

# Logička verifikacija i projektovanje kola za programiranje IP bloka mikrokontrolera

Borisav Jovanović, Dejan Mirković, Milunka Damjanović

**Apstrakt**—U ovom radu detaljno je objašnjen postupak verifikacije projekta, globalna šema i rad ugrađenih kola koja omogućavaju postupak programiranja IP bloka mikrokontrolera. IP blok koristi standardni 8051 skup instrukcija. Projektovani mikrokontroler deo je složenog integrisanog sistema na čipu. Rad mikrokontrolera proverava se kroz SPI interfejs različitim asemblerskim programima.

**Ključne reči**— 8051 mikrokontroler; programiranje mikrokontrolera; modovi rada

## I. UVOD

8051 mikrokontroleri nalaze primenu u mnogim savremenim elektronskim kolima i sistemima, gde se god zahteva neka obrada signala i kontrola procesa, tako da se u zavisnosti od specifikacije koristi odgovarajuća taktna frekvencija mikrokontrolera, skup perifernih jedinica i druge karakteristike.

Postoje različite implementacije 8051 mikrokontrolera. Veoma su česta rešenja u kojima su 8051 mikrokontroleri implementirani na jednom čipu. Takođe, 8051 mikrokontroler može se implementirati na FPGA čipovima ili kolima specifične namene. U našem slučaju projektovani IP blok 8051 mikrokontrolera je deo sistema na čipu. U ovom radu objašnjen je postupak verifikacije projekta, globalna šema i rad ugrađenih kola koja omogućavaju postupak programiranja mikrokontrolera.

## II. OPIS MIKROKONTROLERA, INTERFEJS, MODOVI PROGRAMIRANJA

### A. Opis mikrokontrolera - globalna šema

IP blok mikrokontrolera (MCU) je projektovan da izvršava standardni 8051 skup instrukcija, koji sadrži 255 različitih instrukcija. Zbog obimnog skupa instrukcija i postojanja 6 modova adresiranja, mikrokontroler 8051 može se svrstati u CISC mikroracunare (mikroracunare sa složenim skupom instrukcija). Ovaj skup instrukcija se često koristi i podržan je od strane mnogih alata za razvoj softvera [1, 2].

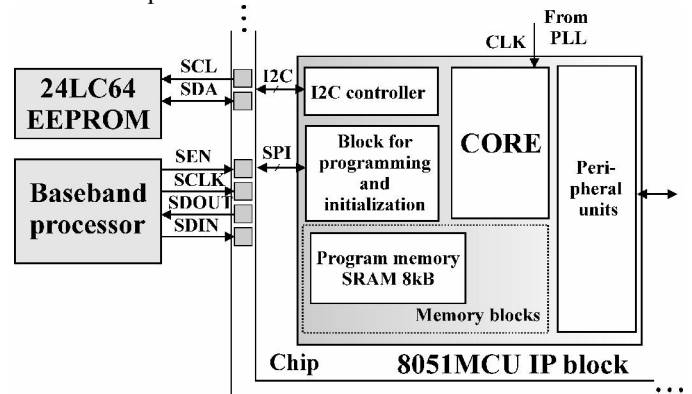
Glavna struktura MCU-a se sastoji od jezgra, memorijskih blokova, bloka za programiranje i inicijalizaciju i perifernih jedinica (Sl.1). Jezgro MCU-a pribavlja, dekodira i izvršava instrukcije. Perifernijske

jedinice se sastoje od tri paralelna digitalna ulazno/izlazna porta (P0, P1 i P2), nekoliko komunikacionih modula - jednog asinhronog univerzalnog prijemnika/predajnika (UART) i I2C komunikacionog kontrolera. Takođe, MCU sadrži tri kola tajmera/brojača.

Svi memorijski blokovi se fizički nalaze na čipu. MCU ne sadrži memoriju za trajno skladištenje programskog koda. Umesto toga, koristi SRAM memoriju na čipu, kapaciteta 8kB i eksterni serijski EEPROM 24LC64 čip (Sl.1). Svaki put nakon resetovanja mikrokontrolera, programska memorija se automatski učitava iz spoljašnjeg EEPROM-a u SRAM blok.

### B. Veza mikrokontrolera ka Baseband procesoru i I2C EEPROM memoriji

Mikrokontroler je povezan sa eksternim Baseband procesorom [3] preko Serial Protocol Interface (SPI) (Sl.1). Baseband processor upravlja radom mikrokontrolera preko skupa od ukupno sedam 8-bitnih mSPI\_REG registara koji su fizički implementirani u mikrokontroleru.



Sl. 1. Blok šema IP bloka mikrokontrolera.

SPI interfejs (Sl.1) sadrži 4 linije:

- SCLK – signal takta
- SDIN - ulazni signal za podatke
- SDOUT - izlazni signal za podatke
- SEN – signal dozvole za prenos podataka.

SCLK, SDIN i SEN predstavljaju ulazne signale mikrokontrolera; generisani su dakle, od strane Baseband procesora. SDOUT je izlazni signal mikrokontrolera.

U serijskom prenosu podataka preko SPI interfejsa mikrokontroler obavlja funkciju slejva, dok je Baseband master. Signal SCLK nije u sinhronizmu sa osnovnim taktom mikrokontrolera – CLK. Ovo predstavlja značajan izazov tokom projektovanja i logičke verifikacije. Projektovano je jedinstveno rešenje SPI kontrolera koji sadrži dodatna podkola za sinhronizaciju taktnih signala SCLK i CLK. Upotrebom ovog kola SPI komunikacija radi i na frekvenciji od 100MHz.

Borisav Jovanović – Elektronski fakultet u Nišu, Univerzitet u Nišu, Ul. Aleksandra Medvedeva 14, 18000 Niš, Srbija (e-mail: borisav.jovanovic@elfak.ni.ac.rs).

Dejan Mirković – Elektronski fakultet u Nišu, Univerzitet u Nišu, Ul. Aleksandra Medvedeva 14, 18000 Niš, Srbija (e-mail: dejan.mirkovic@elfak.ni.ac.rs).

Milunka Damjanović – Elektronski fakultet u Nišu, Univerzitet u Nišu, Ul. Aleksandra Medvedeva 14, 18000 Niš, Srbija (e-mail: milunka.damjanovic@elfak.ni.ac.rs).

Od ukupno sedam mSPI\_REG registara posebno su važni:

- mSPI\_REG2 - kontrolni registar preko koga *Baseband* upravlja radom mikrokontrolera
- mSPI\_REG3 - statusni registar preko kog *Baseband* dobija informacije o radu mikrokontrolera
- registar mSPI\_REG4, preko koga se prenose podaci iz *Baseband-a* u mikrokontroler

Posebno su važna dva bita registra mSPI\_REG2, koja su označena sa *Mode(1:0)*. Preko njih *Baseband* može da resetuje mikrokontroler ili da bira načine (modove) programiranja. Pored toga, *Baseband* procesor upisuje programski kod u mikrokontroler.

Mikrokontroler je povezan dalje, preko I2C serijske magistrale, sa eksternom EEPROM memorijom u kojoj se trajno pamti programski kod (Sl.1). Tokom svoga rada mikrokontroler može da upisuje podatke u EEPROM memoriju. To je važno, jer mikrokontroler ne poseduje u sebi neki vid trajne memorije za čuvanje rezultata. I2C magistrala se sastoji od dve linije:

- SDA - za podatke
- SCL - taktni signal

U I2C komunikaciji mikrokontroler obavlja funkciju mastera, dok je EEPROM memorija slejv. Brzina I2C komunikacije ograničena je na 100kHz.

### C. Modovi za programiranje mikrokontrolera

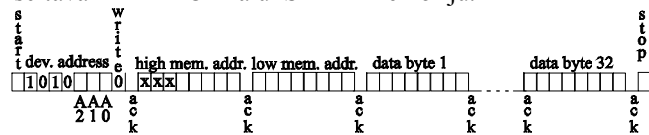
*Baseband* procesor resetuje MCU upisivanjem komande za reset u registar mSPI\_REG2 - *Mode(1:0)*="00". Nakon resetovanja sledi programiranje mikrokontrolera. Dve se opcije koriste za programiranje - označene su kao A i B.

#### Opcija A

Ne koristi EEPROM memoriju. Sastoji se iz sledećih operacija:

1. *Baseband* postavlja bitove *Mode(1:0)*="10".
2. *Baseband* programira 8kB programskog koda u 8kB SRAM mikrokontrolera preko SPI interfejsa
3. Kada se upisivanje programa završi, mikrokontroler započinje izvršavanje programa.

Programska memorija MCU-a kapaciteta je 8kB i implementirana je kao SRAM blok na čipu. SRAM gubi svoj sadržaj nakon nestanka napajanja. Da bi se to sprečilo, potrebno je da se program zapamti u spoljašnji EEPROM čip. Prilikom restartovanja mikrokontrolera, program se iščitava iz EEPROM-a u SRAM memoriju.



Sl. 2. Paket podataka za upisivanje u EEPROM memoriju.

#### Opcija B

Koristi EEPROM memoriju 24LC64 i sastoji se iz sledećih operacija:

1. *Baseband* postavlja bitove *MODE(1:0)*="01"
2. *Baseband* programira 8kB programa u SRAM preko SPI linija
3. Istovremeno dok prima programski kod,

mikrokontroler upisuje isti i u EEPROM 24LC64

4. Kada je prenos programa u SRAM i EEPROM završen, MCU započinje izvršavanje programa.

Programski kod u vidu paketa od 32 bajtova prenosi se preko I2C linija ka EEPROM memoriji. Slika 2 prikazuje strukturu jednog paketa.

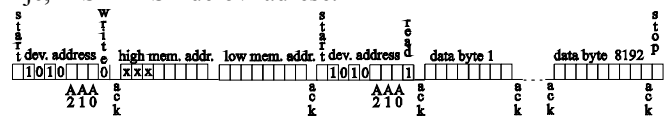
Paket počinje startnim bitom iza koga sledi adresa EEPROM memorije. Zatim slede MSB i LSB bajtovi adrese na kojoj se podaci smeštaju, niz od 32 bajta dela programskog koda i na kraju stop bit za kraj prenosa. Bajtovi su razdvojeni bitom potvrde. Vreme kašnjenja od 5ms je potrebno između uzastopnih paketa radi ispravnog upisivanja u EEPROM memoriju.

*Baseband* procesor proverava da li je proces programiranja gotov preko bita *Programmed* koji se nalazi u registru mSPI\_REG3. Kada je *Programmed*='1' postupak programiranja je završen.

Opcija B takođe podrazumeva sledeće:

1. *Baseband* postavlja bitove *MODE(1:0)*="11". Sadržaj EEPROM-a se čita i prebacuje u SRAM preko I2C-a.
2. Kada je prenos programskog koda u SRAM završen, MCU započinje izvršavanje programa.

Slika 3 prikazuje strukturu podataka koji se prenose preko I2C magistrale. Povorka bajtova ima dva dela. Prvi deo se koristi za upisivanje početne adrese EEPROM-a sa koje se podaci čitaju. Počinje startnim bitom, zatim sledi adresa EEPROM memorije i I2C komanda za upis, i nakon nje, MSB i LSB delovi adrese.



Sl. 3. Paket podataka za upisivanje u EEPROM memoriju.

Drugi deo sekvence počinje bitom ponovljenog starta, sledi adresa EEPROM memorije i komanda za čitanje. Nakon nje, mikrokontroler prima niz od 8192 bajtova. I u ovom slučaju, *Baseband* procesor dobija informaciju o završetku programiranja preko bita *Programmed* koji se nalazi u registru mSPI\_REG3.

## III. IMPLEMENTACIJA, SIMULACIJA I TESTIRANJE POSTUPKA PROGRAMIRANJA MIKROKONTROLERA

### A. Implementacija mikrokontrolera

MCU realizovan je u tehnologiji TSMC 65nm [4] i koristi napon napajanja od 1.2V. Prilikom implementacije korišćeni su Cadence [5] alati za projektovanje kola:

- *RTL Compiler* za sintezu i
- *SoC Encounter* za rameštaj i povezivanje veza.
- *NCSim* za logičku verifikaciju kola

Realizovani mikrokontroler ima sledeće prednosti u odnosu na osnovnu arhitekturu industrijskog 8051 mikrokontrolera:

- kolo je implementirano u TSMC 65nm tehnologiji
- nova protočna arhitektura jezgra mikrokontrolera 8051 ostvaruje brzinu od jedne jedno-bajtnje instrukcije koja se izvršava u dva taktna ciklusa
- kolo može da radi na frekvenciji od 60MHz

- Kolo je logički verifikovano nakon izrade lejauta.
- Korišćeno je više tehnika za smanjenje dinamičke potrošnje. Na frekvenciji od 60MHz potrošnja snage je svega 3mW.

### B. Logička verifikacija postupka programiranja mikrokontrolera

Postupci programiranja SRAM memorije preko SPI interfejsa, programiranja EEPROM memorije i kasnijeg iščitavanja njegovog sadržaja u SRAM obuhvaćeni su opsežnim logičkim proverama.

Postupak logičke verifikacije počinje pisanjem asemblerskog programa koji se kompajlira Keil [1] ili SDCC [2] C kompajlerima u odgovarajući .hex fajl. Nakon toga, posebni C program prevodi .hex fajl u VHDL kôd. VHDL program sa programskim kodom se ubacuje u poseban testbenč opis napisan u VHDL-u, koji proverava postupak programiranja mikrokontrolera. Mikrokontroler koji je predmet provere, je instanciran je u testbenču. Testbenč prolazi kroz sve modove za programiranje. U VHDL opisu testbenča je posebnim procesom simuliran rad SPI komunikacionog kontrolera *Baseband* procesora. Takođe, instanciran je i VHDL opis EEPROM memorije.

Tokom postupka logičke verifikacije, VHDL program sa programskim kodom se iščitava i bajt-po-bajt, prebacuje preko SPI porta u mikrokontroler (Sl.1). Relevantni rezultati logičke simulacije rada MCU-a dobijeni u obliku vremenskih dijagrama omogućavaju uvid u promenu stanja SFR registara kontrolera. Poređenjem dobijenih i očekivanih stanja verifikovan je rad mikrokontrolera.

Posebnu teškoću pri projektovanju mikrokontrolera i naročito, njegovog podsistema SPI komunikacije, predstavlja to što su takt *Baseband* procesora SCLK i takt mikrokontrolera CLK potpuno nezavisni. Projektovana su posebna kola koja sinhronizuju ova dva taktna signala. Simulacije koje proveravaju kola za sinhronizaciju razmatraju razne kombinacije frekvencija taktnih signala SCLK i CLK. Maksimalna taktna frekvencija mikrokontrolera je 60MHz. SPI komunikacija verifikovana je do vrednosti od 100MHz.

Kontrolna logika realizovana testbenčom, koja simulira rad *Baseband* procesora, upravlja postupkom slanja bajtova programskog koda. Bajtovi se šalju preko SPI interfejsa, a zatim, preko I2C magistrale do EEPROM memorije.

Tokom simulacije opcije B, kada se programira EEPROM čip, bitovi mSPI\_REG2 registra se postavljaju u stanje Mode(1:0)="01". Posle reseta, bajtovi se serijski šalju u paketima od po 32 bajtova u EEPROM, na način koje je već opisan. Nakon uspešnog programiranja, mikrokontroler automatski postavlja signal *Programmed* na nivo logičke "1". Kontrolna logika testbenča, koja simulira rad *Baseband* procesora, čita sadržaj statusnog bita *Programmed*.

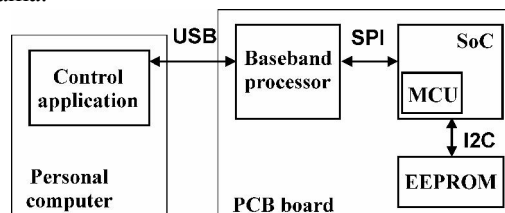
Inicijalizacija programske SRAM memorije iz EEPROM-a počinje takođe nakon resetovanja čipa, kada se postavlja Mode(1:0)="11". Podaci se iščitavaju iz EEPROM-a i smeštaju u 8kB SRAM. Kontrolu prenosa podataka iz jedne memorije u drugu obavlja logika realizovana u testbenču, koja simulira rad *Baseband* procesora. Čitanje EEPROM-a traje sve dok brojač koji prosleđuje adresu ne postigne vrednost adrese zadnje lokacije SRAM-a. Takodje, proces

iščitavanja se završava kada je *Programmed*="1".

### C. Testiranje modova za programiranje čipa

Globalna šema sistema za testiranje mikrokontrolera prikazana je na slici 4. Sastoji se kontrolne aplikacije koja je instalirana na personalnom računaru i štampane ploče koja je projektovana za potrebe testiranja čipa.

Veza između računara i ploče je USB. Pored čipa koji je predmet provere, ploča sadrži *Baseband* procesor i EEPROM memoriju koja se koristi za memorisanje programa.



Sl. 4. Globalna šema sistema za testiranje mikrokontrolera

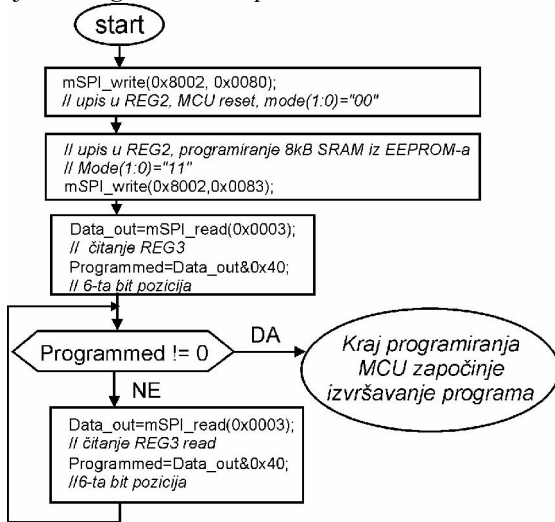
Kontrolna aplikacija se koristi za kontrolu parametara svih podsistema integrisanog kola u kome je ugrađen IP blok MCU 8051. Poseban deo kontrolne aplikacije odnosi se na rad sa MCU 8051 i preko njega moguće je:

- resetovati 8051 mikrokontroler
- uneti (loadovati) .hex fajl kojim se mikrokontroler programira
- odabrati mod programiranja i po njemu isprogramirati mikrokontroler
- testirati rad mikrokontrolera preko posebnog moda mikrokontrolera
- proveriti sadržaj interne RAM memorije mikrokontrolera i svih SFR registara, što je od značaja prilikom verifikacije rada MCU 8051 i provere asemblerskih programa.

Kontrolna aplikacija preko *Baseband* čipa upravlja radom mikrokontrolera. Razvijena je u Code Blocks [6] *open-source* razvojnom okruženju korišćenjem GCC [7] C kompajlera. U firmveru *Baseband* čipa ugrađene su osnovne funkcije za upis i čitanje registara mSPI\_REG. Ove funkcije direktno koriste SPI magistralu za prenos podataka između *Baseband* procesora i mikrokontrolera. Pojedini delovi kontrolne aplikacije koriste ove osnovne funkcije da bi sve gore navedene funkcije bile omogućene. Nadalje, opisani su algoritmi koji su implementirani u kontrolnoj aplikaciji a koriste za programiranje mikrokontrolera.

Sledi opis algoritma koji se koristi za upisivanje programskog koda u mikrokontroler. Na početku, mikrokontroler se resetuje upisom komande za reset u registar mSPI\_REG2. Dva LSB bita registra mSPI\_REG2 (bitovi Mode1 i Mode0) se resetuju. Nakon toga, u ova dva bita upisuju se nove vrednosti - Mode(1:0)="01" ili "10", u zavisnosti od izbora moda za programiranje. Tokom moda 1, kod programa se snima u SRAM i EEPROM. Tokom moda 2, samo se SRAM inicijalizuje novim programskim kodom. Kod programa se šalje u paketima koji se sastoje od 32 bajtova. Signal mikrokontrolera *EmptyFIFO* sadrži informaciju da li je FIFO memorija prazna (deo bloka za Programiranje i inicijalizaciju koji je prikazan na Sl.1). tj. da li je mikrokontroler spreman da primi nove podatke iz *Baseband*-a. *EmptyFIFO* može da se proveriti čitanjem

statusnog registra mSPI\_REG3 i pronalaženjem vrednosti na LSB bit poziciji. Ako EmptyFIFO='1', podaci mogu da se šalju iz *Baseband*-a, tako što se 32 bajtova programa upisuju jedan za drugim u registar mSPI\_REG4. Upisivanje paketa se ponavlja 256 puta za svih 8192 bajtova programa. Na kraju se *Programmed* bit proverava.



Sl. 5. Postupak programiranja mikrokontrolera u modu 3

Slika 5 prikazuje algoritam koji se koristi za inicijalizaciju programskog koda, kada se on čita iz spoljašnjeg EEPROM čipa. Na početku algoritma mikrokontroler se resetuje upisom komande za reset u registar mSPI\_REG2. Nakon toga, u bitove registra Mode1 i Mode0 upisuju se nove vrednosti - Mode (1:0)="11". Na kraju, proverava se *Programmed* bit.

#### IV. ZAKLJUČAK

IP blok 8051 mikrokontrolera predstavlja deo složenog sistema na čipu i uspešno je verifikovan nakon opsežnih simulacija. Sve provere potvrđene su simulacijama nakon izrade lejauta. Posebno su verifikovani delovi koda koji se koriste za programiranje mikrokontrolera.

Mikrokontroler koristi nekoliko modova za programiranje. Jedan od njih direktno inicijalizuje programsku SRAM memoriju mikrokontrolera kodom koji se prenosi preko *Baseband* procesora; druga opcija za

programiranje smešta programski kod istovremeno u SRAM memoriju i spoljašnju I2C EEPROM memoriju, gde se kod trajno čuva.

U serijskom prenosu podataka preko SPI interfejsa mikrokontroler obavlja funkciju slejva, dok je Baseband master. Takni signal SPI komunikacije nije u sinhronizmu sa osnovnim taktom mikrokontrolera. Projektovano rešenje SPI kontrolera efikasno prevazilazi ovaj problem.

#### ZAHVALNICA

Ovaj rad je finansiran od strane Ministarstva prosvete, nauke i tehnološkog razvoja u okviru projekta br TR 32004, pod nazivom : "Napredne tehnologije za merenje, kontrolu i komunikacije na električnoj mreži".

#### LITERATURA

- [1] KEIL C compiler and development tool for 8051 microcontrollers, <http://www.keil.com/c51/devproc.asp>
- [2] Open source SDCC C compiler for 8051 microcontrollers <http://sdcc.sourceforge.net/>
- [3] AT90USB82/162, <http://www.atmel.com/Images/7707s.pdf>
- [4] TSMC 65nm standard cell library, [http://www.europractice.com/technologies\\_TSMC.php?tech\\_id=65nm](http://www.europractice.com/technologies_TSMC.php?tech_id=65nm)
- [5] Cadence, EDA software and verification tools <http://www.cadence.com/us/pages/default.aspx>
- [6] Code::Blocks IDE, <http://www.codeblocks.org/>
- [7] GCC, the GNU Compiler Collection <http://gcc.gnu.org>

#### ABSTRACT

In this paper the procedures for the project verification and global scheme of implemented circuits which allow the programming of microcontroller IP block are given. The IP block uses the standard 8051 instruction set. The designed microcontroller is incorporated into complex integrated system-on-chip. The microcontroller operation is verified through SPI interface using different assembler program codes.

#### The design and logical verification of circuits implementing the programming modes of a microcontroller IP block

Borisav Jovanović, Dejan Mirković, Milunka Damnjanović