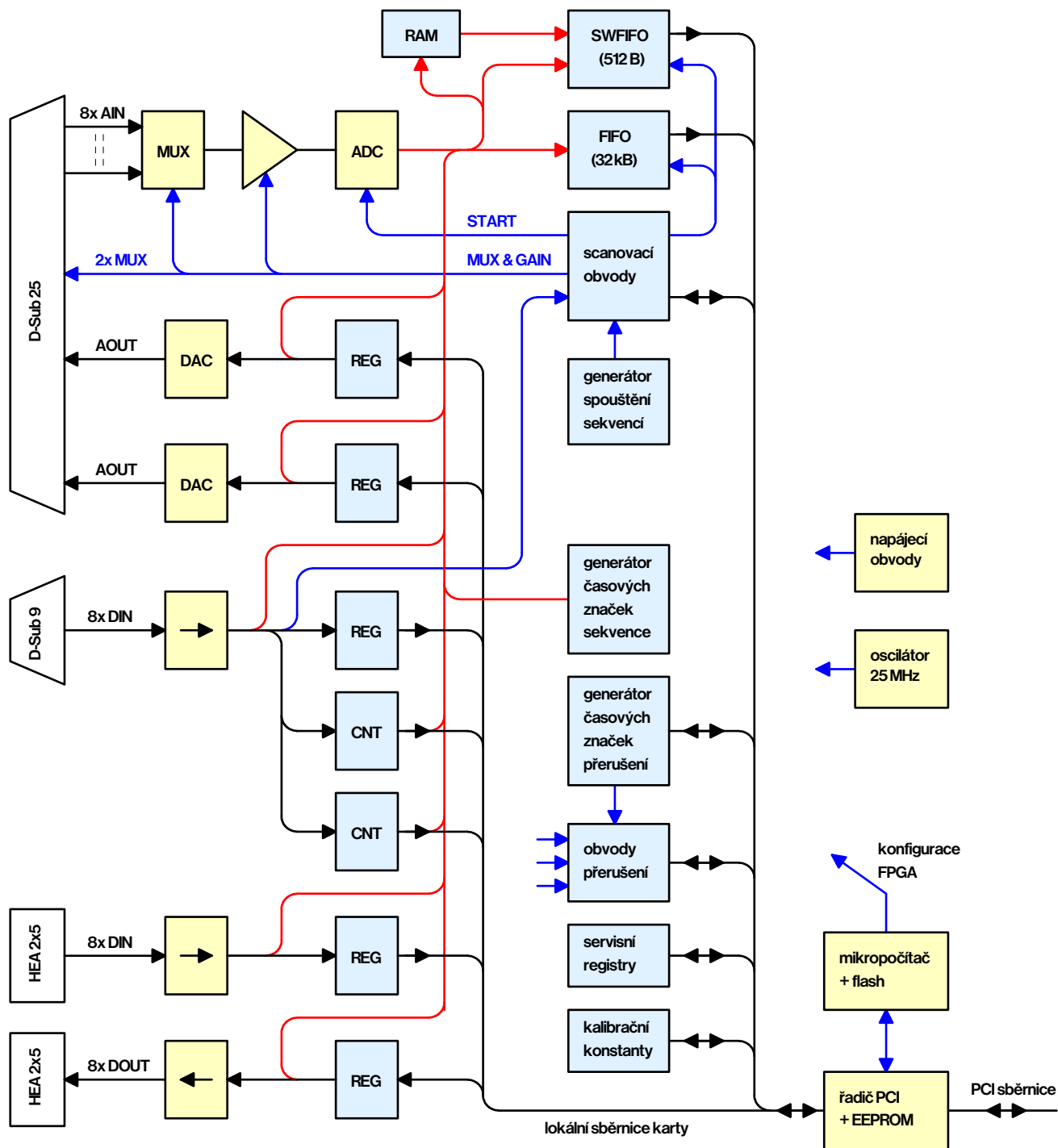
3. Blokové schéma vnitřních obvodů karty

3.1 Úvod

Tato kapitola je věnována stručnému popisu funkcí jednotlivých obvodů karty, struktuře datových tras apod. s cílem napomoci pochopení souvislostí registrů popisovaných v následujícím textu.

3.2 Schéma

Na níže uvedeném obrázku je zakresleno zjednodušené schéma vnitřní struktury karty.



Všechny obvody zakreslené modrou výplní jsou součástí FPGA (jejich funkci lze modifikovat úpravou firmwaru), obvody se žlutou výplní jsou pak tvořeny součástkami mimo FPGA. Datové trasy zakreslené černými čarami představují přístup běžným čtením nebo zápisem registrů, červenými čarami jsou pak zakresleny datové přenosy řízené scanovacími obvody. Modrou barvou jsou zakresleny významné řídicí signály.

3.3 Popis schématu

Osm analogových vstupů je přepínáno multiplexerem (MUX) na vstup jednoho zesilovače s programovatelným zesílením s navazujícím A/D převodníkem (ADC). Základních osm vstupů lze rozšířit na 32 pomocí dvou řídicích signálů vyvedených na konektor D-Sub 25 spolu s analogovými vstupy.

Na konektor D-Sub 25 jsou rovněž vyvedeny signály dvou D/A převodníků (DAC). Oba převodníky jsou řízeny samostatnými registry (REG).

Osm digitálních signálů konektoru D-Sub 9 je zpracováno vstupním oddělovačem a vedeno do FPGA; signály plní nejen funkci běžných vstupních digitálních portů, nýbrž jsou využity i jako vstupy čítačů, resp. vstupy pro externí spouštění scanovací sekvence nebo zachycení hodnoty čítačů a vyvolání přerušování systému.

Další signály jsou vyvedeny na dva konektory umístěné v zadní části karty; jeden z konektorů je připojen na osminásobný vstupní oddělovač (signály jsou využity jako běžný vstupní digitální port), druhý na osminásobný výstupní oddělovač (signály jsou využity jako běžný výstupní digitální port).

K datům všech popsaných I/O obvodů lze přistupovat současně běžnými registry (černé trasy) a záznamem do FIFO paměti (červené trasy); jedinou výjimku tvoří obvody analogových vstupů, jejichž data jsou dostupná výhradně záznamem do FIFO paměti.

Další pomocné obvody slouží k řízení datových toků, umožňují vyvolávat přerušování systému nebo přenášejí různé servisní informace.

Bližší popis si zasluhují scanovací obvody umožňující

- definovat scanovací sekvence (tedy sestavy vstupů, které je potřeba měřit, resp. zaznamenávat, ideálně v nekratším možném časovém rozestupu); scanovat lze nejen vstupní periferie (analogové vstupy, digitální vstupy a čítače), ale také aktuální stav výstupních periferií (analogové výstupy a digitální výstupy) a navíc i vyhrazený generátor časové značky sekvence (umožňuje k uloženým datům přiřadit i přesný časový údaj okamžiku zápisu; má význam především v režimu spouštění externím signálem)
- řídit A/D převodník, programovatelný zesilovač a multiplexer
- zpracovávat startovací události (časovač pro periodické spouštění, zpracování signálů externího digitálního vstupu apod.)
- zaznamenávat data do dvou FIFO pamětí (lze i současně); první označená SWFIFO s kapacitou 512 B je určena pro záznam v režimu softwarového spouštění, resp. v režimu softwarového požadavku o přenos naměřených dat ze zásobníku RAM, druhá označená FIFO s kapacitou 32 kB je určena pro záznam v režimu spouštění časovačem nebo externím signálem
- generovat stavové a chybové příznaky

4. Funkční registry

4.1 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)
+008		DINExtReg
+044/040	DAC0Reg	(zpětné čtení)
+04C/048	DAC1Reg	(zpětné čtení)
+084/080	XCNT0SetReg	
+08C/088	XCNT1SetReg	
+090	XCNTCtrlReg	
+094	XCNTENReg	(zpětné čtení)
+180	IRQCfgReg	IRQStatusReg
+184	IRQClrReg	
+18C	INTEnReg	(zpětné čtení)
+1A0	FIFONoSmpIStbReg	FIFONoSmpReg
+1A4	FIFOIRQReg	
+1AC		FIFODataReg
+1C0	CWReg	StatusReg
+1C4	SWTrigReg	SWTrigStatusReg
+1C8		SWFIFODataReg
+1E8	ScanAdrReg	(zpětné čtení)
+1FC/1F8/1F4/1F0	ScanDataReg	(zpětné čtení)
+20C/208/204/200	CNT0SetReg	CNT0StrReg
+21C/218/214/210	CNT0CWReg (pro CNTSelReg=0000) CNT0RngReg (pro CNTSelReg=0001)	CNT0StatReg (pro CNTSelReg=0000) CNT0XStrReg (pro CNTSelReg=0001)
+22C/228/224/220	CNT1SetReg	CNT1StrReg
+23C/238/234/230	CNT1CWReg (pro CNTSelReg=0000) CNT1RngReg (pro CNTSelReg=0001)	CNT1StatReg (pro CNTSelReg=0000) CNT1XStrReg (pro CNTSelReg=0001)
+304/300	CNTEnReg	(zpětné čtení)
+30C/308	CNTCtrlReg	(zpětné čtení)
+320	CNTSelReg	(zpětné čtení)
+338	CNTXSTREnReg	CNTXSTRStatusReg
+33C	CNTXSTRClrReg	
+3C4/3C0	CalibAdrReg	(zpětné čtení)
+3C8	CalibDataReg	CalibDataReg
+3CC	CalibCtrlReg	CalibStatReg
+3D0		DACRangeReg
+3EC/3E8/3E4/3E0	FreeRunCNTStbReg	FreeRunCNTReg (32 bitů @ 100 kHz)
+3F0	TimerReg	TimerReg
+3F4		CardIDReg
+3F8		FPGATypeReg
+3FC	ResetReg	FPGAVerReg

Upozornění: Registry *CNT0RngReg* a *CNT1RngReg* jsou po startu karty (resp. konfiguraci nebo resetu FPGA) nastaveny na hodnotu $FFFFFF_{16}$, 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 předešlým programem. Kalibrační konstanty, stav analogových výstupů a stav digitálních výstupů jsou nastaveny hodnotami přenesenými z flash paměti karty. Stav registrů shodný jako po startu karty lze také nastavit zápisem do registru *ResetReg*.

4.2 Rozdělení registrů

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

oranžová	registry digitálních vstupů/výstupů
tmavě modrá	registry analogových výstupů
světle modrá	registry 16bitových čítačů zpětnou kompatibilitu s PCA-7428A
šedá	registry 32bitových programovatelných čítačů s enkodérem
červená	registry určené pro obsluhu scanovacích obvodů a FIFO paměti
žlutá	registry související s přerušením systému
zelená	pomocné servisní a 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).

4.3 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 ...

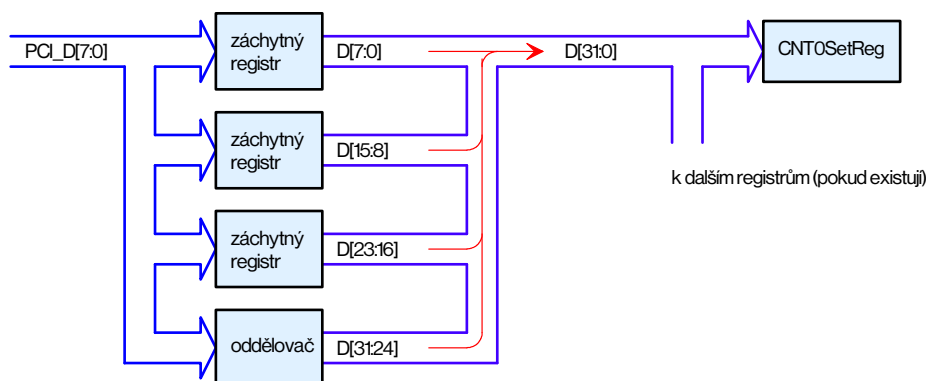
- osmibitový registr s nejnižší adresou obsahuje nejnižší bity vícebytového registru a osmibitový registr s nejvyšší adresou obsahuje nejvyšší bity vícebytového registru
- čtení i zápis osmibitových registrů musí probíhat v pořadí od registru s nejnižší adresou po registr s nejvyšší adresou
- zapisovány musí být vždy všechny osmibitové registry příslušné jednomu vícebytovému registru
- čtení musí být vždy nejnižší registr a pak potřebné vyšší (tzn. není nutné číst všechny, pokud je obsah nevýznamný)

Pro příklad postup čtení a výpočtu 32bitové hodnoty registru CNT0StrReg:

1. čtení B_0 = obsah registru na adrese $+200_H$
 2. čtení B_1 = obsah registru na adrese $+204_H$
 3. čtení B_2 = obsah registru na adrese $+208_H$
 4. čtení B_3 = obsah registru na adrese $+20C_H$
- výpočet $\text{výsledná hodnota} = 16777216 * B_3 + 65536 * B_2 + 256 * B_1 + B_0$

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 32bitových 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 $+204_H$), v třetím cyklu se zapisuje do záchytného registru s významem D[23:16] (pro CNT0SetReg jde o adresu $+208_H$) a ve čtvrtém cyklu při zápisu dat s významem D[31:24] (pro CNT0SetReg jde o adresu $+20C_H$) se do odpovídajícího funkčního registru přenáší všech 32 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.

5. Registry digitálních vstupů a výstupů

5.1 Úvod

V následujících odstavcích budou popsány registry související s digitálními vstupy a výstupy, viz přehled ve 4. kapitole.

Registry lze dále rozdělit na skupinu datových registrů

DINReg	registr vstupního portu
DINExtReg	registr vstupního portu plnicího současně funkci vstupů čítačů (tzn. D-Sub 9 na panelu karty)
DOUTReg	registr výstupního portu

5.2 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).

5.3 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).

5.4 Registr DINExtReg (RD)

Tento registr slouží ke čtení stavu vstupního digitálního portu plnicího současně funkci vstupů čítačů, každý bit registru zpřístupňuje jeden signál osmibitového digitálního portu (nejnižší bit DINExt0, nejvyšší bit DINExt7).

5.5 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. Registry analogových výstupů

6.1 Úvod

V následujících odstavcích budou popsány registry související s analogovými výstupy, viz přehled ve 4. kapitole.

Registry lze dále rozdělit na skupinu datových registrů

DAC0Reg	registr analogového výstupu 0
DAC1Reg	registr analogového výstupu 1
DACRangeReg	pomocný registr pro detekci zvolených pracovních rozsahů (čte stav konfiguračních propojek)

Upozornění: Analogovými výstupy je vybavena pouze verze PCA-7428CS, karty verze CL a CE sice mají registry popsané v této kapitole implementovány, nejsou však osazeny D/A převodníky a výstupy proto nejsou funkční.

6.2 Funkce analogových výstupů

Analogové výstupy patří k nejjednodušším I/O portům karty a lze je ovládat prostým zápisem do registru (bez vlivu na ostatní funkce karty).

Analogové výstupy lze přepnout do tří rozsahů (0÷5 V, ±5 V, 0÷10 V) pomocí propojek umístěných v horní části karty, stav propojek lze zjistit pomocí vyhrazeného registru DACRangeReg.

Analogové výstupy jsou kalibrovány interními konstantami uloženými ve flash paměti karty (karta obsahuje nezávislé konstanty pro každý rozsah), kalibraci lze provést k tomu účelu určeným uživatelským programem.

6.3 Registr DAC0Reg (WR)

Tento registr slouží k ovládní stavu analogového výstupu 0, zapsaná 16bitová hodnota je po zápisu vyššího byte přepočtena interními kalibračními konstantami a přenesena do 12bitového D/A převodníku. Rychlost přepočtu a přenosu dat umožňuje plnou rychlostí zapisovat 16bitová data.

Data jsou přenášena v přímém kódu, tzn.

- v případě unipolárních rozsahů zapsaná hodnota 0000_H (tzn. 0_D) nastaví výstupní napětí na 0 V a hodnota $FFFF_H$ (tzn. 65535_D) nastaví maximální kladné napětí;
- v případě bipolárních rozsahů zapsaná hodnota 8000_H (tzn. 32768_D) nastaví výstupní napětí na 0 V, hodnota $FFFF_H$ (tzn. 65535_D) nastaví maximální kladné napětí a hodnota 0000_H (tzn. 0_D) nastaví maximální záporné napětí.

6.4 Registr DAC1Reg (WR)

Tento registr slouží k ovládní stavu analogového výstupu 1, funkce je analogická registru DAC0Reg.

6.5 Registr DACRangeReg (RD)

Tento registr umožňuje detekovat pracovní rozsahy analogových výstupů zvolené propojkami na desce.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV				DAC1_Range		DAC0_Range	

DAC0_Range nastavený rozsah analogového výstupu 0
 00 rozsah 0÷5 V
 01 rozsah ±5 V
 10 rozsah 0÷10 V
 11 rezerva

DAC1_Range nastavený rozsah analogového výstupu 1
 00 rozsah 0÷5 V
 01 rozsah ±5 V
 10 rozsah 0÷10 V
 11 rezerva

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

7. Registry pro obsluhu čítačů kompatibilních s PCA-7428A

7.1 Úvod

V následujících odstavcích budou popsány registry související s 16bitovými čítači pro zpětnou kompatibilitu s PCA-7428A (označeny XCNT); není-li kompatibilita s předešlými verzemi karet vyžadována, je vhodné použít funkčně dokonalejší programovatelné 32bitové čítače popsané v 8. kapitole (označeny CNT).

Upozornění: *Nepřehlédněte registr XCNTEnReg popsaný v této kapitole sloužící k přepínání digitálního vstupního portu do režimu programovatelných čítačů, nebo režimu čítačů kompatibilních s PCA-7428A. Současně je přepínán i signál pro externí spouštění scanovací sekvence.*

Registry lze rozdělit na skupinu společnou všem čítačům

XCNTEnReg	slouží k přepínání digitálního vstupního portu do režimu programovatelných čítačů, nebo do režimu čítačů kompatibilních s PCA-7428A
XCNTCtrlReg	registr pro povolení/zakázání čítání, resp. konfiguraci hradlovacích signálů GATE
XCNT0SetReg	registr pro nastavení počáteční hodnoty čítače XCNT0 při startu scanování
XCNT1SetReg	registr pro nastavení počáteční hodnoty čítače XCNT1 při startu scanování

7.2 Funkce XCNT čítačů

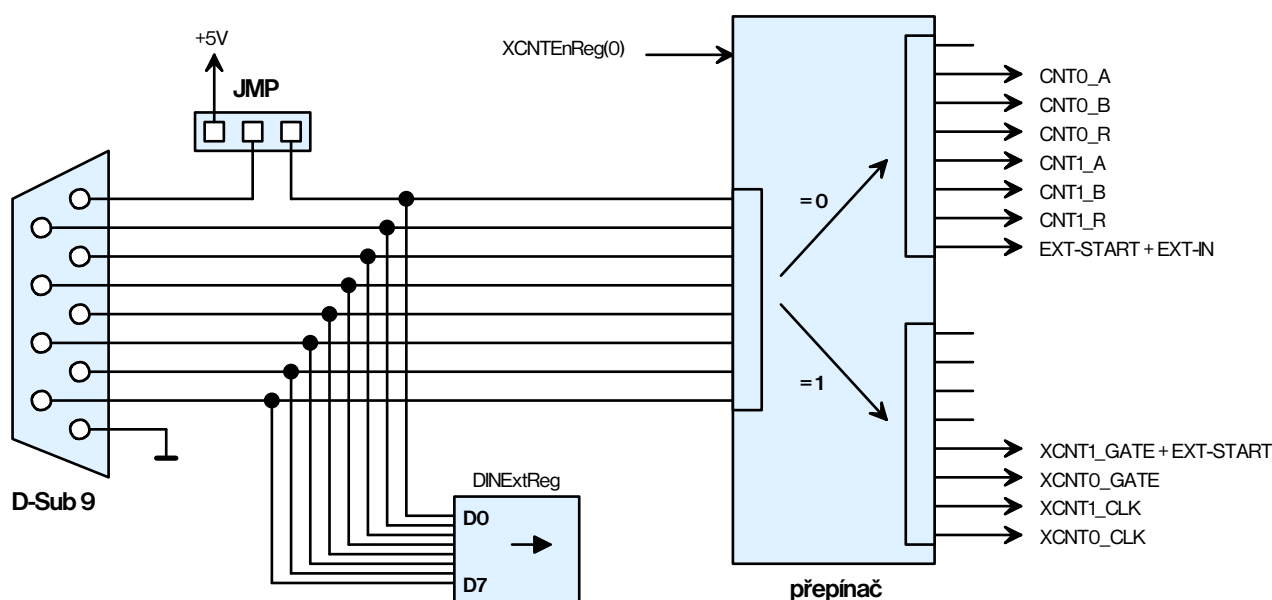
Čítače kompatibilní s PCA-7428A lze charakterizovat jako 16bitové jednosměrné (čítají nahoru) čítače s možností blokování čítání softwarově nebo externím signálem. Jejich aktuální hodnota je dostupná výhradně prostřednictvím scanovacích obvodů a FIFO paměti (tzn. nelze jejich hodnotu přímo číst). Programovat lze hodnotu při zahájení scanování a následně ji již nelze měnit.

Čítače jsou určeny pro aplikace zpětně kompatibilní s předešlou řadou karet PCA-7428A, ve všech ostatních případech je vhodné použít programovatelné 32bitové čítače popsané v 8. kapitole.

7.3 XCNTEnReg (WR)

Tento registr slouží k přepínání digitálního vstupního portu do režimu programovatelných čítačů (D0=0), nebo režimu čítačů kompatibilních s PCA-7428A (D0=1). Současně je přepínán i signál pro externí spouštění sekvence.

Významný je pouze bit D0, ostatní bity jsou rezervovány a z důvodu dopředné kompatibility doporučena hodnota 0.



7.4 XCNTCtrlReg (WR)

Tento registr slouží pro povolení/zakázání čítání, resp. konfiguraci hradlovacích signálů GATE.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV				XCNT1_MODE		XCNT0_MODE	

XCNT0_MODE	volba pracovního režimu čítače XCNT0
00	čítač je blokován (tzn. nezpracovává pulsy na signálu XCNT0_CLK)
01	čítač zpracovává signál XCNT0_CLK (aktivní sestupná hrana), signál XCNT0_GATE ignorován
10	čítač zpracovává signál XCNT0_CLK (aktivní sestupná hrana) za podmínky XCNT0_GATE=1
11	čítač zpracovává signál XCNT0_CLK (aktivní sestupná hrana) za podmínky XCNT0_GATE=0
XCNT1_MODE	volba pracovního režimu čítače XCNT1
00	čítač je blokován (tzn. nezpracovává pulsy na signálu XCNT1_CLK)
01	čítač zpracovává signál XCNT1_CLK (aktivní sestupná hrana), signál XCNT1_GATE ignorován
10	čítač zpracovává signál XCNT1_CLK (aktivní sestupná hrana) za podmínky XCNT1_GATE=1
11	čítač zpracovává signál XCNT1_CLK (aktivní sestupná hrana) za podmínky XCNT1_GATE=0
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

7.5 Registr XCNT0SetReg (WR)

Tento registr definuje počáteční hodnotu 16bitového čítače XCNT0. Při zahájení scanování je obsah registru přenesen do čítače XCNT0 a čítač následně zpracovává vstupní signály.

7.6 Registr XCNT1SetReg (WR)

Tento registr definuje počáteční hodnotu 16bitového čítače XCNT1, funkce je analogická registru XCNT0SetReg.

8. Registry pro obsluhu programovatelných čítačů

8.1 Úvod

V následujících odstavcích budou popsány registry související s programovatelnými čítači (dále jen čítači), viz přehled ve 4. kapitole.

Registry lze rozdělit na skupinu společnou oběma 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... nebo CNT1...)

CNT0SetReg	obsahuje hodnotu programovanou do čítače CNT0
CNT0RngReg	definuje rozsah čítání čítače CNT0 (0+1 až 0+4.294.967.295)
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

Upozornění: *Nepřehledněte registr XCNTEnReg popsáný v 7. kapitole sloužící k přepínání digitálního vstupního portu do režimu programovatelných čítačů, nebo režimu čítačů kompatibilních s PCA-7428A.*

8.2 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 PCA-7428C 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 scanovacími obvody

Tento režim je určen zejména 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 analogové vstupy, čítače, digitální porty a diagnostické informace.

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

8.3 Registrová struktura

Programovatelné čítače jsou obsluhovány registry analogickými (a také stejnojmennými) čítačům karet PCT-7303C/E, registrová struktura (resp. adresové mapování registrů) je však odlišná.

Čítače karet PCA-7428C mají předřazeny identické enkodéry jako karty řady PCT-7303, jelikož rozsah čítání byl zvýšen ze 24 bitů na 32 bitů došlo k rozšíření odpovídajících registrů.

Při programování je potřeba vzít v úvahu, že ...

- rozsah čítání byl zvýšen ze 24 bitů na 32 bitů a došlo proto k rozšíření odpovídajících datových registrů;
- nová registrová struktura předpokládá až 8 čítačů, řídicí registry CNTEnReg a CNTCtrlReg jsou proto 16bitové;
- z důvodu minimalizace nároků na adresový prostor sdílí méně užívané registry stejné adresy a jejich přepínání je řešeno pomocným registrem CNTSelReg (základní datové registry čítačů CNTxSetReg a CNTxStrReg jsou mapovány přímo); přepínání registrů je společné všem čítačům a je proto nutné ošetřit případné kolize zejména v případech, kdy k registrům přistupuje více souběžných procesů.

Upozornění: Korektní přístup k vícebytovým registrům byl popsán ve 4. kapitole.

8.4 Registr CNTSelReg (WR)

Tento registr slouží k volbě registrů sdílejících společnou adresu v registrové struktuře.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV				REG_NO			

REG_NO volba registrů sdílejících společnou adresu
 0000 na odpovídající adresy jsou mapovány registry CNTxCWReg (WR) a CNTxStatReg (RD)
 0001 na odpovídající adresy jsou mapovány registry CNTxRngReg (WR) a CNTxXStrReg (RD)
 rezerva

RSRV rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

8.5 Registr CNTEnReg (WR)

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

D15 ... D10	D9	D8	D7 .. D2	D1	D0
RSRV	EN_AB1	EN_AB0	RSRV	EN_R1	EN_R0

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_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

RSRV rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

8.6 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.

D15 ... D10	D9	D8	D7 .. D2	D1	D0
RSRV	STR_1	STR_0	RSRV	SET_1	SET_0

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)
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)
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

8.7 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 a CNT1XStrReg 0 obvody pro zachycení stavu čítačů neaktivní 1 obvody pro zachycení stavu čítačů aktivovány
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

8.8 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 a CNT1XStrReg 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)

8.9 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 a CNT1XStrReg 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)

8.10 Registr CNT0CWReg (WR, CNTSelReg=0000)

Tento registr slouží ke konfiguraci čítače CNT0, pro čítač CNT1 je implementován analogický registr.

D31 ... 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.

8.11 Registr CNT0StatReg (RD, CNTSelReg=0000)

Tento registr slouží ke čtení stavových příznaků čítače CNT0, pro čítač CNT1 je implementován analogický registr.

D31 ... 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)

8.12 Registr CNT0SetReg (WR)

Tento registr plní funkci 32bitového vyrovnávacího datového registru pro zápis dat do čítače CNT0, data jsou do čítače CNT0 přenesena pomocí registru CNTCtrlReg. Pro čítač CNT1 je implementován analogický registr.

Je-li do čítače CNT0 zapsána hodnota mimo rozsah 0÷CNT0RngReg, pracuje čítač v plném 32bitovém rozsahu až do okamžiku, kdy hodnota čítače vstoupí do zvoleného rozsahu 0÷CNT0RngReg.

8.13 Registr CNT0RngReg (WR, CNTSelReg=0001)

Tento 32bitový registr je určen pro nastavení rozsahu čtení čítače CNT0, platné hodnoty jsou 1 až 4.294.967.295 (čítač CNT0 pak pracuje v rozsahu 0÷CNT0RngReg). Pro čítač CNT1 je implementován analogický registr.

8.14 Registr CNT0StrReg (RD)

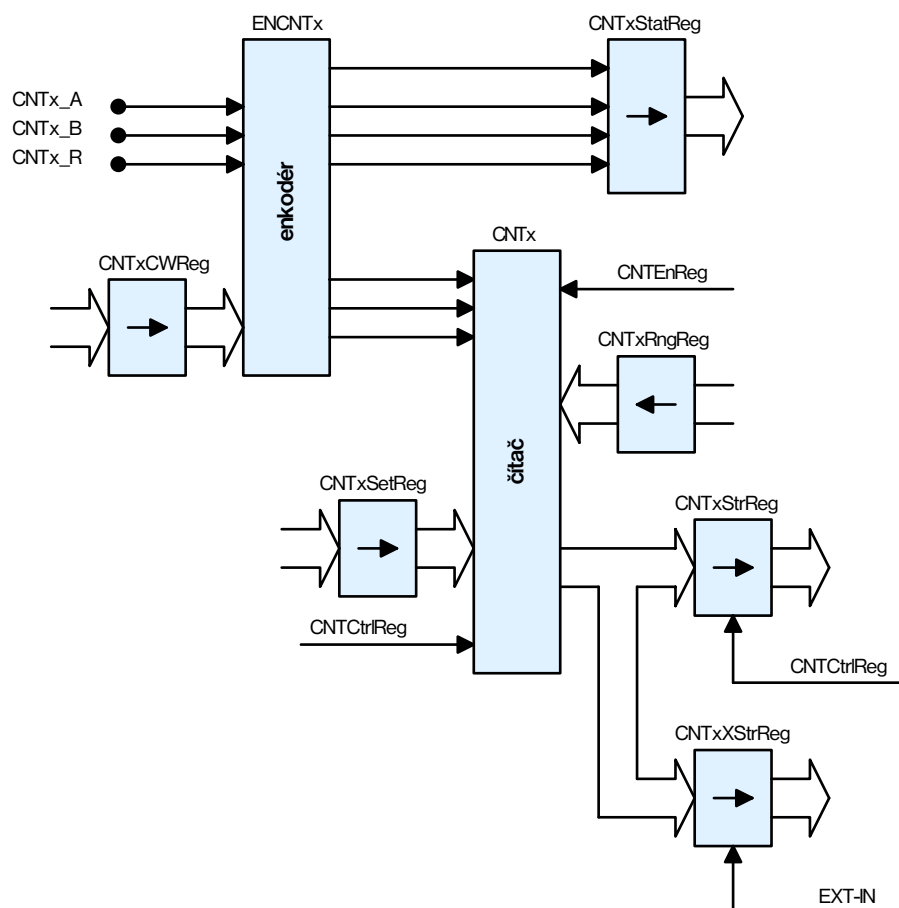
Tento registr plní funkci 32bitového vyrovnávacího datového registru pro softwarové čtení dat z čítače CNT0, data jsou z čítače CNT0 přenesena pomocí registru CNTCtrlReg. Pro čítač CNT1 je implementován analogický registr.

8.15 Registr CNT0XStrReg (RD, CNTSelReg=0001)

Tento registr plní funkci 32bitového vyrovnávacího datového registru pro zachycení aktuální hodnoty čítače CNT0 externím signálem (viz popis registrů CNTXSTREnReg, CNTXSTRStatusReg a CNTXSTRClrReg). Pro čítač CNT1 je implementován analogický registr.

8.16 Schéma registrové struktury čítače

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



9. Registry pro obsluhu scanovacích obvodů a FIFO paměti

9.1 Úvod

V následujících odstavcích budou popsány registry určené pro obsluhu scanovacích obvodů a FIFO paměti, viz přehled ve 4. kapitole.

Registry lze rozdělit na skupinu určenou pro základní řízení scanovacích obvodů,

CWReg	základní řídicí registr scanovacích obvodů
StatusReg	základní stavový registr scanovacích obvodů

skupinu určenou pro softwarové spouštění scanovací sekvence a přenos dat,

SWTrigReg	registr pro softwarové spuštění scanovací sekvence
SWTrigStatusReg	stavový registr scanovacích obvodů
SWFIFODataReg	datový zásobník typu FIFO výhradně pro data softwarově spouštěné scanovací sekvence

skupinu související se čtením FIFO paměti určené výhradně pro data scanovací sekvence spouštěné časovačem nebo externím signálem EXT-IN

FIFONoSmplStrbReg	slouží k zachycení aktuálního zaplnění FIFO paměti (tzn. počtu vzorků uložených ve FIFO)
FIFONoSmplReg	zachycená hodnota zaplnění FIFO paměti (tzn. počet vzorků uložených ve FIFO)
FIFODataReg	data čtená z FIFO paměti

a skupinu určené pro definici scanovací sekvence

ScanAdrReg	registr pro volbu scanovacího parametru
ScanDataReg	registr pro zápis/čtení scanovacího parametru

9.2 Funkce scanovacích obvodů

Scanovacími obvody jsou nazývány části karty určené pro

- definici scanovací sekvence (tedy sestavy vstupů, které je potřeba měřit, resp. zaznamenávat, ideálně v nekratším možném časovém rozestupu)
- řízení A/D převodníku, programovatelného zesilovače a multiplexeru
- obvody pro generování nebo zpracování startovací události (časovač pro periodické spouštění, zpracování signálů externího digitálního vstupu apod.)
- zápis dat do dvou FIFO pamětí
- generování stavových a chybových příznaků

9.3 Princip a funkce FIFO paměti

FIFO má strukturu fronty dat "první dovnitř - první ven". Můžeme si ji tedy představit jako trubku, do které jsou shora vhazovány míčky (= zapisována naměřená data) a na dolním konci jsou ve vhodný čas odebírány (= data jsou čtena); trubku lze vyprázdnit buď odebráním všech míčků (= vyčtením celého obsahu) nebo zastavit vhazování a trubku vysypat (= reset obsahu při ukončení záznamu).

Na straně zápisu jsou FIFO paměti řízeny scanovacími obvody, na straně čtení pak registry zpřístupňujícími zapsaná data.

Funkce paměti označené FIFO s kapacitou 32 kB

FIFO je využito v režimech spouštění scanovací sekvence časovačem a spouštění scanovací sekvence externím digitálním signálem EXT-ST. Obsah je resetován výhradně při zastaveném měření (tzn. CWReg=0), naměřená data se tedy doplňují nezávisle na průběh čtení a obslužný software musí vyhodnocovat stav zaplnění FIFO (softwarovým pollingem nebo s podporou přerušování) a data z FIFO paměti vyčítat. Nebudou-li data z FIFO čtena dostatečně rychle, dojde k chybovému zastavení měření při pokusu o zápis do FIFO obsahujícího 32768 B dat; data již uložená do FIFO však lze načíst.

Funkce paměti označené SWFIFO s kapacitou 512 B

SWFIFO je využito v režimu softwarového spouštění scanovací sekvence, resp. při softwarovém požadavku o přenos naměřených dat v ostatních režimech. Obsah je resetován při každém spuštění scanovací sekvence (resp. při požadavku o přenos naměřených dat) ovládaném registrem SWTrigReg a rovněž při zastaveném měření (tzn. CWReg=0); po ukončení zápisu dat obsahuje SWFIFO vždy data jediné sekvence a nemůže tak dojít k jeho přeplnění. Obslužný software musí vyhodnocovat obsah registru SWTrigStatusReg (softwarovým pollingem nebo s podporou přerušování) a po dokončení měření (resp. přenosu dat do SWFIFO) načíst data jedné sekvence.

9.4 Registr CWReg (WR)

Tento registr definuje režim scanovacích obvodů a slouží k zahájení scanování, nulování obsahu obou FIFO paměti a nulování příznaků ve stavovém registru StatusReg.

Při programování je potřeba vzít v úvahu, že nenulovou hodnotu registru lze nastavit výhradně z výchozího nulového stavu (tzn. nelze měnit různé metody scanování, aniž by bylo mezi nimi měření zastaveno).

D7	D6	D5	D4	D3	D2	D1	D0
RSRV				SCAN_MODE			

SCAN_MODE definuje režim scanovacích obvodů (viz popis v samostatné kapitole)

- 0000 scanování zastaveno, příznaky ve StatusReg nulovány
- 0001 softwarové spouštění scanovací sekvence s přenosem do SWFIFO
- 0010 spouštění scanovací sekvence časovačem (+ možnost přenosu do SWFIFO)
- 0011 spouštění scanovací sekvence externím digitálním signálem (+ možnost přenosu do SWFIFO)
- 0101 kontinuální spouštění scanovací sekvence se softwarovým přenosem do SWFIFO
- rezerva

RSRV rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

9.5 StatusReg (RD)

Tento registr poskytuje informaci o stavu scanovacích obvodů a FIFO paměti, resp. probíhajícím záznamu do FIFO.

K nulování příznaků dojde nastavením režimu "scanování zastaveno" registrem CWReg.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV				ERROR	RSRV	FAULT	RSRV

FAULT signalizuje startovací puls po dobu probíhající sekvence (např. při spouštění externím signálem); při nastavení tohoto příznaku měření pokračuje, příznak má jen informativní význam

- 0 od startu scanování nebyl detekován startovací puls po dobu probíhající sekvence
- 1 od startu scanování detekován minimálně jeden startovací puls po dobu probíhající sekvence

ERROR úroveň 1 signalizuje, že měření bylo ukončeno z důvodu fatální chyby (např. došlo k přetečení FIFO určeného pro spouštění scanovací sekvence časovačem nebo externím signálem); v takovém případě dojde k ukončení zápisů dat do FIFO (již zapsaný obsah je však dostupný) a program musí nastavit CWReg do režimu "scanování zastaveno"

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

9.6 SWTrigReg (WR)

Tento registr slouží ke spuštění scanovací sekvence v režimu softwarového spouštění (viz CWReg).

D7	D6	D5	D4	D3	D2	D1	D0
RSRV							SW_ST

SW_ST slouží ke spuštění scanovací sekvence

- 0 zápis do registru ignorován
- 1 inicializuje probíhající scanovací sekvenci, resp. probíhající přenos do SWFIFO

RSRV rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

9.7 SWTrigStatusReg (RD)

Tento registr poskytuje informaci o průběhu scanovací sekvence v režimu softwarového spouštění.

D7	D6	D5	D4	D3	D2	D1	D0
RSRV							SW_RUN

SW_RUN úroveň 1 signalizuje probíhající scanovací sekvenci v režimu softwarového spouštění, resp. probíhající přenos do SWFIFO

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

9.8 SWFIFODataReg (RD)

Tento registr slouží pro přístup k datům uložených ve SWFIFO paměti určené pro režim softwarové spouštění.

Pro dokončení sekvence (viz SWTrigStatusReg) SWFIFO obsahuje data jedné sekvence.

9.9 FIFONoSmpIstrbReg (WR)

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

9.10 FIFONoSmpIReg (RD)

Tento registr poskytuje informaci o stavu zaplnění FIFO paměti v rozsahu 0 až 32768; obsah tohoto registru je aktualizován zápisem do registru FIFONoSmpIstrbReg.

9.11 FIFODataReg (RD)

Tento registr slouží pro přístup k datům uložených ve FIFO paměti určené pro režimy spouštění časovačem nebo externím signálem.

Pro zjištění stavu zaplnění FIFO paměti je určen registr FIFONoSmpIReg.

9.12 ScanAdrReg (WR)

Tento registr slouží jako selektor scanovacího parametru dostupného prostřednictvím ScanDataReg.

Registr má implementovanou funkci autoinkrementace přístupem do nejvyššího byte ScanDataReg; pokud se tedy zapisuje nebo čte souvislé pole scanovacích registrů, postačuje nastavit jen počáteční hodnotu ScanAdrReg.

Podrobně viz popis scanovacích parametrů dále.

9.13 ScanDataReg (WR/RD)

Tento registr slouží k zápisu zvoleného scanovacího parametru (viz ScanAdrReg), resp. zpětného čtení zapsaného obsahu za předpokladu režimu "scanování zastaveno" registrem CWReg.

Podrobně viz popis scanovacích parametrů dále.

9.14 Struktura scanovacích parametrů

Zatímco v předešlých odstavcích byla stručně popsána funkce scanovacích obvodů včetně FIFO paměti a příslušné registry, následující odstavce budou věnovány podrobnému popisu scanovacích parametrů a současně i podrobnějšímu popisu činnosti scanovacích a s nimi souvisejících obvodů.

Všechny scanovací parametry jsou na kartě uloženy v RAM bloku s kapacitou 256*32bitů. Pro adresování je použitý 8bitový registr ScanAdrReg, pro vlastní přístup k datům pak 32bitový registr ScanDataReg. Za připomenutí stojí funkce autoinkrementace ScanAdrReg, tzn. hodnota registru ScanAdrReg je automaticky inkrementována každým zápisem nebo čtením nejvyššího byte ScanDataReg.

Scanovací parametry zůstávají v paměti zachovány i po ukončení měření.

Jak je patrné z tabulky níže, RAM blok obsahuje kromě scanovacích parametrů navíc dva pomocné registry.

ScanAdrReg	ScanDataReg
0	scanovací parametr 0, definuje vlastnosti 1. kanálu (význam viz samostatná tabulka)
1	scanovací parametr 1, definuje vlastnosti 2. kanálu (význam viz samostatná tabulka)
...
126	scanovací parametr 126, definuje vlastnosti 127. kanálu (význam viz samostatná tabulka)
127	scanovací parametr 127, definuje vlastnosti 128. kanálu (význam viz samostatná tabulka)
128÷191	rezerva (RAM nemusí být implementována)
192	hodnota v rozsahu 0÷127 definuje poslední aktivní scanovací parametr (scanovací parametry musejí být konfigurovány jako kontinuální pole parametrem 0 počínaje)
193	hodnota v rozsahu 250÷16.777.215 definuje vzorkovací frekvenci spouštění scanovací sekvence v režimu spouštění časovačem (řešeno 24bitovou děličkou z oscilátoru s frekvencí 25 MHz)
194÷255	rezerva (RAM nemusí být implementována)

Zatímco předešlá tabulka zobrazovala mapování scanovacích registrů v rámci RAM bloku, v následující tabulce je popsán formát scanovacích parametrů ukládaných na pozici 0+127.

D31 ... D24	D23 ... D16	D15 ... D8	D7 ... D0	délka dat
10÷255 (definuje dobu měření kanálu v rozsahu 10÷255 μ s)	00 _H zesílení 1x 01 _H zesílení 2x 02 _H zesílení 4x 03 _H zesílení 8x 04 _H zesílení 16x 05 _H zesílení 32x 8x _H zesílení 1x÷32x s průměrováním osmi A/D měření	00 _H = AIN	00 _H AIN0 01 _H AIN1 1E _H AIN30 1F _H AIN31	2B 2B ... 2B 2B
0	0	01 _H = CNT/XCNT	00 _H CNT0 01 _H CNT1 F0 _H XCNT0 F1 _H XCNT1	4B 4B 2B 2B
0	0	02 _H = DIN	00 _H DINExt (MSB), DIN (LSB)	2B
0	0	03 _H = ostatní vstupy	00 _H časová značka sekvence	4B
0	0	10 _H = zpětné čtení	00 _H 00 _H (MSB), DOUReg (LSB) 80 _H DAC0Reg 81 _H DAC1Reg	2B 2B 2B

Jak je patrné z tabulky, scanovací parametr v rozsahu bitů D15...D8 slouží pro rozlišení typu kanálu, zatímco bity D7...D0 definují číslo vstupů/čítače apod. Z jiného pohledu (budeme-li uvažovat oba byte společně) lze říci, že kanály 0÷255 jsou vyhrazeny pro analogové vstupy, kanály 256÷511 pro čítače, atd.

Scanovací parametr v rozsahu bitů D31...D16 slouží pro předávání specifických informací souvisejících s typem kanálu; pro analogové vstupy jde o zesílení a dobu měření, pro ostatní typy je tato část scanovacího parametru rezervována.

Upozornění: Veškeré hodnoty neuvedené v tabulce jsou rezervovány a jejich zápisem může dojít k nedefinovanému chování scanovacích obvodů karty.

Délka dat zapisovaných do obou FIFO závisí na typu kanálu (viz poslední sloupec) a data jsou do FIFO přenášena

- v pořadí definovaném scanovacími parametry
- nejnižším bytem počínaje
- číselné hodnoty analogových vstupů v přímém kódu
- zpětné čtené hodnoty ve formátu zapsaných dat

Analogové vstupy

Analogové vstupy pracují s volitelnými rozsahy od ± 10 V (zesílení 1x) až do $\pm 0,3125$ V (zesílení 32x), data jsou přenášena v 16bitovém přímém kódu, tzn.

- hodnota 8000_H (tzn. 32768_D) představuje vstupní napětí 0 V
- hodnota FFFF_H (tzn. 65535_D) představuje maximální kladné vstupní napětí (např. +10 V pro rozsah ± 10 V)
- hodnota 0000_H (tzn. 0_D) představuje maximální záporné vstupní napětí (např. -10 V pro rozsah ± 10 V)

Scanovací parametr analogových vstupů má nejvíce možností nastavení; kromě multiplexeru (tzn. volby vstupu) a zesílení (tzn. vstupního rozsahu) umožňuje ještě nastavit dobu měření.

Pro běžné aplikace (tzn. zdroj signálu s impedancí nižší než 1 kOhm) vyhovují následující minimální doby měření:

- 10 μ s pro základní rozsahy se zesílením 1x, 2x, 4x a 8x
- 13 μ s pro základní rozsah se zesílením zesílení 16x
- 18 μ s pro základní rozsah se zesílením zesílení 32x
- + 2 μ s navíc pro kanál, u něhož dochází ke změně osmice vstupů, tzn. změně hodnoty MUX na bitu D3 nebo D4 (slouží k ustálení vnějšího multiplexeru rozšiřujícího základních 8 vstupů karty na 32)
- + 20 μ s navíc pro kanál s nastaveným rozsahem "8x_H", tzn. s průměrováním osmi výsledků A/D převodníku

Pokud je vstupní signál buzen ze zdroje s vyšší impedancí, je vhodné dobu měření prodloužit; nedostatečná doba pro ustálení se projevuje výrazně vyšším šumem a přeslechem mezi kanály.

Čítače a digitální vstupy

Scanovací parametr těchto kanálů nenabízí žádná specifická nastavení.

Doba pořízení a zápisu dat kanálu do FIFO paměti je kratší než 1 μ s.

Ostatní vstupy

Karta umožňuje ukládat časovou značku sekvence vytvářenou 32bitovou děličkou pracující se vstupní frekvencí 1 MHz spuštěnou od nuly v okamžik startu měření. Scanovací parametr nenabízí žádná specifická nastavení.

Doba pořízení a zápisu dat kanálu do FIFO paměti je kratší než 1 μ s.

Poznámka: Časová značka sekvence má význam především v případě spouštění externím digitálním signálem.

Zpětné čtení

Tato skupina kanálů umožňuje zpětné čtení hodnot výstupních kanálů ovládaných programem. Scanovací parametr nenabízí žádná specifická nastavení.

Doba pořízení a zápisu dat kanálu do FIFO paměti je kratší než 1 μ s.

Poznámka: Zpětné čtení má význam v aplikacích, kdy program generuje řídicí signály do testovaného systému a současně měří odezvy; synchronní záznam eliminuje zpoždění vzniklé přenosem dat FIFO paměti.

9.15 Programová obsluha scanovacích obvodů

Výchozím stavem pro popis obsluhy scanovacích obvodů je režim "scanování zastaveno", tzn. nulová hodnota CWReg. Všechny příznaky související se scanovacími obvody jsou vynulovány, obsah obou FIFO paměti vyprázdněn.

Prvním krokem obsluhy je konfigurace scanovacích parametrů, tzn. nastavení obsahu souvislého bloku scanovacích registrů od nultého kanálu výše. Následně je nutné do registru mapovaného ScanAdrReg=192 zapsat číslo posledního využitého scanovacího registru.

V případě spouštění sekvence časovačem je potřeba nastavit vzorkovací frekvenci, resp. periodu vzorkování pomocí registru mapovaného ScanAdrReg=193. Minimální povolenou hodnotu limituje současně několik pravidel:

- musí být větší nebo rovna 250 (odpovídá periodě vzorkování $250 \cdot 0,04 = 10 \mu$ s)
- musí být větší nebo rovna době sekvence (tzn. součet doby měření analogových vstupů a 1 μ s pro každý další kanál)
- datový tok nesmí překračovat 200 kB/s (např. pro záznam jediného kanálu CNT0 s délkou dat 4 B by perioda vzorkování neměla být nižší 20 μ s); toto omezení však není zcela striktní jako dvě předešlá

Po dokončení konfigurace lze zahájit měření (tzn. scanování) nastavením registru CWReg do požadovaného režimu (viz popis v samostatné kapitole).

Od okamžiku zahájení měření scanovací obvody čekají na spouštěcí událost (tzn. softwarový pokyn generovaný zápisem do registru SWTrigReg v režimu 0001, puls generovaný časovačem v režimu 0010 nebo puls na digitálním vstupu EXT-IN v režimu 0011), nebo sekvence probíhají zcela kontinuálně (režim 0101).

Po detekování události scanovací obvody nastaví multiplexer a vstupní zesilovač podle parametrů scanovacího registru nultého kanálu, vyčkají na ustálení, provedou měření, v případě analogového vstupu přepočtou data A/D převodníku kalibračními konstantami a zapíší výsledek do jedné z FIFO paměti v závislosti na typu spouštění.

Je-li kanál konfigurovaný pro jiný typ než analogový vstup, zůstanou multiplexer a zesilovač nastaveny na předcházející kanál a namísto měření A/D převodníkem jsou do jedné z FIFO paměti zapsána data zvoleného kanálu.

Popsaným mechanismem jsou postupně interpretovány všechny nastavené scanovací parametry (tzn. od nultého po číslo nastavené registrem mapovaným ScanAdrReg=128) a poté je sekvence ukončena; ve FIFO paměti jsou zapsána data všech kanálů sekvence.

Spouštěcí událost detekovaná v průběhu probíhající sekvence (např. příliš nízká hodnota periody časovače nebo externí puls v průběhu sekvence) je ignorována a tento stav lze vyhodnotit pomocí registru StatusReg.

V případě softwarového spouštění (resp. požadavku o přenos naměřených dat) iniciovaném zápisem do registru SWTrigReg je je nejprve vymazán obsah SWFIFO a následně jsou zapsána data jedné sekvence. Probíhající sekvence (resp. probíhající přenos dat do SWFIFO) je signalizována v registru SWTrigStatusReg (lze jej využít pro polling), data lze načíst po jejím dokončení.

V případě spouštění časovačem nebo externím signálem je potřeba vyhodnocovat zaplnění FIFO paměti daty buď pollingem nebo pomocí přerušení systému. Program musí průběžně vyčítat data z FIFO paměti; v případě přetečení (tzn. pokus o zápis při úplném zaplnění daty) dojde k ukončení scanování (je signalizováno pomocí registru StatusReg), data doposud zapsaná do FIFO paměti zůstávají k dispozici. Nezávisle na záznamu do FIFO paměti lze pomocí registru SWTrigReg iniciovat přenos aktuálně naměřených dat (tzn. poslední dokončené sekvence) do SWFIFO.

Pro vyčítání dat lze kromě pollingu využít obvodů přerušení (jsou popsány v samostatné kapitole); přerušení může být vyvoláno dokončením sekvence (resp. přenosu dat do SWFIFO) nebo definovaným zaplněním FIFO paměti.

K ukončení měření (tzn. scanování) postačuje nastavit CWReg=0. Po ukončení měření je vymazán obsah FIFO paměti, obsah všech scanovacích registrů však zůstává zachován (tzn. měření lze znovu spustit).

9.16 Porovnání pracovních režimů

Karty umožňují využívat scanovací obvody v několika režimech, viz popis registru CWReg.

V režimu **0001** start scanovací sekvence čeká softwarový pokyn; zápisem do registru SWTrigReg dojde ke startu jedné sekvence a data jsou zapisována do SWFIFO. Po dokončení sekvence (její průběh je signalizován příznakem v registru SWTrigStatusReg) SWFIFO obsahuje data všech naměřených kanálů a scanovací obvody čekají na další pokyn.

Režim **0101** je alternativou k předešlému režimu; scanovací sekvence jsou spouštěny automaticky nejvyšší možnou frekvencí (tzn. měření probíhá prakticky kontinuálně) a naměřená data jsou ukládána ve zvláštní vyrovnávací paměti. Při požadavku u "softwarový start sekvence" zápisem do SWTrigReg dojde k rychlému přenosu dat poslední naměřené sekvence z vyrovnávací paměti do SWFIFO. Výhodou režimu je absence čekání na provedení sekvence, drobnou nevýhodou (spíše však teoretickou) pak vyšší jitter prováděného měření.

Přenos dat do SWFIFO je velmi rychlý, data lze číst prakticky ihned po zápisu do SWTrigReg (příznak probíhajícího přenosu dat v registru SWTrigStatusReg "problikne" jen na velmi krátkou dobu a program jej zpravidla ani nestihne zaregistrovat); výjimkou je však případ, kdy k zápisu do SWTrigReg dojde bezprostředně po nastavení režimu 0101 registrem CWReg, v takovém případě je potřeba počkat na provedení první sekvence a příznak v registru SWTrigStatusReg je nastaven po celou dobu provádění sekvence. Z uvedeného důvodu je tedy vhodné zachovat stejný algoritmus jako u režimu 0001, tedy start měření pomocí SWTrigReg, vyčkání na dokončení sekvence (resp. přenosu dat do SWFIFO) a vyčtení dat ze SWFIFO.

Režim **0010** umožňuje spouštění scanovací sekvence zvolenou frekvencí v rozsahu od 1,5 Hz do 100 kHz generovanou časovačem. Každým přetečením tohoto časovače (tzn. poprvé po uplynutí doby odpovídající periodě vzorkování určené jako převrácená hodnota vzorkovací frekvence) dojde ke startu sekvence a naměřená data jsou zapisována do paměti FIFO, aniž by byl modifikován obsah zapsaný předešlými sekvencemi. Čtení FIFO probíhá nezávisle na zapisování, obsluhý software musí vyhodnocovat stav zaplnění FIFO (softwarovým pollingem nebo s podporou přerušení) a data z FIFO paměti vyčítat.

Souběžně s probíhajícím měřením spouštěným časovačem lze vyžádat přenos aktuálně naměřených dat do SWFIFO; funkce je zcela analogická režimu 0101 s tím, že do zvláštní vyrovnávací paměti jsou ukládána data měřená nikoliv kontinuálně, nýbrž spouštěná časovačem. Z uvedeného plyne, že první pokyn pro přenos dat do SWFIFO bude vykonán až po naměření první sekvence časovačem, tzn. za dobu odpovídající periodě měření. Právě tak platí, že při nízké vzorkovací frekvenci nastavené časovačem v kombinaci s vysokou frekvencí požadavků o přenos dat do SWFIFO budou opakovaně přenášena tatáž data.

Funkce softwarového požadavku o přenos dat do SWFIFO je vhodná pro aplikace, kdy je prováděno měření s vyšší vzorkovací frekvencí (např. jeden proces načítá data z FIFO a ukládá je na disk) a souběžně je potřeba řešit orientační náhled naměřených dat (např. druhý proces tedy občasné načte aktuální data a zobrazí je obsluze).

Režim **0011** umožňuje spouštění scanovací sekvence externím digitálním signálem a platí pro něj vše popsané v souvislosti s režimem 0010 s jedinou výjimkou - ke spouštění nedochází s pevnou periodou, nýbrž s časovým rozestupem definovaným externím signálem. Signál tedy může generovat puls i v době, kdy probíhá sekvence (puls je ignorován; viz popis StatusReg), nebo naopak nemusí dojít ke spuštění sekvence po dlouhou dobu, případně vůbec.

I v tomto režimu je zpracováván pokyn pro přenos dat do SWFIFO, využitelnost této funkce je však omezena na aplikace, u nichž se lze spolehnout na přítomnost spouštěcích pulsů.

10. Registry pro obsluhu přerušení

10.1 Úvod

V následujících odstavcích budou popsány registry související s obvody pro přerušení systému, viz přehled ve 4. 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
FIFOIRQReg	nastavení prahu pro vyvolání přerušení
TimerReg	generátor časových značek pro periodické vyvolávání přerušení

a skupinu registrů souvisejících s funkcemi popsanými v předešlých kapitolách

CNTXSTREnReg	viz kapitola s popisem obsluhy programovatelných čítačů
CNTXSTRStatusReg	viz kapitola s popisem obsluhy programovatelných čítačů
CNTXSTRClrReg	viz kapitola s popisem obsluhy programovatelných čítačů

10.2 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:

Dokončení scanovací sekvence

Umožňuje vyvolávat přerušení po dokončení scanovací sekvence (funkční pro všechny typy spouštění).

Generátor časových značek

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

Zachycení hodnot programovatelných čítačů

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

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 zachycení hodnot čítačů apod.

10.3 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)

Upozornění: Nepřehledněte popis obsluhy obvodů přerušení uvedený v kapitole 10.9.

10.4 Registr IRQCfgReg (WR)

Tento registr slouží k povolení základních zdrojů přerušení.

D7	D6	D5	D4	D3	D2	D1	D0
FIFO	XSTR	RSRV	TIM	RSRV			EOS

EOS	povoluje vyvolání přerušení odvozené od konce scanovací sekvence (má smysl zejména v režimu softwarového spouštění 0001, od firmware FPGA verze 1.0 je podporováno i v režimu 0101) 0 záchytný registr navázaný na signál ukončující scanovací sekvenci je blokován 1 funkce záchytného registru je aktivována
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
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), resp. nadlimitním zaplněním FIFO v okamžik pokusu o vynulování příznaku
RSRV	rezerva (z důvodu dopředné kompatibility doporučena hodnota 0)

10.5 Registr IRQStatusReg (RD)

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

D7	D6	D5	D4	D3	D2	D1	D0
FIFO	XSTR	RSRV	TIM	RSRV			EOS

EOS	stav záchytného registru navázaného na signál ukončující scanovací sekvenci 0 registr není nastaven, tzn. od posledního nulování nedošlo k ukončení sekvence, resp. k dokončení přenos naměřených dat do SWFIFO 1 registr je nastaven, tzn. od posledního nulování došlo k ukončení sekvence, resp. přenosu dat
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
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, resp. obsah registru CNTXSTRStatusReg se nezměnil z nulového hodnoty na nenulovou 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 nad úroveň definovanou FIFOIRQReg 1 registr je nastaven, tzn. od posledního nulování došlo k zaplnění FIFO paměti
RSRV	rezerva (z důvodu dopředné kompatibility je doporučeno hodnoty ignorovat)

10.6 Registr IRQClrReg (WR)

Tento registr slouží nulování záchytných registrů povolovaných registrem IRQCfgReg.

D7	D6	D5	D4	D3	D2	D1	D0
FIFO	XSTR	RSRV	TIM	RSRV			EOS

EOS	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)
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)
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)

10.7 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).

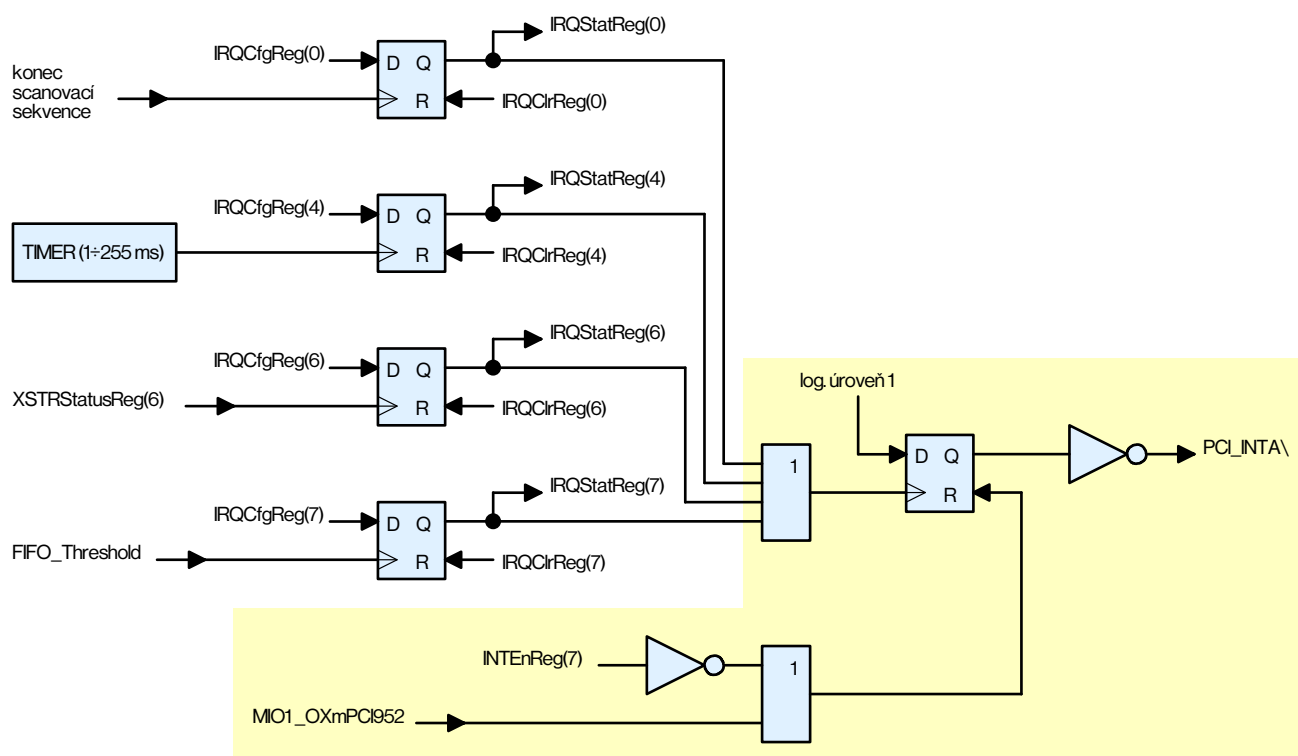
Pro příklad, zápisem hodnoty 100 bude první přerušení vyvoláno 100 ms po zápisu do registru a následně vždy po dalších 100 ms. Čtením budou získávány hodnoty 0, 1, ..., 98, 99, 0, 1, ..., přerušení je vyvoláno v okamžik přechodu z 99 na 0.

10.8 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 $0 \div 14$ a představuje práh 2^N (tzn. v rozsahu $1 \div 16384$).

10.9 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.

11. Pomocné diagnostické a servisní registry

11.1 Úvod

V následujících odstavcích budou popsány pomocné diagnostické a inicializační registry, viz přehled ve 4. kapitole.

Přehled diagnostických 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
ResetReg	registr pro nastavení registrů FPGA do výchozího stavu

Přehled registrů rozhraní inicializačních dat:

CalibAdrReg	16bitová adresa do bloku kalibračních a inicializačních konstant
CalibDataReg	8bitová data bloku kalibračních a inicializačních konstant
CalibCtrlReg	pomocný registr povolující zápis do bloku kalibračních a inicializačních konstant
CalibStatReg	pomocný registr signalizující připravenost bloku kalibračních a inicializačních konstant

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, zjištění čísla CardIDReg nebo resetování registrů karty.

Inicializační registry slouží zpravidla pro uchování kalibračních a inicializačních konstant.

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

11.2 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.

11.3 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.

11.4 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.

11.5 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 PCA-7428C je uvedena v 1. kapitole.

11.6 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 PCA-7428C je uvedena v 1. kapitole.

11.7 Registr ResetReg (WR)

Zápisem hodnoty AA_H do tohoto registru dojde nastavení všech registrů FPGA (s výjimkou scanovacích registrů) včetně obnovení kalibračních konstant do výchozího stavu. Zápis generuje krátký resetovací puls, následný zápis jiné hodnoty není vyžadován.

11.8 Rozhraní bloku inicializačních dat

Blok inicializačních dat pracuje v modelu nepřímé adresované paměti (je tedy mapována prostřednictvím rozhraní popsaného v následujících odstavcích), jejíž obsah je inicializován po zapnutí/restartu obsahem flash paměti karty.

Aktuální verze firmware využívá následující rozdělení bloku:

00FF _H ÷ 0000 _H	obsahuje kalibrační a inicializační konstanty použité výpočetní jednotkou; blok je dostupný pro čtení i zápis, změna se tedy okamžitě projeví ve výsledných datech; konstanty jsou po každém zapnutí/restartu inicializovány hodnotami z flash paměti karty
FEFF _H ÷ 0100 _H	rezervováno
FFFF _H ÷ FF00 _H	obsahuje kopii bloku kalibračních a inicializačních konstant (tzn. blok 00FF _H ÷ 0000 _H), data jsou však dostupná výhradně pro čtení (tzn. blok je chráněn proti nechtěnému přepsání dat)

Jak je patrné z předešlého popisu:

- program nemusí blok kalibračních a inicializačních konstant nijak obsluhovat, protože jeho obsah je nastaven automaticky po zapnutí nebo restartu karty daty uloženými ve flash paměti karty (do flash paměti jsou konstanty zapisovány kalibračním programem)
- program může zapsat vlastní kalibrační konstanty; pro přesnou identifikaci karty lze využít výrobní číslo
- program může obnovit výchozí stav přenesením kopie bloku FFxx_H do 00xx_H

11.9 Registr CalibAdrReg (WR, RD)

Tento registr definuje 16bitovou adresu do bloku kalibračních a inicializačních konstant. Každým čtením paměti a každým uskutečněným zápisem do paměti (viz CalibCtrlReg) je obsah tohoto registru inkrementován.

Obsah registru je zpětně čitelný.

11.10 Registr CalibDataReg (WR, RD)

Tento registr zpřístupňuje obsah bloku kalibračních a inicializačních konstant, adresa je definována registrem CalibAdrReg. Každým čtením a každým provedeným zápisem (viz CalibCtrlReg) je obsah registru inkrementován.

11.11 Registr CalibCtrlReg (WR)

Tento registr povoluje zápis do bloku kalibračních a inicializačních konstant. Obsah registru je po startu vynulován, zápisem dat AA_H je povolen zápis do paměti a po jeho provedení (tzn. zápisem do CalibDataReg) je obsah registru CalibCtrlReg opět vynulován.

Před každým zápisem do bloku kalibračních a inicializačních konstant je tedy nutné nastavit obsah CalibDataReg.

11.12 Registr CalibStatReg (RD)

Tento registr signalizuje připravenost paměti kalibračních a inicializačních konstant, tzn. možnost číst nebo zapisovat obsah bloku kalibračních a inicializačních konstant. Po dobu přenosu dat z flash paměti do FPGA karty (trvá několik milisekund po zapnutí napájecího napětí nebo restartu FPGA) je registr nastaven na hodnotu 0, po dokončení přenosu a zpřístupnění obsahu prostřednictvím registrů popsaných v předcházejících odstavcích je nastaven na hodnoty 1.

11.13 Blok s kalibračními a inicializačními konstantami

Dále uvedená tabulka popisuje aktuálně implementované kalibrační a inicializační konstanty.

Upozornění: *Ačkoliv byly struktura a význam konstant vytvořeny se záměrem dopředné kompatibility, nelze vyloučit, že v budoucích verzích firmware nedojde ke změnám; v případě jejich využití je tedy potřeba ověřovat verzi FPGA pomocí registru FPGATypeReg.*

ADR	význam dat	
	název	popis
0001/0000	ADC_R0_K	multiplikační konstanta ADC pro rozsah 0 (tzn. zesílení 1x)
0003/0002	ADC_R0_Q	aditivní konstanta ADC pro rozsah 0 (tzn. zesílení 1x)
0005/0004	ADC_R1_K	multiplikační konstanta ADC pro rozsah 1 (tzn. zesílení 2x)
0007/0006	ADC_R1_Q	aditivní konstanta ADC pro rozsah 1 (tzn. zesílení 2x)
0009/0008	ADC_R2_K	multiplikační konstanta ADC pro rozsah 2 (tzn. zesílení 4x)
000B/000A	ADC_R2_Q	aditivní konstanta ADC pro rozsah 2 (tzn. zesílení 4x)
000D/000C	ADC_R3_K	multiplikační konstanta ADC pro rozsah 3 (tzn. zesílení 8x)
000F/000E	ADC_R3_Q	aditivní konstanta ADC pro rozsah 3 (tzn. zesílení 8x)
0011/0010	ADC_R4_K	multiplikační konstanta ADC pro rozsah 4 (tzn. zesílení 16x)
0013/0012	ADC_R4_Q	aditivní konstanta ADC pro rozsah 4 (tzn. zesílení 16x)
0015/0014	ADC_R5_K	multiplikační konstanta ADC pro rozsah 5 (tzn. zesílení 32x)
0017/0016	ADC_R5_Q	aditivní konstanta ADC pro rozsah 5 (tzn. zesílení 32x)
001F÷0018	rezerva	rezerva pro případné rozsahy 6 a 7
0021/0020	DAC0_R0_K	multiplikační konstanta DAC0 pro rozsah 0 (tzn. 0÷5 V)
0023/0022	DAC0_R0_Q	aditivní konstanta DAC0 pro rozsah 0 (tzn. 0÷5 V)
0025/0024	DAC0_R1_K	multiplikační konstanta DAC0 pro rozsah 1 (tzn. ±5 V)
0027/0026	DAC0_R1_Q	aditivní konstanta DAC0 pro rozsah 1 (tzn. ±5 V)
0029/0028	DAC0_R2_K	multiplikační konstanta DAC0 pro rozsah 2 (tzn. 0÷10 V)
002B/002A	DAC0_R2_Q	aditivní konstanta DAC0 pro rozsah 2 (tzn. 0÷10 V)
002F÷002C	...	rezerva pro případný rozsah 3
0031/0030	DAC1_R0_K	multiplikační konstanta DAC1 pro rozsah 0 (tzn. 0÷5 V)
0033/0032	DAC1_R0_Q	aditivní konstanta DAC1 pro rozsah 0 (tzn. 0÷5 V)
0035/0034	DAC1_R1_K	multiplikační konstanta DAC1 pro rozsah 1 (tzn. ±5 V)
0037/0036	DAC1_R1_Q	aditivní konstanta DAC1 pro rozsah 1 (tzn. ±5 V)
0039/0038	DAC1_R2_K	multiplikační konstanta DAC1 pro rozsah 2 (tzn. 0÷10 V)
003B/003A	DAC1_R2_Q	aditivní konstanta DAC1 pro rozsah 2 (tzn. 0÷10 V)
003F÷003C	...	rezerva pro případný rozsah 3
007F÷0040	...	rezerva pro další kalibrační konstanty
0081/0080	DAC0_R0_data	inicializační data DAC0 při rozsahu 0
0083/0082	DAC0_R1_data	inicializační data DAC0 při rozsahu 1
0085/0084	DAC0_R2_data	inicializační data DAC0 při rozsahu 2
0087/0086	...	rezerva pro případný rozsah 3
0089/0088	DAC1_R0_data	inicializační data DAC1 při rozsahu 0
008B/008A	DAC1_R1_data	inicializační data DAC1 při rozsahu 1
008D/008C	DAC1_R2_data	inicializační data DAC1 při rozsahu 2
008F/008E	...	rezerva pro případný rozsah 3
0090	DOOUT0_data	inicializační data DOOUT portu
00EF÷0091	...	rezerva pro další inicializační konstanty
00FF÷00F0	SerNo	sériové číslo (řetězec 16 ASCII znaků, viz popis v kapitole Servisní rozhraní) např. "174284001_____" (znak "1" představuje ASCII kód 31 _H na adrese 00F0 _H)

Jak je zřejmé z tabulky blok s kalibračními a inicializačními konstantami obsahuje

- samostatný pár kalibračních konstant pro každý rozsah A/D převodníku
- samostatný pár kalibračních konstant pro každý D/A převodník a každý rozsah
- inicializační hodnoty obou D/A převodníků a výstupního digitálního portu (jsou na výstup přeneseny po zapnutí nebo restartu karty)
- ASCII řetězec délky 16 znaků obsahující unikátní výrobní číslo karty (devítimístný řetězec obsahující výhradně čísla doplněný na konci sedmi mezerami)

11.14 Formát konstant a kalibrace A/D převodníku

Hodnota signálu naměřeného na analogovém vstupu přenášena kartou do počítače je přepočtena z hodnoty A/D převodníku (popř. z průměru 16 hodnot) lineární interpolací, tzn. zjednodušeně podle vzorce ...

$$AIN = K * (ADC + Q)$$

kde

AIN	je hodnota naměřeného signálu přenášena v 16bitovém přímém binárním kódu; např. pro rozsah ± 10 V
-10 V	hodnota 0
0 V	hodnota 32768
+10 V	hodnota 65535
K	je multiplikativní kalibrační konstanta; podle aktuálního rozsahu ADC_R0_K až ADC_R5_K 16bitová hodnota 0=65535 uložena v bloku představuje multiplikativní konstantu 1,0000 až 1,1250
ADC	je hodnota A/D převodníku v přímém binárním kódu; při rozsahu ± 10 V zpracovává A/D převodník signál v rozsahu cca $\pm 10,4$ V, tzn. hodnota A/D převodníku odpovídá signálům
-10,4 V	hodnota cca 0
-10 V	hodnota cca 1260
0 V	hodnota cca 32768
+10 V	hodnota cca 64276
+10,4 V	hodnota cca 65535
Q	je aditivní kalibrační konstanta; podle aktuálního rozsahu ADC_R0_Q až ADC_R5_Q hodnota 0 až 65535 představuje konstantu -32768 až 32767

Pro upřesnění - skutečný výpočet probíhá podle vzorce (uvedeno pro rozsah R0, tzn. ± 10 V) ...

$$AIN = (1 + ADC_R0_K / 524288) * [(ADC - 32768) + (ADC_R0_Q - 32768)] + 32768$$

Postup kalibrace

V první kroku kalibrace je potřeba vstupy propojit s GND (tzn. přivést nulové napětí) a pro všechny rozsahy postupně nastavit aditivními konstantami Q hodnotu 32768 v přenášných datech. V druhém kroku pak na vstup přivádět vhodná napětí s ohledem na kalibrovaný rozsah (ideálně větší než 90% rozsahu) a postupně pro všechny rozsahy nastavit multiplikativní konstantou K odpovídající hodnoty v přenášných datech.

Obvyklá hodnota konstanty K je přibližně 1,04 (tzn. hodnota registru cca 20000) a obvyklá hodnota konstanty Q kolísá v závislosti na zesílení v rozsahu ± 70 (tzn. hodnota registru 32700 až 32840).

Poznámka: Karta obsahuje jeden pár konstant K/Q pro každý rozsah.
Pokud vyhoví nepřesnost jednotlivých zesílení 2x až 32x vstupního zesilovače (maximálně $\pm 0,2\%$; typicky $\pm 0,1\%$), lze kalibrovat pouze rozsah ± 10 V a hodnotu ADC_R0_K použít pro všechny rozsahy.

11.15 Formát konstant a kalibrace D/A převodníků

Hodnota zapisovaná do D/A převodníků je přepočtena z hodnoty zapsané do registrů DACxReg lineární interpolací analogicky kalibraci A/D převodníku popsané v předešlém odstavci, rozdílem je rozsah multiplikativní konstanty K přibližně 0,8750 až 1,0000 (D/A převodník je schopen generovat napětí větší než odpovídá rozsahu a hodnotu zapsanou do registrů DACxReg je potřeba zmenšit).

Pro D/A převodník 0 a rozsah 0 má vzorec formát ...

$$DAC0 = K * DAC0Reg + Q$$

kde

DAC0	jsou data přenesená do D/A převodníku D/A převodníky generují napětí přesahující kalibrovaný rozsah (např. -0,1 V až 10,2 V pro rozsah 10 V) hodnota 0 generuje vždy největší záporné a hodnota 65535 největší kladné napětí v závislosti na rozsahu
K	je multiplikativní kalibrační konstanta; podle aktuálního rozsahu DAC0_R0_K až DAC0_R2_K 16bitová hodnota 0=65535 uložena v bloku představuje multiplikativní konstantu 0,8750 až 1,0000
DAC0Reg	je hodnota zapsaná do registru určeného pro řízení D/A převodníku pro unipolární rozsahy hodnota 0 generuje napětí 0 V a 65535 napětí +5 V nebo +10 V (podle rozsahu) pro bipolární rozsah hodnota 32768 generuje napětí 0 V, 0 napětí -5 V a 65535 napětí +5 V
Q	je aditivní kalibrační konstanta; podle aktuálního rozsahu DAC0_R0_Q až DAC0_R2_Q 16bitová hodnota 0 až 65535 představuje konstantu -32768 až 32767

Pro upřesnění - skutečný výpočet probíhá podle vzorců ...

$$\text{DAC0} = (0,875 + \text{DAC0_R0_K} / 524288) * \text{DAC0Reg} + (\text{DAC0_R0_Q} - 32768)$$

(vzorec pro unipolární rozsah R0, tzn. 5 V)

$$\text{DAC0} = (0,875 + \text{DAC0_R1_K} / 524288) * (\text{DAC0Reg} - 32768) + (\text{DAC0_R1_Q} - 32768) + 32768$$

(vzorec pro bipolární rozsah R1, tzn. ± 5 V)

$$\text{DAC0} = (0,875 + \text{DAC0_R2_K} / 524288) * \text{DAC0Reg} + (\text{DAC0_R2_Q} - 32768)$$

(vzorec pro unipolární rozsah R2, tzn. 10 V)

Výpočty pro D/A převodník 1 jsou analogický s tím, že všechny hodnoty a konstanty DAC0x jsou nahrazeny DAC1x.

Postup kalibrace

V první kroku kalibrace je potřeba pro hodnotu DACxReg odpovídající nulovému napětí (tzn. 0 nebo 32768 v závislosti na rozsahu, resp. zda je rozsah unipolární nebo bipolární) nastavit aditivní konstantou Q nulové výstupní napětí. V druhém kroku pak pro hodnotu DACxReg odpovídající maximálnímu napětí nastavit správné výstupní napětí.

Obvyklá hodnota konstanty K je přibližně 0,96 (tzn. hodnota registru cca 46000) a hodnota konstanty Q kolísá v závislosti na rozsahu od 250 do 800 (tzn. hodnota registru 33000 až 33600).

Poznámka: Karta obsahuje jeden pár konstant K/Q pro každý D/A převodník a každý rozsah; rozsahy jsou tedy kalibrovány samostatně, uvedený postup je třeba pro každý D/A převodník a každý rozsah opakovat.

12. Servisní rozhraní

12.1 Úvod

V následujících odstavcích bude popsán komunikační protokol servisního rozhraní.

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

12.2 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.

12.3 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".

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

12.5 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:

```
PCA-7428CS          typ karty
174284001          výrobní číslo karty (shodné se štítkem na kartě)
PCIloader v.1.20   identifikace firmware mikropočítače
FPGA-1D-0.9       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.

12.6 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

12.7 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

12.8 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

12.9 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

12.10 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

12.11 Rozdělení flash paměti

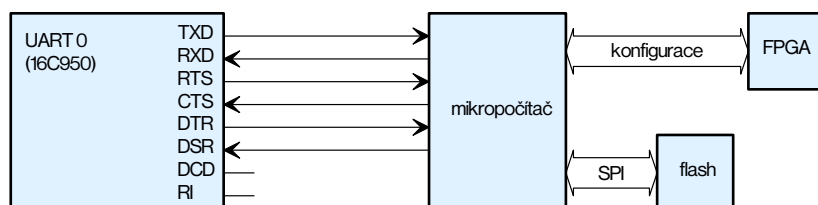
Flash má kapacitu 512 kB rozdělenou do 2048 stránek velikosti 256 B.

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)

12.12 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)

12.13 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 přerušeni 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>