

IMPLEMENTACIJA SPI KOMUNIKACIONOG KONTROLERA NA FPGA ČIPU

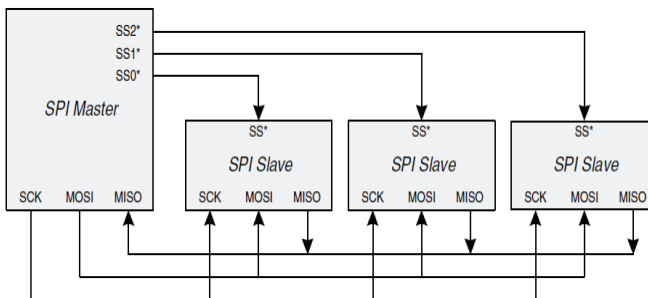
Dejan Mirković, *Elektronski fakultet, Univerzitet u Nišu*, dejan.mirkovic@elfak.ni.ac.rs
Milunka Damjanović, *Elektronski fakultet, Univerzitet u Nišu*, milunka.damjanovic@elfak.ni.ac.rs
Milun Jevtić, *Elektronski fakultet, Univerzitet u Nišu*, milun.jevtic@elfak.ni.ac.rs

Sadržaj – U radu je predstavljena jedna realizacija komunikacionog kontrolera koji zadovoljava SPI komunikacioni protokol. Kontroler se može konfigurisati kao master ili slave komunikacioni uređaj. Realizovan je korišćenjem VHDL jezika i implementiran na Alterinom FPGA čipu. Za verifikaciju je kreirana test sekvenca. Prikazani rezultati simulacije i snimljeni talasni oblici napona u komunikaciji između dva realizovana kontrolera u EP2C20F484C7 Altera FPGA čipovima dokazuju korektno funkcionisanje projektovanog kontrolera.

1. UVOD

SPI je akronim od Serial Peripheral Interface i predstavlja jednu od najrasprostranjenijih komunikacionih veza za serijsku razmenu podataka. Ovu komunikacionu vezu prvi put je hardverski realizovala kompanija Motorola u mikrokontrolerima serije MC68HCXX za komunikaciju sa perifernim uređajima. Ubrzo su i ostali proizvođači mikrokontrolera (Atmel, Microchip, ...) prihvatili SPI rešenje. Kako sam naziv SPI nije *trademark*, proizvođači čipova su implementirali različite verzije SPI komunikacije. Koristi se za komunikaciju sa EEPROM memorijama, memorijskim karticama i tako dalje.

Glavna karakteristika SPI komunikacije je brz sinhroni serijski prenos digitalnih podataka na malim rastojanjima. SPI se u literaturi najčešće označava kao full-duplex, serijski interfejs sa četiri signalne linije, mada se u praksi sreću realizacije i sa tri signalne linije [1]. Na slici 1. prikazana je blok šema interkonekcije više uređaja (jedan *master* i tri nezavisna *slave*) preko SPI komunikacione veze.



Slika 1. Blok šema interkonekcije uređaja koji učestvuju u SPI komunikaciji

Mada uređaji mogu preko SPI-a biti ulančani, ovakvoj vrsti komunikacije uvek se jedan od uređaja identifikuje kao *master* a ostali kao *slave* uređaji. Način označavanja signalnih linija na slici 1. odgovara Motorolinim oznakama koje imaju sledeće značenje: MOSI (*Master Out Slave In*) – signalna linija kojom se bit podatka prenosi od *master* ka *slave* uređaju, MISO (*Master In Slave Out*) – signalna linija kojom se bit podatka prenosi od *slave* ka *master* uređaju, SCK

(Serial Clock) – signalna linija za serijski takt, SS0-SS2 (*Slave Select*) – signalna linija(e) za selekciju odgovarajućeg *slave* uređaja. Generalno SPI se može posmatrati kao interfejs od $3 + N$ komunikacionih linija gde je N broj *slave* uređaja. Simboličko označavanje signalnih linija kao i njihov broj varira od proizvođača do proizvođača. Tako, na primer, kod *Microchip* kontrolera MOSI/MISO signalnim linijama odgovaraju oznake SDO/SDI (Serial Data Out/Serial Data In) gledano na strani *master* uređaja, i obrnuto, dok se kod *Fujicu* kontrolera čak i ne realizuju SS linije [2]. Slično važi i za tip pina (porta) na koji se ove signalne linije dovode. Negde su to ulazni ili izlazni portovi dok se negde realizuju i kao bidirekciono (uazno izlazni) [3].

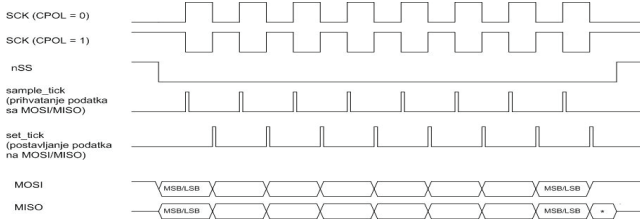
U narednim poglavljima biće detaljnije objašnjen sam SPI protokol kao i VHDL opis projektovanog SPI kontrolera. Takođe je dat pregled rezultata funkcionalne simulacije kojom je verifikovana ispravnost rada ovog komunikacionog modula. Na kraju je kontroler implementiran na Alterinom FPGA čipu EP2C20F484C7 korišćenjem DE1 razvojna ploča i odgovarajuće Quartus II, softversko okruženje.

2. SPI PROTOKOL

SPI se definiše kao protokol razmene podataka. Nijedan uređaj ne može da bude samo predajnik ili samo prijemnik. Sam prenos inicira *master* uređaj postavljanjem podatka u odgovarajući interni registar. Kod *slave* uređaja komunikacija se inicira njegovom selekcijom tj. detektovanjem niskog nivoa na SS signalnoj liniji. *Master* uređaj kontroliše prenos generisanjem takta na SCK portu, dok *slave* ne sme dodeljivati vrednost SCK liniji tj. SCK port na *slave* strani mora biti u stanju visoke impedanse [4].

Kada se radi o komunikaciji sa više *slave* uređaja, MISO linija se deli među *slave* uređajima. To znači da se za nju mora implementirati neki od mehanizama zaštite od kolizije podataka. Drugim rečima ovaj port mora biti ili trostatički ili open collector. *Master* inicijalno postavlja SS liniju u stanje logičke jedinice, tako da se selekcija odgovarajućeg *slave* uređaja vrši dodeljivanjem stanja logičke nule ovoj signalnoj liniji od strane *master*. Na taj način se odgovarajući *slave* uređaj "adresira", a da pri tome nije potrebno poslati inicijalni adresni bajt kao što je to slučaj kod I2C komunikacije [5]. Kada se radi o komunikaciji sa jednim *slave* uređajem, SS port na *slave* strani se mora postaviti na nivo logičke nule (obično preko pull-down otpornika). U ovom slučaju SS port kod *master* uređaja nema funkciju, tako da se njegovo stanje definiše postavljanjem na nivo logičke jedinice (obično preko pull-up otpornika).

Iz razloga kompatibilnosti sa raznim proizvođačima inicijalno stanje signala na SCK liniji može biti logička nula ili logička jedinica, zavisno od vrednosti signala - programirljivog bita CPHA.



Slika 2. Format prenosa podataka za CPHA = 0.

Prenos podataka je sinhron sa ivicom takta na SCK liniji. Prema SPI protokolu bit podatka se na jednu ivicu taktnog signala postavlja na MOSI/MISO liniju, a na drugu prihvata tj. sempluje sa MOSI/MISO linije nezavisno od polariteta takta. Na osnovu tipa ivice taktnog signala (rastuća - prednja ili opadajuća - zadnja) na koju se dešavaju promene na MOSI/MISO liniji, mogu se definisati sledeći formati prenosa podatka putem SPI interfejsa:

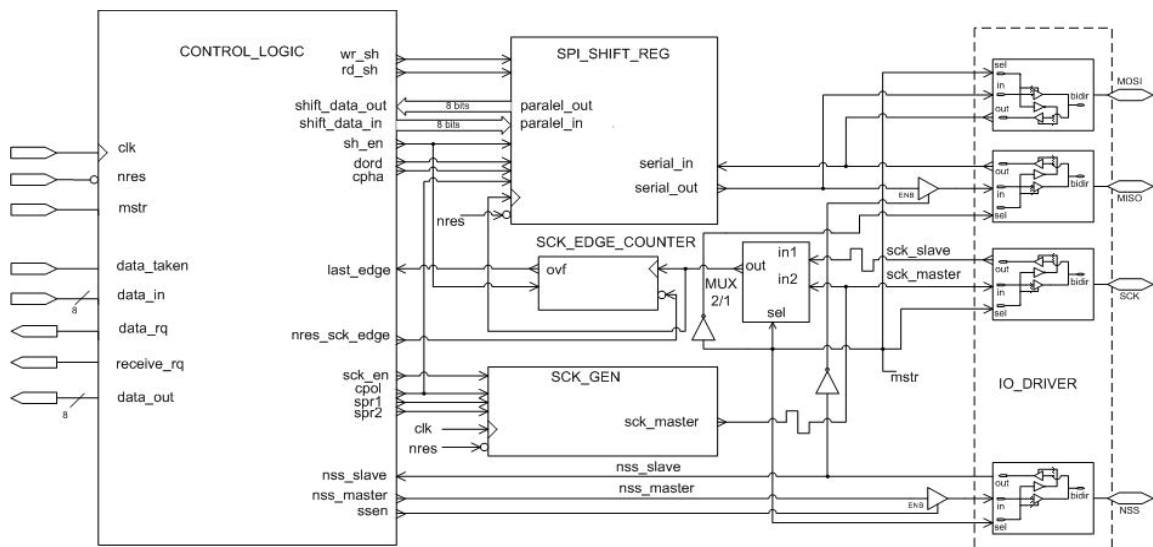
Način prenosa bajta podatka kada je CPHA = 0

Prednja ivica taktnog signala na SCK portu se koristi za prihvatanje prvog bita podatka sa MOSI/MISO linije. Neophodno je da prvi bit podatka bude dostupan na MOSI/MISO liniji najmanje pola taktnog perioda pre pojave prednje ivice taktnog signala na SCK liniji. Zadnjom ivicom taktnog signala novi bit podatka se postavlja na MOSI/MISO liniju. Ovaj postupak se ponavlja za svih 16 ivica taktnog signala na SCK. Na neparne ivice taktnog signala bit podatka se prihvata sa MOSI/MISO linije, a na parne postavlja na MOSI/MISO liniju kao što je prikazano na slici 2.

Način prenosa bajta podatka kada je CPHA = 1

Ovaj način prenosa je praktično komplementaran prethodnom. U ovom slučaju na prednju ivicu taktnog signala se bit podatka postavlja, a na zadnju prima preko MOSI/MISO linije. Prvi bit u povorci može biti LSB ili MSB bit podatka koji se šalje. Takođe i veličina podatka koji se šalje može da varira u zavisnosti od realizacije. U praksi se najčešće sreće veličina od jednog bajta ili jende reči (dva bajta [6]).

3. VHDL OPIS SPI MASTER/SLAVE KONTROLERA



Slika 3. Blok šeme SPI digitalnog bloka.

Prilikom projektovanja SPI digitalnog bloka usvojena je arhitektura prikazana na slici 3. Sa blok šeme se uočavaju osnovni gradivni blokovi od kojih su ključni: CONTROL_LOGIC, SPI_SHIFT_REG, CLK_GEN, IO_DRIVER, SCK_EDGE_COUNTER i MUX2/1.

CONTROL_LOGIC blok predstavlja upravljačku jedinicu i realizovan je kao konačni automat. SPI_SHIFT_REG je blok koji objedinjuje pomerački registar i logiku za postavljanje odnosno prihvatanje podatka sa MOSI/MISO linije. Način promena stanja na MOSI/MISO liniji shodno SPI protokolu odlučuje se setovanjem odgovarajućih upravljačkih signal od strane upravljačke jedinice (*dord*, *cpha* i *cpol*). Blok SCK_EDGE_COUNTER služi za brojanje ivica taktnog signala i detektovanja zadnje ivice istog.

SCK_GEN je blok u kome se generiše taktni signal SCK linije koji određuje takt prenosa podataka. Slično kao i u prethodnom slučaju selekcija odgovarajuće učestanosti i polariteta taktnog signala se obezbeđuje upravljačkim signlima iz bloka SPI_CONTROL (*spr1*, *spr2* i *cpol*).

U zavisnosti od toga da li se radi o *master* ili *slave* uređaju odgovarajući taktni signal se propušta ka SPI_SHIFT_REG i SCK_EDGE_COUNTER bloku preko MUX2/1 multipleksera. Selekcija se vrši upravljačkim signalom *mstr*. Pored toga, ovaj signal određuje i funkciju bidirekcionih portova preko odgovarajućeg IO_DRIVER bloka.

IO_DRIVER je bidirekciono trostatički bafer. Ovim blokom se obezbeđuje da se svaki bidirekciono port, u slučaju kada nije konfigurisan kao ulazni, postavi u stanje visoke impedanse i praktično oslobodi odgovarajuću SPI liniju.

Kada se radi o komunikaciji sa više *slave* uređaja mora se razrešiti stanje MISO linije koja je deljiva među *slave* uređajima. U tom slučaju *slave* uređaj će moći da postavi svoj podatak na MISO liniju tek kada se detektuje stanje logičke nule na NSS liniji. To se obezbeđuje postavljanjem upravljačkog signala *ssen* kod *master* uređaja. Na taj način se *nss_master* (ulaz IO_DRIVER bloka zaduženog za NSS liniju) signal propušta na NSS liniju *master* uređaja. Kada je uređaj konfigurisan kao *slave*, ovaj signal će se pojaviti na

Tabela 1. Sadržaj konfiguracionog registra *config_reg*

Bit	Oznaka	Značajne	Aktivnost
7	-	Ne koristi se.	-
6	<i>ssen</i>	<i>Slave Select Enable</i> - Bit za dozvolu postavljanja <i>nss_master</i> signala na NSS liniju u <i>master</i> modu.	1 – dozvola, 0 – zabrana.
5	<i>dord</i>	Bit kojim se određuje redosled postavljanja podataka na MOSI/MISO liniji.	1 – MSB prvi u povorci, 0 – LSB prvi u povorci.
4	<i>cpol</i>	Bit kojim se određuje inicijalno stanje taktnog signala na SCK liniji.	1 – inicijalno 1, 0 – inicijalno 0
3	<i>cpha</i>	Bit kojim se određuje redosled postavljanja na i prihvatanja sa MOSI/MISO linija.	1 – neparne ivice postavljanje, parne prihvatanje, 0 – neparne ivice prihvatanje, parne postavljanje.
2	<i>mstr</i>	Bit kojim se određuje <i>master/slave</i> mod rada.	1 – <i>master</i> , 0 – <i>slave</i>
1	<i>spr1</i>	Bit kojim se određuje frekvencija taktnog signala na SCK liniji	$spr1 \& spr2: 00 - fosc/4, 01 - fosc/16, 10 -$
0	<i>spr2</i>	Bit kojim se određuje frekvencija taktnog signala na SCK liniji	$fosc/64, 11 - fosc/128.$

njegovoj *nss_slave* liniji (izlaz IO_DRIVER bloka zaduženog za NSS liniju) i propustiti podatak na MISO liniju.

U komunikaciji sa jednim *slave* uređajem, NSS port je neophodno dovesti na nivo logičke nule, dok kod *master* uređaja ovaj port nema funkciju i ostaje u stanju visoke impedanse. Dobra je praksa da se stanje ovog porta definiše eksterno, postavljanjem na nivo logičke jedinice. Inicijalno stanje takta, redosled postavljanja bita podatka (tj. smer pomeranja podatka) na MOSI/MISO linije i frekvencija taktnog signala na SCK definišu se upravljačkim signalima *cpol*, *dord*, *spr1*, *spr2* respektivno. Svi pomenuti upravljački signali se definišu putem konfiguracionog registra *config_reg*. Detaljnije objašnjenje kao i ilustrativni prikaz ovog registra prikazani su u Tabeli 1.

4. REZULTATI LOGIČKE SIMULACIJE

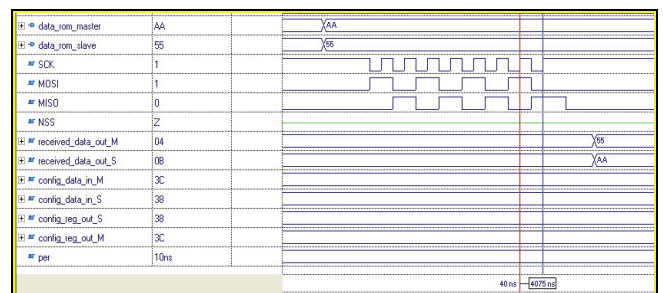
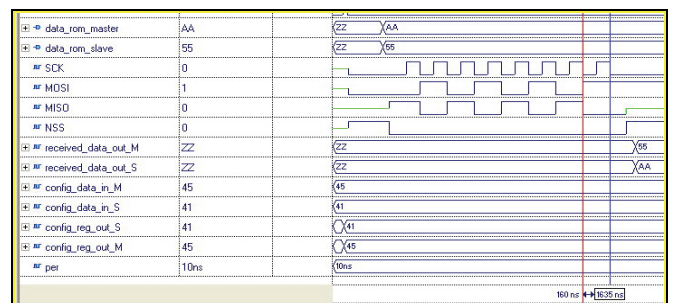
U cilju realizacije logičke simulacije formiran je *testbench* koji objedinjuje dve instance projektovanog SPI bloka. Jedna instanca predstavlja *master*, a druga *slave* uređaj. Za proveru ispravnosti postavljanja signala na SPI komunikacionim linijama, kreiran je sledeći scenario događaja:

- Reset signalom se oba komunikaciona modula dovode u inicijalno stanje. Svaki od modula poseduje svoj interni bafer memorijski prostor veličine 16B iz koga cikično čita podatak po podatak koji treba serijski proslediti putem MOSI/MISO linije,
- Nakon prijema, podatak se šalje na odgovarajući izlazni port i zadržava do prijema narednog podatka.

Na taj način *master* i *slave* uređaj razmenjuju podatke prethodno upisane u pomeračke registre. Ovim testom je pokriven samo slučaj komunikacije sa jednim *slave* uređajem pri istoj frekvenciji sistemskog takta za oba uređaja. Cilj ovakvog testa je da se proveru ispravnost rada hardvera tj. *data path* dela preprojektovanog kontrolera. Problem sinhronizacije nije pokriven ovim testom. Na sledećim slikama prikazana su dva dijagrama sa talasnim oblicima karakterističnih signala dobijenih nakon simulacije korišćenjem opisane test sekvence. Bitni signali su: *data_rom_master/slave* – podatak koji *master/slave* šalje, *received_data_out_M/S* – podatak koji *master/slave* prima, *config_reg_out_M/S* – stanje konfiguracionog registra *master/slave* uređaja, *per* – perioda takta oscilatora.

Sa slike 4. očigledno je da se podatak na MOSI/MISO liniju postavlja prednjom ivicom taktnog impulsa. U trenutku pojave zadnje ivice taktnog impulsa podatak je stabilan i može se prihvatiti sa MOSI/MISO linije (*cpha*=1). U ovom

slučaju prvi bit u povorci je MSB bit (*dord*=1). Kako *slave* select opcija nije dozvoljena (*ssen*=0), NSS linija ostaje u stanju visoke impedanse i može se eksterno vezati na odgovarajući logički nivo. Komplementaran mod rada je prikazan na slici 5.

Slika 4. Stanje SPI linija za *config_reg_master=0x3C* i *config_reg_slave=0x38*.Slika 5. Stanje SPI linija za *config_reg_master=0x45* i *config_reg_slave=0x41*.

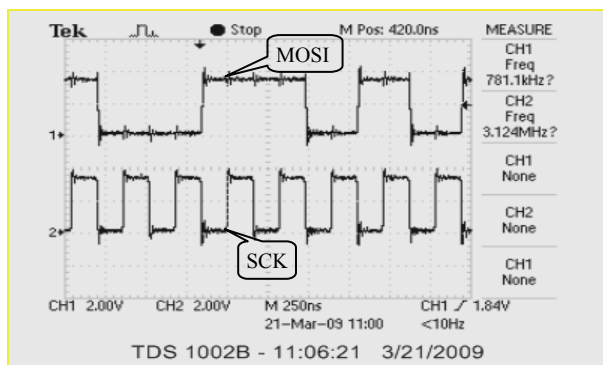
5. REALIZOVAN SPI KONTROLER NA FPGA ČIPU

Za realizaciju projektovanog SPI kontrolera iskorišćena je razvojna ploča Altera DE1 i Quartus IIv8.1 razvojno softversko okruženje za PC [7]. Genisani VHDL opis je sintetizovan i implementiran u EP2K20F484C7 Altera FPGA čipu serije CycloneII. Čip je proizveden u 90nm tehnologiji i ima 20K logičkih elekmenata.

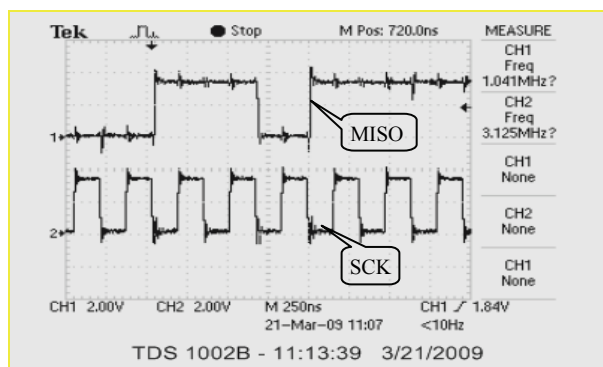
Za verifikaciju realizovanog SPI kontrolera ostvarena je SPI komunikacija između dve DE1 razvojne ploče. Na sledećim slikama je prikazan par karakterističnih talasnih oblika signala odgovarajućih SPI linija snimljenih osciloskopom. Za taktni oscilator frekvencije 50MHz odabrana frekvencija taktnog signala na SCK liniji je 3,125 MHz ($fosc/16$).

Na slici 6 prikazani su talasni oblici signalnih linija SPI kontrolera koji je konfigurisan kao *master*. Sadržaj njegovog konfiguracionog registra je 0x45, podatak koji on šalje je

0x59. Sa slike se može uočiti da se signal prihvata na prednju ivicu taktnog signala, a postavlja na zadnju ($cpha=0$). Prvi bit u povorci je LSB bit ($dord=0$), a frekvencija taktnog signala 3.127MHz ($\approx 3,125\text{MHz}$ tj. $spr1=0, spr2=1$). Na slici 7 su prikazani talani oblici SPI signalnih linija na strani *slave* uređaja koji ima istu konfiguraciju, a šalje podatak 0xEC.



Slika 6. Komunikacija master - slave.



Slika 7. Komunikacija slave-master.

6. ZAKLJUČAK

SPI interfejs predstavlja jedan od najrasprostranjenijih serijskih interfejsa. Odlikuje se jednostavnom hardverskom strukturom. Adresiranje se obezbeđuje selekcijom, a ne slanjem adresnog podatka. U komunikaciji više uređaja samo se jedan od njih može identifikovati kao *master* na jedinstvenim SPI linijama. Omogućava velike brzine prenosa podataka reda nekoliko Mbit/s i *full duplex* vezu. Ne postoji problem licenciranja kao što je to slučaj kod I2C interfejsa.

U većini slučajeva se koristi za komunikaciju između dva uređaja na malim rastojanjima (unutar čipa, ili između čipova na PCBu). Prilikom komunikacije sa više uređaja povećava se broj signalnih linija (kompleksnija struktura). Postoji problem sinhronizacije. Ne implementira mehanizam potvrde prijema. Ne postoji jednoznačna, zvanična definicija samog komunikacionog protokola što može da izazove probleme u realizaciji upravljačke logike SPI kontrolera.

Usvojivši scenario razmene podataka po SPI protokolu projektovano je i u radu prikazano rešenje SPI komunikacionog kontrolera. Generisan je VHDL opis koji je moguće sintetizovati. Funkcionalnost projektovanog hardvera je najpre verifikovana simulacijom. Zatim je i fizički proverena funkcionalnost projektovanog SPI kontrolera implementiranog na EP2C20F484C7 Altera FPGA čipu serije CycloneII.

ZAHVALNOST

Rezultati prikazani u ovom radu ostvareni su u okviru projekata TR11007 i TR11029 čiju realizaciju finansira Ministarstvo za nauku i tehnološki razvoj Republike Srbije.

LITERATURA

- [1] Mark Balch, "COMPLETE DIGITAL DESIGN", Copyright © 2003 by The McGraw-Hill Companies, Inc., 2003, pp. 119-120.
- [2] F2MC-16LX, 16-BIT MICROCONTROLLER, MB90340 Series HARDWARE MANUAL.pdf, July 2008 the third edition. pp. 443-504. http://mcu.emea.fujitsu.com/document/products_mcu/mb90340/documentation/hm90340e-cm44-10143-3e.pdf.
- [3] M68HC11 Reference Manula, M68HC11RM/D Rev. 6.1, pdf data sheet, Copyright Freescale Semiconductor, Inc. 2007, pp. 44-82, pp. 292-311. <http://www.enseirb.fr/~kadionik/68hc11/m68hc11rm.rev4.1.pdf>.
- [4] "Overview and Use of the SPI-PICmicro Serial Peripheral Interface", pdf tutorial, <http://ww1.microchip.com/downloads/en/devicedoc/spi.pdf>.
- [5] K.Pratyush Aditya, Vivek Rathi, Praneet Koppula, "Interfacing using Serial Protocols", <http://intranet.daiict.ac.in/~ranjan/esp2005/presentation/Interfacing%20using%20spi%20and%20i2c.pdf>.
- [6] TMS320x281x Serial Peripheral Interface, Reference Guide, Literature Number: SPRU059E June 2002–Revised February 2009, <http://focus.ti.com/lit/ug/spru059e/spru059e.pdf>.
- [7] Milun S. Jevtić, Bojan B. Jovanović, DIGITALNA ELEKTRONIKA – PRAKTIKUM ZA LABORATORIJSKE VEŽBE, Edicija: Pomoćni udžbenici, ISBN 978-86-85195-68-6, Elektronski fakultet, Niš 2008.
- [8] Vančo B. Litovski, PROJEKTOVANJE ELEKTRONSKIH KOLA: SIMULACIJA, OPTIMIZACIJA, TESTIRANJE, FIZIČKO PROJEKTOVANJE, I izdanje, DGIP "Nova Jugoslavija", Vranje, 2000.
- [9] Predrag P. Petković, Mr. Miljana Sokolović i Mr. Bojan Andjelković, PROJEKTOVANJE INTEGRISANIH KOLA – VHDL simulacija i sinteza, Elektronski fakultet u Nišu u saradnji sa Austrian Cooperation Eastern Europe WUS Austria, interna, publikacija, Niš, 2005.
- [10] M. Zwoliński, Digital System Desing with VHDL, First published 2000, Personal Education & Prentice Hall, Printed by Ashford Color Press Ltd., Gosport.

Abstract – One realization of communication controller which implement SPI communication protocol is presented in this paper. Controller can be configured as *master* or *slave* device. It is realized using VHDL language and implemented on Altera FPGA chip. Test sequence is created for functional verification. Presented waveforms prove correct functioning of designed entity.

SPI COMMUNICATION CONTROLLER FPGA IMPLEMENTATION

Dejan Mirković, Milunka Damnjanović, Milun Jevtić