

Application Note



SPI FLASH Core – Funkční vzorek řadiče paměti SPI Serial Flash

Jiří Kadlec, Leoš Kafka, Jiří Svozil

kadlec@utia.cas.cz

Obsah

1. Úvod.....	2
2. Popis systému.....	2
3. Parametry modulu.....	6
4. Použité/Potřebné vybavení a nástroje.....	7
5. Funkční vzorek HW modulu řadiče spi_flash_core pro desku S3E1600.....	7
6. Poděkování.....	10
7. SPI_FLASH_CORE demo.....	11
8. Licensing and availability (anglicky).....	11
9. Disclaimer (anglicky).....	11
10. Reference.....	12

Revize

Revize	Datum	Autor	Popis změn v dokumentu
0	20.10.2008	Jiří Svozil	Vytvoření dokumentu
1	13.11.2008	Jiří Svozil	Revize dokumentu
2	12.12.2009	Jiří Kadlec	Verze pro ISE 11.3 s popisem příkladu
3	31.12.2010	Jiří Kadlec	Verze pro ISE 12.4 s popisem příkladu
4	2.08.2011	Jiří Kadlec	Verze pro ISE 13.2 s popisem příkladu

1. Úvod

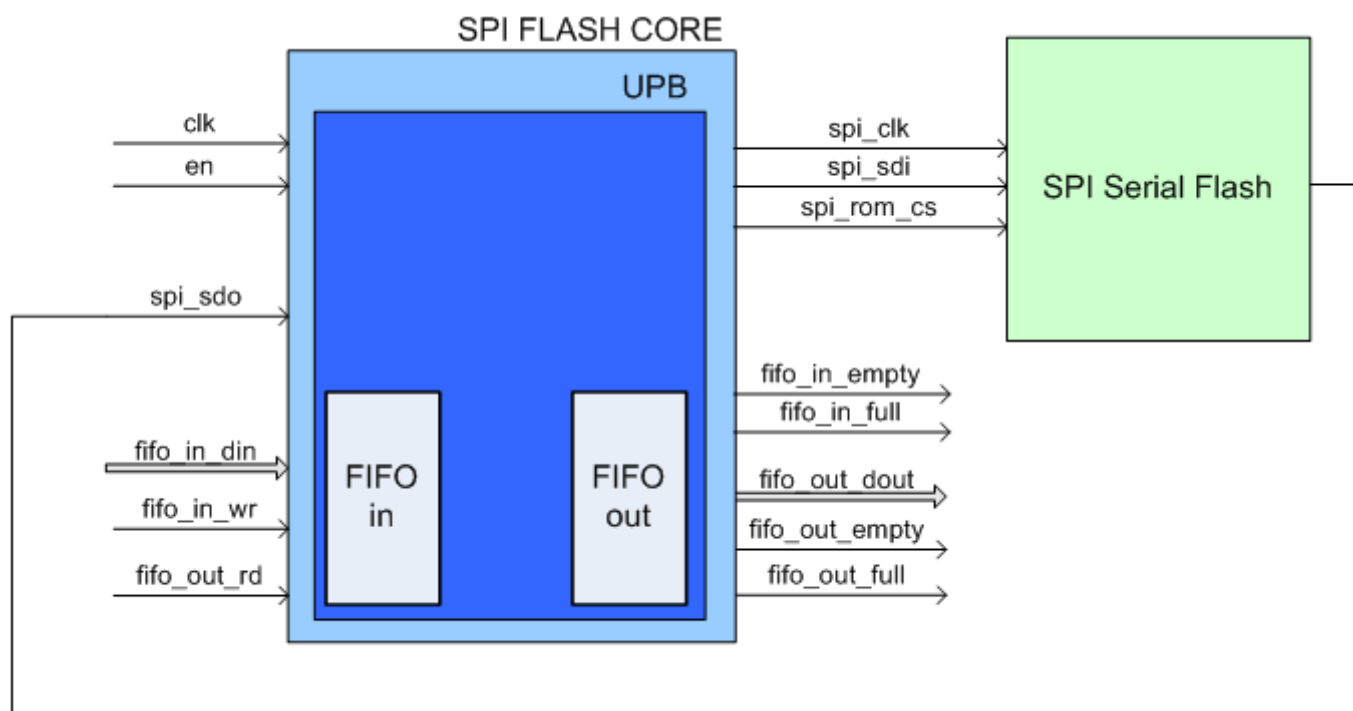
Tento dokument popisuje funkční vzorek HW modul řadiče paměti SPI Serial Flash. Jádrem je založeno na UPB (Universal PicoBlaze Wrapper) jehož autorem je Ing. Leoš Kafka. UPB je konfigurovatelný wrapper procesoru PicoBlaze [1]. HW modul stejně tak jako UPB byl vytvořen v rámci projektu VLAM.

Modul řadiče je vytvořen tak, aby uživateli umožnil práci s pamětí Flash na vyšší úrovni, tzn. i bez rozsáhlých znalostí použitého hardware. Uživatel může tento modul zařazovat do složitějších systémů, kde tento řadič pouze připojí a nemusí se zabývat samotným procesem řízení paměti. Řadič je nakonfigurován pro M25P16 SPI Serial Flash [5], která je osazena na přípravku Xilinx S3E1600 [2].

Testovací příklad vychází z příkladu [6f], jehož autorem je Ken Chapman.

2. Popis systému

Řadič paměti SPI Flash je založen na konfigurovatelném wrapperu UPB, který řídí činnost celého modulu a to včetně SPI komunikace. K ovládání modulu, stejně tak jako datový vstup slouží vstupní paměť FIFO o volitelné délce. Jako datový výstup slouží výstupní FIFO také o volitelné délce. Blokové schéma modulu znázorňuje Obr. 1.



Obrázek 1 Blokové schéma modulu řadiče paměti SPI Flash

Ovládání modulu se provádí pomocí příkazů zapsaných do vstupní paměti FIFO. Každá operace (příkaz) má přidělenou 8bitovou hodnotu. Přehled podporovaných příkazů je v tabulce 1.

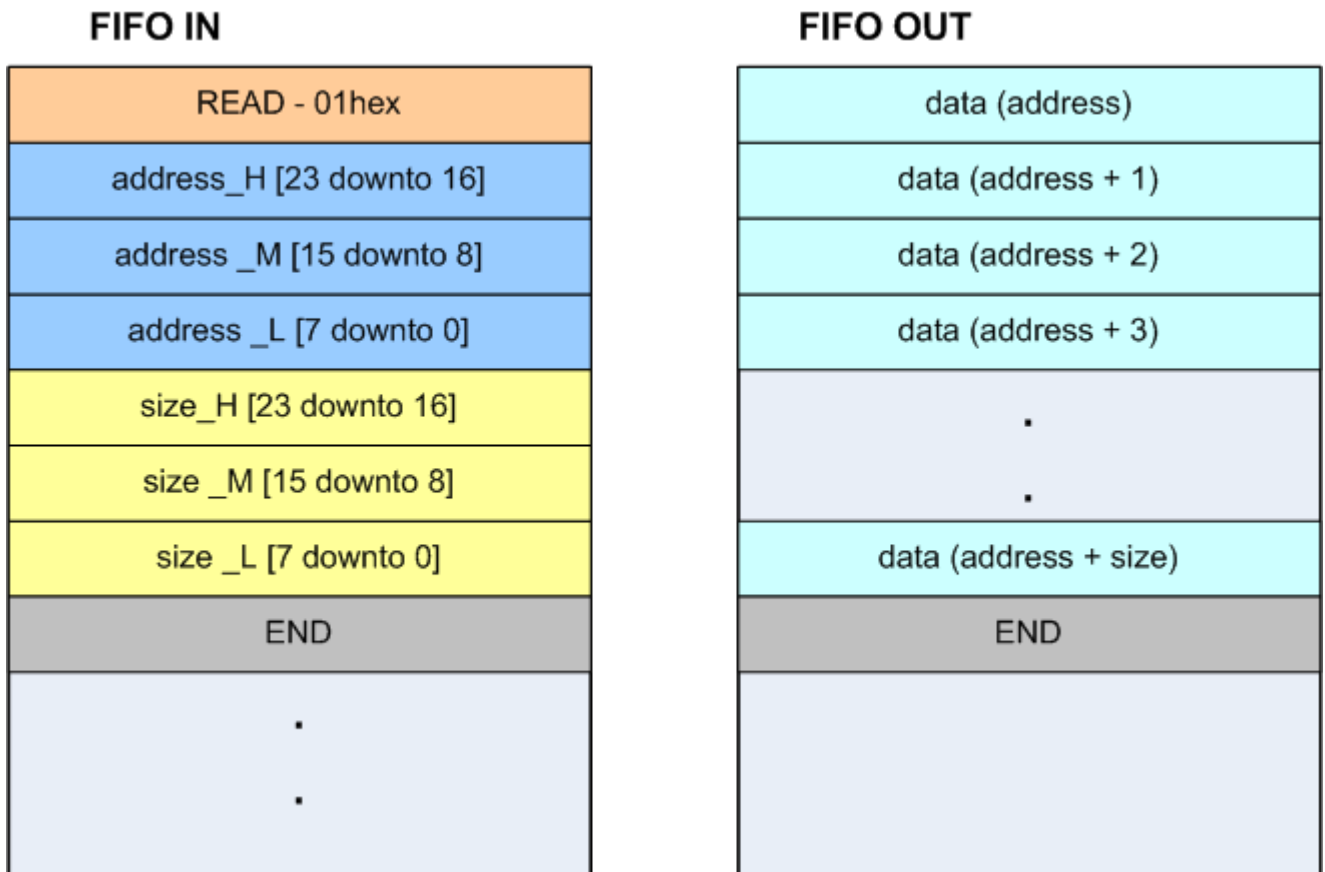
Tabulka 1 - Přehled příkazů modulu

příkaz	hodnota
READ	01hex
WRITE	02hex
ERASE SECTOR	04hex
ERASE FLASH	08hex

Ovládací příkaz musí být do vstupní paměti zapsán vždy jako první a teprve po něm následují další parametry příkazu (adresa, data, délka, atd.). Tyto parametry se liší dle zvolené operace.

READ – 01hex

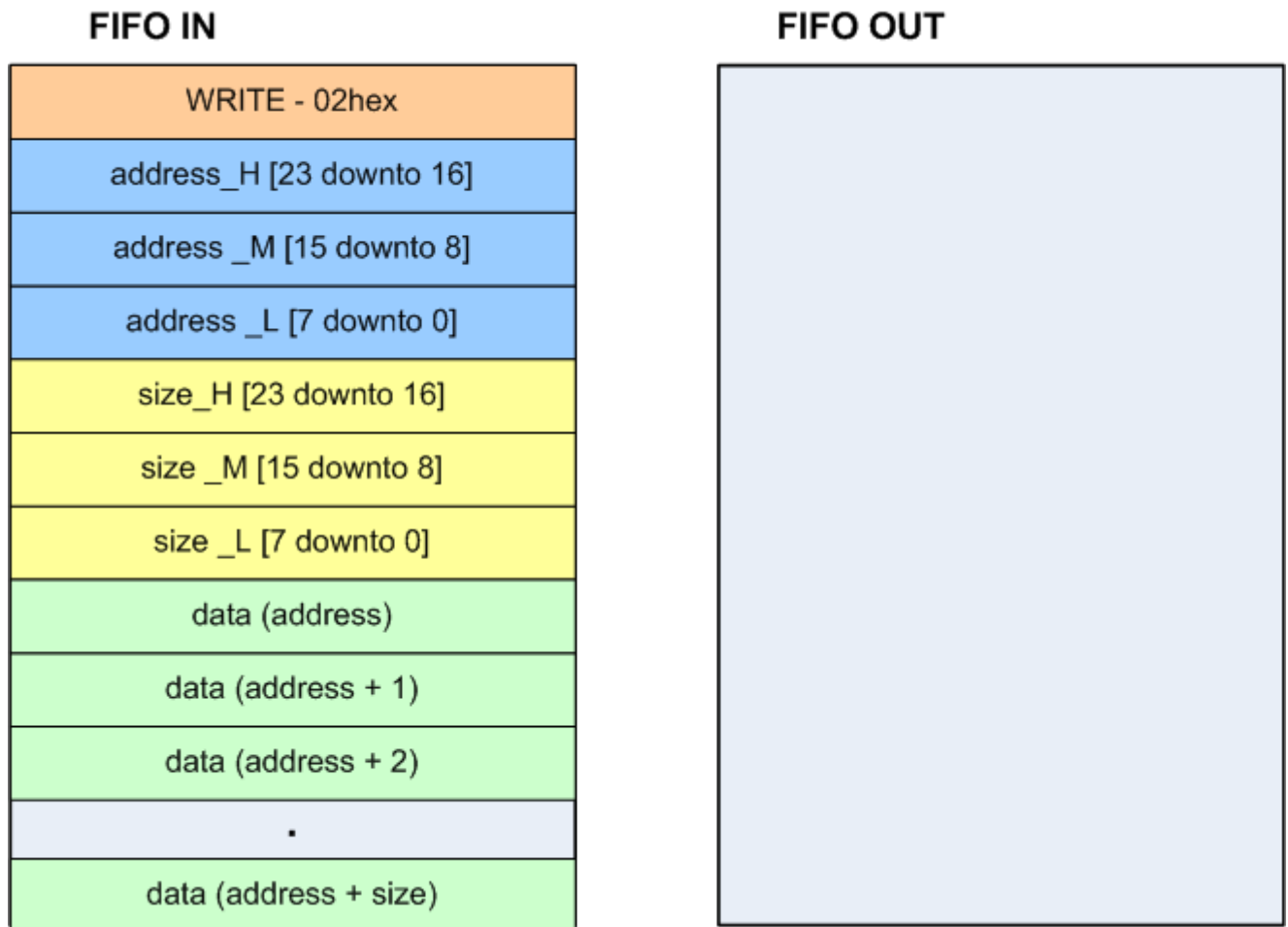
Příkaz READ umožňuje čtení libovolného počtu dat od libovolné adresy z paměti Flash. Po zápisu příkazu READ do FIFO následuje 24bitová hodnota představující adresu (address), od které se data začnou vyčítat, a dále pak 24bitová hodnota obsahující délku bloku (počet čtených bytů, size). Čtené hodnoty jsou zapisovány do výstupní FIFO. Sekvenci příkazu READ znázorňuje Obr. 2



Obrázek 2 Sekvence příkazu READ

WRITE – 02hex

Příkaz WRITE umožňuje zápis libovolného počtu dat od libovolné adresy. Po zápisu příkazu WRITE do FIFO následuje 24bitová hodnota adresy, od které se data začnou zapisovat, dále pak 24bitová hodnota počtu zapisovaných byte a nakonec data pro zápis. Sekvenci příkazu WRITE znázorňuje Obr. 3.



Obrázek 3 Sekvence příkazu WRITE

ERASE SECTOR

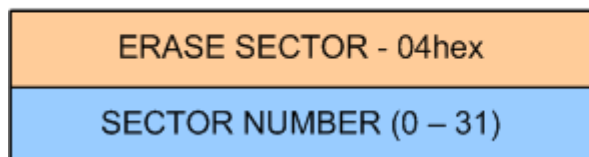
Příkaz ERASE SECTOR vymaže libovolný z 32 sektorů paměti Flash. Paměť je o velikosti 16Mbit, jeden sektor tedy odpovídá 512kbit. Po zápisu příkazu následuje číslo mazaného sektoru. Úspěšné dokončení mazání je potvrzeno zápisem hodnoty FFhex do výstupní FIFO. Sekvenci příkazu ERASE SECTOR znázorňuje Tabulka 2.

Číslo a rozsah adres pro jednotlivé sektory upřesňuje tabulka 2.

Tabulka 1 - Sektorové rozdělení paměti

Sektor číslo	Rozsah paměti
0	000000hex – 00FFFFhex
1	010000hex – 01FFFFhex
2	020000hex – 02FFFFhex
...	...
31	1F0000hex - 1FFFFFhex

FIFO IN



FIFO OUT



Obrázek 4 Sekvence příkazu ERASE SECTOR

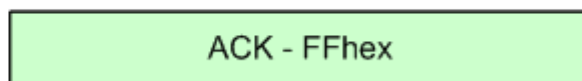
ERASE FLASH

Příkaz ERASE FLASH provede výmaz celé paměti Flash. Úspěšné dokončení mazání je potvrzeno zápisem hodnoty FFhex do výstupní FIFO. Sekvenci příkazu ERASE FLASH znázorňuje Obr. 5.

FIFO IN



FIFO OUT



Obrázek 5 Sekvence příkazu ERASE FLASH

Přehled IO portů modulu popisuje Tabulka 2.

Tabulka 2 - Přehled portů modulu SPI Flash

Port	Vstup/Výstup	Šířka [bit]	Význam
clk	Vstup	1	Hodinový signál
en	Vstup	1	Povolení funkce modulu (aktivní v log.1)
spi_sdo	Vstup	1	Vstup SPI dat
fifo_in_din	Vstup	8	Vstupní data FIFO IN
fifo_in_wr	Vstup	1	Zápis dat do FIFO IN
fifo_out_rd	Vstup	1	Čtení FIFO OUT
spi_clk	Výstup	1	Hodinový signál SPI
spi_sdi	Výstup	1	Datový výstup SPI
spi_rom_cs	Výstup	1	Chip-select (aktivní v log. 0)
fifo_in_empty	Výstup	1	Příznak prázdné FIFO IN
fifo_in_full	Výstup	1	Příznak plné FIFO IN
fifo_out_dout	Výstup	8	Výstupní data FIFO OUT
fifo_out_empty	Výstup	1	Příznak prázdné FIFO OUT
fifo_out_full	Výstup	1	Příznak plné FIFO OUT

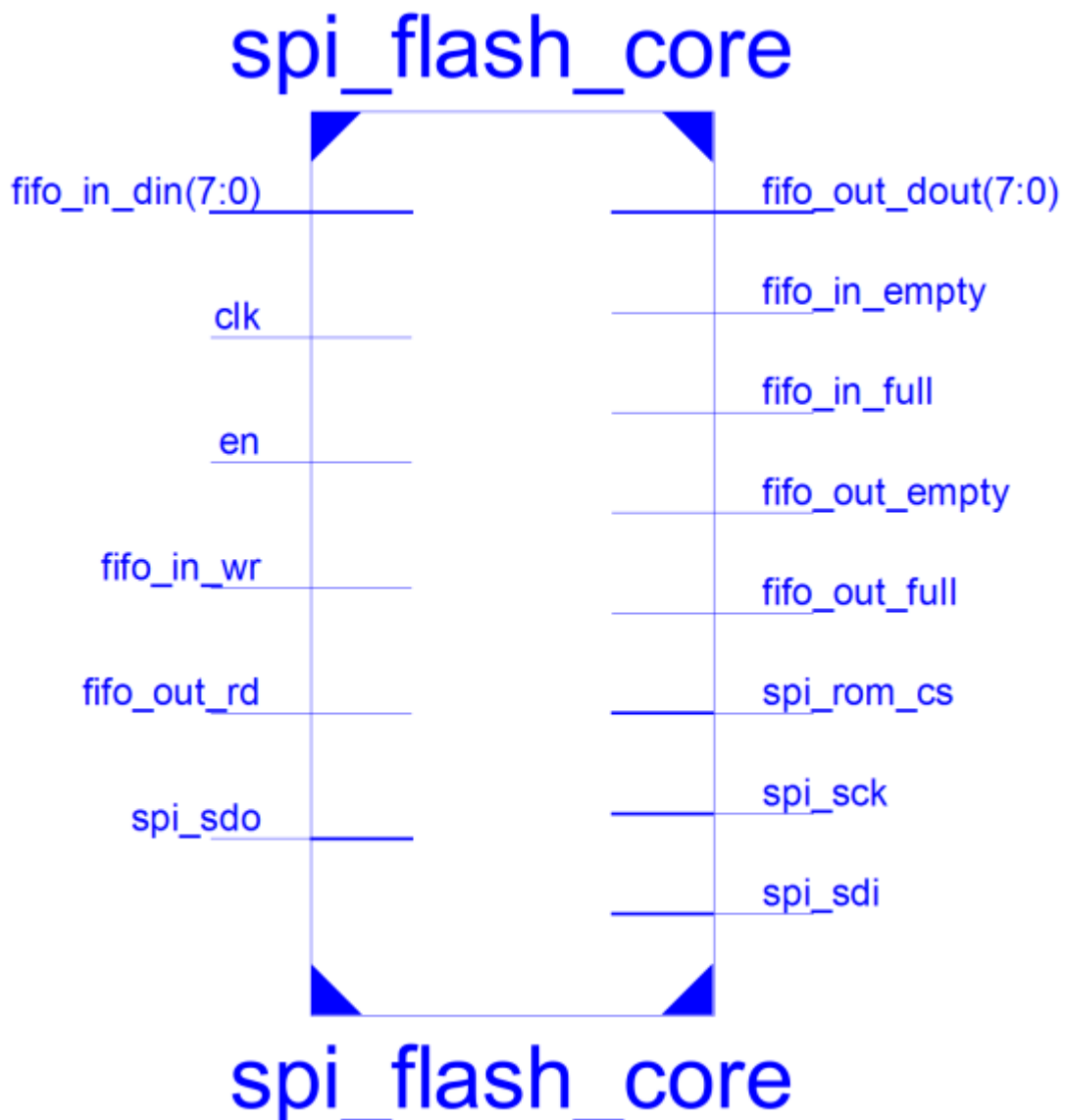
3. Parametry modulu

Velikost designu spi_flash_core modulu v ISE 13.2 pro hodiny systému 75 MHz uvádí tabulka 3.

Tabulka 3 - Velikost designu spi_flash_core

Number of Slices	186
Number of Slice Flip Flops	143
Number of 4 input LUTs	457
Number of BRAMs	1
Number of DCMs	0

Vstupy a výstupy funkčního vzorku spi_flash_core jsou znázorněny na obrázku 6.



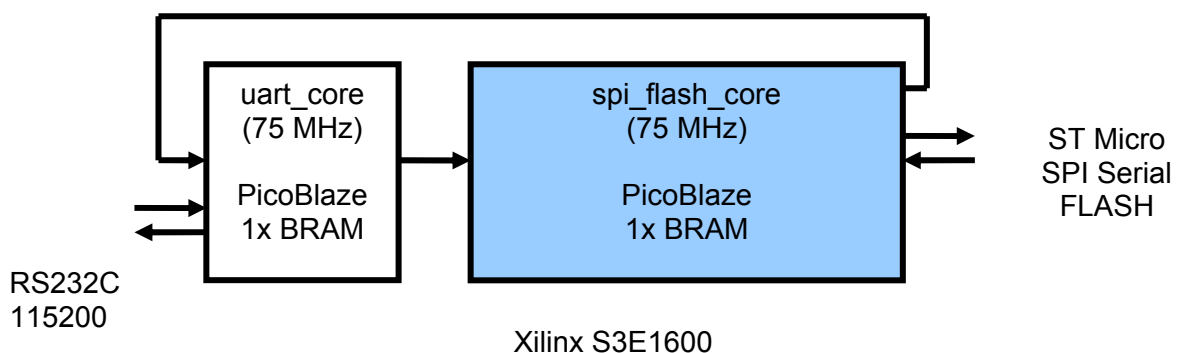
Obrázek 6 Modul spi_flash_core v ISE 13.2

4. Použité/Potřebné vybavení a nástroje

Modul generátoru frekvence byl napsán v programovacím jazyce VHDL ve vývojovém prostředí Xilinx ISE verze 13.2 [3]. Program procesoru PicoBlaze byl napsán v assembleru a k překladač byl použit assembler *KCPSM3.exe*, který je součástí balíčku procesoru PicoBlaze [1]. Testovací příklad byl vytvořen na vývojovém kitu Xilinx S3E1600 [2], viz příložené CD.

5. Funkční vzorek HW modulu řadiče spi_flash_core pro desku S3E1600

Příklad práce s pamětí SPI FLASH prostřednictvím modulu spi_flash_core. Příklad umožňuje pomocí sériové linky RS232C ověřit zápis, čtení a mazání paměti SPI FLASH na desce S3E1600. Modul uart_core interpretuje příkazy a data ze sériové linky. Modul uart_core komunikuje s modulem spi_flash_core. Blokové schéma funkčního vzorku zachycuje Obr. 7.



Obrázek 7: Blokové schéma funkčního vzorku s SPI_FLASH_CORE

Připojení desky S3E1600 k PC:

kabel pro sériový port (přímý kabel) mezi:
RS232C konektor PC (nebo USB adapter emulující RS232C)
RS232C konektor J9 (DCE) na desce S3E1600

Nastavení PC terminálu sériové linky:

- baud rate = 115200
- data bites = 8
- parity = N
- stop bits = 1
- připojovat kód odřádkování za přicházející řádky - ANO

Příklad reaguje na následující uživatelské příkazy zadatelné z PC terminálu po sériové lince:

- "r" read
- "w" write
- "e" erase flash
- "s" erase sector 0 flash (address 000000 - 00FFFF)
- "h" help

Příkaz read ("r"), přečte libovolný počet byte od zadané adresy:

- "read address" - počáteční adresa čtení (příklad: read adres= 01F000)
- "read size" - požadovaná délka čtení (v bytech) od počáteční adresy (příklad: read size= 000002)

Příkaz write ("w"), zapíše žádanou hodnotu na zadanou adresu

- "write adres" - adresa pro zápis (příklad: write adres= 01F000)

- "write byte" - hodnota pro zápis (příklad: write byte = AB)

Příkaz erase ("e"), vymaže celý obsah paměti

- po zadání příkazu je očekáváno potvrzení "Confirm Erase (Y/n)" - case sensitive!

- po dokončení potvrzeno "OK" / časová náročnost několik desítek sekund

Příkaz erase ("s"), vymaže sektor paměti (address 000000 - 00FFFF)

- po zadání příkazu je očekáváno potvrzení "Confirm Erase (Y/n)" - case sensitive!

- po dokončení potvrzeno "OK" / časová náročnost několik desítek sekund

Poznámka:

Funkce "Lock/Unlock block" není v této verzi modulu implementována. Pokud je blok, do kterého mají být zapsána data uzamčený, zápis se neprovede.

Konkrétní ukázka výpisu na Terminálu:

```
>w
write address=010000
write byte=01
OK
>w
write address=010001
write byte=02
OK
>r
read address=010000
read size=000005
OK
01 02 FF FF FF
>e
Confirm Erase (Y/n) Y
Erase in Progress
OK
>r
read address=010000
read size=000005
OK
FF FF FF FF FF
>
```

Velikost a proudový odběr implementovaného demonstrátoru funkčního vzorku v ISE 13.2 je pro desku 3SE1600 a hodiny systému 75 MHz uvedena v tabulce 4.

Tabulka 4 – Příklad ADC_CORE velikost a odběr celého HW

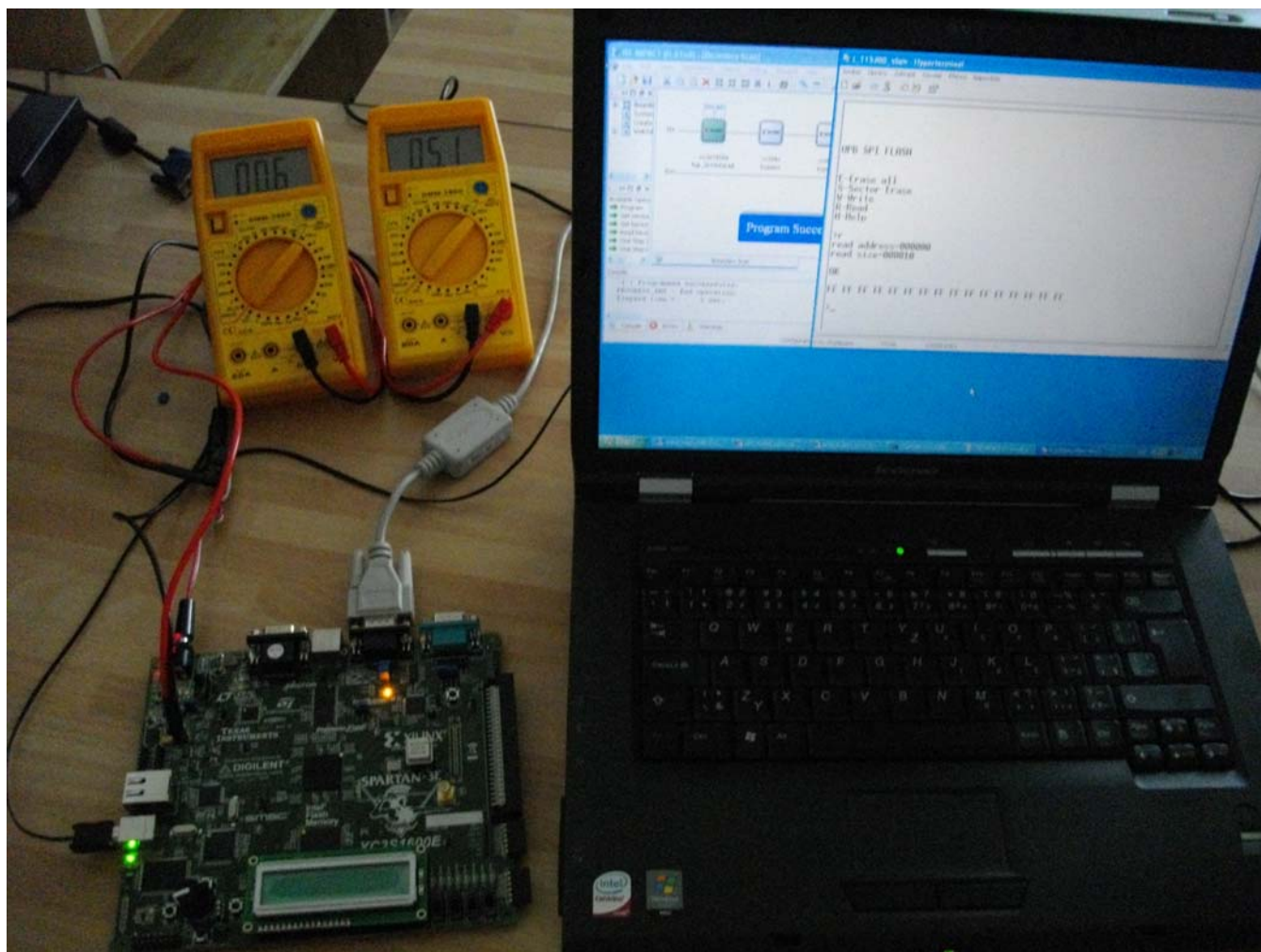
	XC3S1600E-4
	top_3s1600e
Number of Slices	530
Number of Slice Flip Flop	325
Number of 4 input LUTs	984
Number of BRAMs	2
VCCINT 1,2V (FPGA)	60mA
Odběr na 5,0V zdroji	510mA

Vstupy a výstupy demonstrátoru funkčního vzorku spi_flash_core popisuje tabulka 5 pro desku S3E1600

Tabulka 5 – Vstupy a výstupy demonstrátoru funkčního vzorku spi_flash_core na desce S3E1600

	<p>TIMESPEC TS01 = FROM : FFS : TO : FFS : 20 ns; TIMESPEC TS02 = FROM : RAMS : TO : FFS : 20 ns; TIMESPEC TS03 = FROM : FFS : TO : RAMS : 20 ns; TIMESPEC TS04 = FROM : RAMS : TO : RAMS : 20 ns; TIMESPEC TS05 = FROM : FFS : TO : PADS : 20 ns; TIMESPEC TS06 = FROM : PADS : TO : FFS : 20 ns; TIMESPEC TS07 = FROM : PADS : TO : RAMS : 20 ns;</p> <p>NET "clk" LOC = "C9";</p> <p>NET "tx_female" LOC = "M14"; NET "rx_female" LOC = "R7";</p> <p>NET "spi_sck" LOC = "U16"; NET "spi_sdi" LOC = "T4"; NET "spi_sdo" LOC = "N10"; NET "spi_rom_cs" LOC = "U3"; NET "spi_amp_cs" LOC = "N7"; NET "spi_adc_conv" LOC = "P11"; NET "spi_dac_cs" LOC = "N8"; NET "spi_amp_shdn" LOC = "P7"; NET "spi_dac_clr" LOC = "P8";</p> <p>NET "strataflash_oe" LOC = "C18"; NET "strataflash_ce" LOC = "D16"; NET "strataflash_we" LOC = "D17";</p> <p>NET "platformflash_oe" LOC = "T3";</p> <p>NET "led_hw<0>" LOC = "F12" IOSTANDARD = LVTTTL SLEW = SLOW DRIVE = 4; NET "led_hw<1>" LOC = "E12" IOSTANDARD = LVTTTL SLEW = SLOW DRIVE = 4; NET "led_hw<2>" LOC = "E11" IOSTANDARD = LVTTTL SLEW = SLOW DRIVE = 4; NET "led_hw<3>" LOC = "F11" IOSTANDARD = LVTTTL SLEW = SLOW DRIVE = 4; NET "led_hw<4>" LOC = "C11" IOSTANDARD = LVTTTL SLEW = SLOW DRIVE = 4; NET "led_hw<5>" LOC = "D11" IOSTANDARD = LVTTTL SLEW = SLOW DRIVE = 4; NET "led_hw<6>" LOC = "E9" IOSTANDARD = LVTTTL SLEW = SLOW DRIVE = 4; NET "led_hw<7>" LOC = "F9" IOSTANDARD = LVTTTL SLEW = SLOW DRIVE = 4;</p>
--	---

Demonstrátor funkčního vzorku spi_flash_core na desce S3E1600 je zobrazen na obr. 8.



Obrázek 8: Funkční vzorek spi_flash_core na desce S3E1600

6. Poděkování

Návrh a realizace tohoto funkčního vzorku byla podpořena projektem ministerstva školství a tělovýchovy číslo 2C06008, <http://www.vlam.cz>.

7. SPI_FLASH_CORE demo

S3E1600\ impl\ spi_flash_core\ spi_flash_core.ngc	Modul pro cteni a programovani ST Micro Serial Flash M25P16 na 3s1600e
spi_flash_uart\ spi_flash_uart.ngc	Modul RS232C terminalu
dcm_mhz\ dcm_mhz.ngc	Modul generující hodinový signál 75 MHz
top_3s1600e\ ...	Projekt pro překlad funkčního vzorku pomocí Xilinx ISE 13.2
top_3s1600e_bit\ top_3s1600e_mcf\ src\ ucf\ top_3s1600e.ucf	Konfigurační data funkčního vzorku Konfigurační data pro platform flash
vhdl\ top_3s1600e.vhd	Definice vstupů a výstupů pro 3s1600e Zdrojový kód ve VHDL (top level)

8. Licensing and availability (anglicky)

This functional sample is provided in form of netlist compiled by Xilinx ISE 13.2 [3] for the Xilinx xc3s1600e-4fg320 [2]. If you plan to use this functional sample netlist for education purposes, you can contact UTIA AV CR, v.v.i. for support. The contact person in UTIA is Jiri Kadlec kadlec@utia.cas.cz tel. +420 2 6605 2216 fax. +420 2 6605 2511.

If you consider commercial use of this functional sample in form of netlist compiled by Xilinx ISE 13.2 [3] for the Xilinx xc3s1600e-4fg320 FPGAs or if you need the source code of this functional sample, please contact UTIA AV CR v.v.i. Commercial End User License Agreement (Commercial EULA) between you and UTIA AV CR, v.v.i. needs to be signed. The contact person in UTIA is Jiri Kadlec kadlec@utia.cas.cz tel. +420 2 6605 2216 fax. +420 2 6605 2511.

9. Disclaimer (anglicky)

This disclaimer is not a license and does not grant any rights to the materials distributed herewith. Except as otherwise provided in a valid license issued to you by UTIA AV CR v.v.i., and to the maximum extent permitted by applicable law:

- (1) THIS APPLICATION NOTE AND RELATED MATERIALS LISTED IN THE PACKAGE CONTENT ARE MADE AVAILABLE "AS IS" AND WITH ALL FAULTS, AND UTIA AV CR V.V.I. HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and
- (2) UTIA AV CR v.v.i. shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under or in connection with these materials, including for any direct, or any indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or UTIA AV CR v.v.i. had been advised of the possibility of the same.

Critical Applications:

UTIA AV CR v.v.i. products are not designed or intended to be fail-safe, or for use in any application requiring fail-safe performance, such as life-support or safety devices or systems, Class III medical devices, nuclear facilities, applications related to the deployment of airbags, or any other applications that could lead to death, personal injury, or severe property or environmental damage (individually and collectively, "Critical Applications"). Customer assumes the sole risk and liability of any use of UTIA AV CR v.v.i. products in Critical Applications, subject only to applicable laws and regulations governing limitations on product liability.

10. Reference

- [1] PicoBlaze 8-bit Embedded Processor UG129 June 22, 2011
http://www.xilinx.com/support/documentation/ip_documentation/ug129.pdf
- [2] MicroBlaze Development Kit Spartan-3E 1600E Edition User Guide UG257 (v1.1) December 5, 2007
http://www.xilinx.com/support/documentation/boards_and_kits/ug257.pdf
- [3] Xilinx ISE 13.2, <http://www.xilinx.com/>
- [4] řadič HD44780 <http://www.datasheetarchive.com/HD44780-datasheet.html>
- [5] STMicroelectronics M25P16 SPI Serial Flash Data Sheet
<http://www.st.com/stonline/books/pdf/docs/10027.pdf>
- [6] Xilinx Spartan-3E FPGA Starter Kit Board Design Examples
http://www.xilinx.com/products/boards/s3estarter/reference_designs.htm
- [6a] Ken Chapman, Initial Design for Spartan-3E Starter Kit (LCD Display Control), ISE 8.1i, Xilinx Ltd 16th, February 2006
http://www.xilinx.com/products/boards/s3estarter/files/s3esk_startup.pdf
- [6b] Ken Chapman, Rotary Encoder Interface for Spartan-3E Starter Kit, ISE 8.1i, Xilinx Ltd, 20th February 2006
http://www.xilinx.com/products/boards/s3estarter/files/s3esk_rotary_encoder_interface.pdf
- [6c] Ken Chapman, Amplifier and A/D Converter Control for Spartan-3E Starter Kit, ISE 8.1i, Xilinx Ltd, 23rd February 2006
www.xilinx.com/products/boards/s3estarter/files/s3esk_picoblaze_amplifier_and_adc_control.pdf
- [6d] Ken Chapman, D/A Converter Control for Spartan-3E Starter Kit, ISE 8.1i, Xilinx Ltd, 21st February 2006
http://www.xilinx.com/products/boards/s3estarter/files/s3esk_picoblaze_dac_control.pdf
- [6e] Ken Chapman, SPI FLASH Programmer for Spartan-3E Starter Kit, ISE 8.1i, Xilinx Ltd, March 2006
http://www.xilinx.com/products/boards/s3estarter/files/s3esk_picoblaze_spi_flash_programmer.pdf
- [6f] Ken Chapman, SPI FLASH Programmer for Spartan-3E Starter Kit, ISE 7.1i, Xilinx Ltd, November 2004
http://www.xilinx.com/products/boards/s3estarter/files/s3esk_picoblaze_spi_flash_programmer.pdf
- [6g] Ken Chapman, Frequency Generator for Spartan-3E Starter Kit, ISE 8.2i, Xilinx Ltd, 18th July 2006
http://www.xilinx.com/products/boards/s3estarter/files/s3esk_frequency_generator.pdf
- [6h] Ken Chapman, Frequency Counter for Spartan-3E Starter Kit (with test oscillators), ISE 8.1i, Xilinx Ltd, 7th March 2006
http://www.xilinx.com/products/boards/s3estarter/files/s3esk_frequency_counter.pdf
- [6i] Ken Chapman, Software Implementation of Pulse Width Modulation (PWM). A reference design using the Spartan-3E Starter Kit, ISE 8.1i, Xilinx Ltd, 24th May 2006
http://www.xilinx.com/products/boards/s3estarter/files/s3esk_picoblaze_pwm_control.pdf