

## JEDNO REŠENJE IMPLEMENTACIJE RUKOVAOCA POBOLJŠANOM NAND-FLASH MEMORIJOM NA PUC 3030A MIKROKONTROLERU

Siniša Grgić, Zvonimir Kaprocki, Željko Lukač, Dragan Simić, MicronasNIT, Novi Sad

**Sadržaj** – Prikazana je implementacija rukovaoca poboljšane NAND-flash memorije na PUC 3030A mikrokontroleru. Cilj je implementirati rukovalac sa što većim brojem mogućnosti uz minimalno korišćenje resursa.

### 1. UVOD

U ovom radu je predstavljeno jedno rešenje rukovaoca poboljšanom NAND-flash memorijom. Rukovalac podržava sledeće: formatiranje na najnižem nivou, podršku za standardnu i poboljšanu NAND-flash memoriju, podršku za rukovanje sa više od dva uređaja NAND-flash memorije, podršku za NAND-flash memoriju koja ima više od 4096 blokova, podršku za NAND-flash memoriju veličine 32, 64, 128, 256, 512 MB i više (samo 8-bitni uređaji). Rukovalac radi sa USB(Universal Serial Bus) i FAT(File Allocation Table) bibliotekama, upis podataka preko USB veze je ključni pokazatelj efikasnosti, ciljna brzina pisanja je 5Mb/s. Rukovalac je tako projektovan da obezbeđuje integritet podataka i nakon gubitka napajanja platforme odnosno ciljnog uređaja.

Rukovalac NAND-flash memorijom je implementiran na PUC(Programmable Universal Controller) 3030A programabilni univerzalni kontroler koji je zasnovan na ARM7TDMI procesoru. PUC 3030A zadovoljava sve zahteve za digitalne audio plejere, diktafone, specifične prenosive mrežne uređaje i bežične aplikacije kao što je Bluetooth. Osnovne karakteristike mikrokontrolera su: takt procesora je 64MHz, 56kB ugrađene SRAM memorije, 256kB ugrađene flash memorije, poseduje određene sigurnosne jedinice i USB vezu.

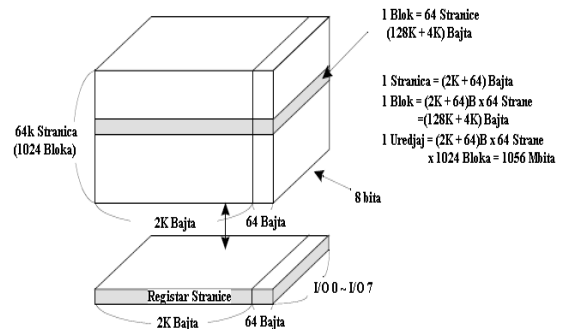
### 2. POBOLJŠANA NAND-FLASH MEMORIJA

U prošlosti NOR-flash memorija je bila najčešće upotrebljavan medij za skladištenje izvršnog koda za startovanje sistema i BIOS programa. Zbog relativno kratkog vremena pristupa i mogućnosti linearnog adresiranja u okviru memorijskog prostora platforme otvorila se mogućnost upotrebe NOR-flash memorije za skladištenje veće količine izvršnog koda. NAND-flash memorija, koja ima veće vreme pristupa i nelinearno okruženje, obično nije upotrebljavana za skladištenje izvršnog koda već je predstavljana kao medij za skladištenje podataka za aplikacije. Njen veći kapacitet i mnogo bolje karakteristike pri upisu podataka je čine pogodnom za platforme koje zahtevaju skladištenje velike količine podataka.

Zbog težnje da se smanje troškovi proizvodnje računara ova tradicionalna podela flash memorije je preispitana. Pošto je cena NAND-flash memorije skoro tri puta manja od cene NOR-flash memorije, razvijene su nove metode za upotrebu NAND-flash memorije kako za skladištenje izvršnog koda tako i za skladištenje podataka.

Poboljšana NAND-flash memorija smešta podatke na memorijske oblasti zvane stranice. Tipična stranica ima 2048 bajta za smeštanje podataka (data area) i 64 bajta za smeštanje dodatnih informacija (spare area) potrebnih za rukovalac kao što su CRC, oznaka neispravnog bloka, ECC i tome slično. Ove stranice su organizovane u grupe koje se

nazivaju blokovi. Primer ove organizacije stranica i blokova je prikazan na slici 1. za uređaj SAMSUNG K9F1G08X0M. Povećanje kapaciteta NAND-flash memorije se postiže promenom veličine stranice kao i promenom broja blokova po jednom integrisanom kolu NAND-flash memorije.[3]



Slika 1. Organizacija memorije 8-bitnog uređaja SAMSUNG K9F1G08X0M

### 3. METODOLOGIJA (PROBLEMI I REALIZACIJA)

Osnovni zadatak rukovaoca NAND-flash memorijom je da obezbedi programsku podršku za rad sa memorijom u kojoj korisnik ne mora da vodi računa o veličini bloka, brisanju bloka, označavanju neispravnih blokova, broja uređaja NAND-flash memorije na ciljnoj platformi, itd. Rukovalac podržava sledeće:

- Formatiranje na najnižem nivou

Pre prve upotrebe poboljšane NAND-flash memorije potrebno je uraditi formatiranje na najnižem nivou. Formatiranje na najnižem nivou podrazumeva formiranje memorijske mape (koja služi za preslikavanje logičkih adresa u fizičke), proveru ispravnosti svih blokova i upis identifikacionog koda. Polje fizičkih adresa se odnosi na fizičke adrese blokova flash memorije. Polje logičkih adresa predstavlja neprekidni linearni adresni prostor i omogućava zamenu fizičkih blokova bez potrebe premeštanja podataka. Svaka logička adresa se odnosi na jednu fizičku adresu flash memorije. Ispravnost bloka se detektuje proverom polja dodatnih informacija datog bloka. Identifikacioni kod predstavlja posebnu oznaku koja se upisuje u nulti fizički blok flash memorije kao potvrda ispravno urađenog formatiranja na najnižem nivou.

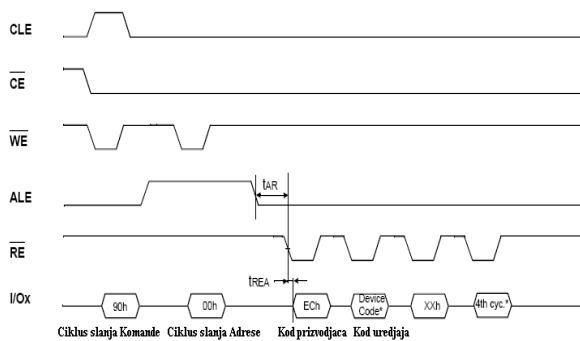
Proces formatiranja na najnižem nivou započinje čitanjem nultog fizičkog bloka i proverom identifikacionog koda. Samo u slučaju da identifikacioni kod nije upisan, moguće je nastaviti formatiranje i popunjavanje memorijske mape. Sledeća faza je provera ispravnosti svih blokova i označavanje u memorijskoj mapi neispravnih blokova. Nulti fizički blok se označava kao specijalan, jer služi za upis identifikacionog koda. Nakon što su svi ispravni blokovi markirani, u memorijsku mapu se upisuju adrese ispravnih blokova i to u odnosu 1:1 što znači da svakoj slobodnoj

logičkoj adresi odgovara ista po apsolutnoj vrednosti fizička adresa. Na ovaj način je memorijska mapa popunjena. Sledi njeno sortiranje tako da svi neispravni blokovi budu postavljeni na kraj memorijske mape. Takođe se na kraju memorijske mape rezervišu prostor potreban za čuvanje same memorijske mape i određeni broj blokova koji služe kao rezerva u slučaju pojave novih neispravnih blokova. Ovi takozvani rezervni blokovi se ne koriste za upis i čitanje podataka pri normalnom radu uređaja niti se vide u ukupnom memorijskom prostoru, jedina njihova uloga je da zamene neispravne blokove kada se oni pojave. To znači da rukovalac rezervišu  $n$  rezervnih blokova. Pri pojavi novog neispravnog bloka jedan rezervni blok se zamenjuje neispravnim, broj rezervnih blokova pada na  $n-1$  a memorijski prostor koji korisnik vidi ostaje nepromenjen. Samo kada se potroši svih  $n$  rezervnih blokova, pri pojavi novog neispravnog bloka će doći do promene veličine memorijskog prostora kojeg vidi korisnik. Ovim postupkom je memorijska mapa spremna za snimanje u blokove koji su za nju predviđeni, a završni korak formatiranja na najnižem nivou je upis identifikacionog koda u nulti fizički blok.

Svaki put kada se ciljna platforma sa ugrađenom poboljšanom NAND-flash memorijom resetuje rukovalac proverava nulti fizički blok i njegov sadržaj. U slučaju da nulti fizički blok poseduje identifikacioni kod, proces formatiranja se preskače.

- Podršku za standardnu i poboljšanu NAND-flash memoriju

Rukovalac podržava rad sa standardnom i poboljšanom NAND-flash memorijom. Pri startovanju uređaja rukovalac pokreće komandu za čitanje koda prepoznavanja (koji nije isto što i identifikacioni kod u nultom fizičkom bloku). Ova komanda je ista za sve vrste NAND-flash memorije (standardne i poboljšane) i prikazana je na slici 2.



Slika 2. Komanda za čitanje koda prepoznavanja NAND-flash memorije

Sekvenca slanja komande je sledeća: nakon otvaranja porta za slanje komande (CLE), šalje se komanda 90h, sledi otvaranje porta za slanje adrese (ALE) i šalje se adresa 00h, zatim se otvara port za čitanje (RE) i preuzimaju podaci koji stižu kao odgovor na poslatu komandu i to kod proizvođača (Maker code) i kod prepoznavanja uređaja (Device code). Na osnovu koda za prepoznavanje uređaja se utvrđuje da li se radi o standardnoj ili poboljšanoj NAND-flash memoriji i koji je tip, a na osnovu toga se formiraju podaci potrebni za rad sa istom kao što su: broj blokova, veličina stranice, veličina prostora za dodatne informacije itd.

- Podršku za rukovanje sa više od dva uređaja NAND-flash memorije

Rukovaoc podržava rad sa više uređaja NAND-flash memorije na jednoj ciljnoj platformi. U toku razvoja rukovaoca na raspolaganju je bila platforma sa dva uređaja NAND-flash memorije a maksimum koji rukovalac treba da podrži je osam uređaja. Rad sa više uređaja NAND-flash memorije se postiže formiranjem memorijskih mapa za svaki uređaj posebno, ali tako da se celokupan memorijski prostor predstavi kao jedinstven i linearan. To znači da se formatiranje na najnižem nivou obavlja za svaku NAND-flash memoriju posebno s tim da logičke adrese svake sledeće (npr. druge) memorije počinju od poslednje vrednosti logičke adrese svake prethodne (npr. prve), a ne od nule, i takva procedura se ponavlja za svaku narednu memoriju.

- Podršku za memoriju koja ima više od 4096 blokova

Kod implementacije rukovaoca NAND-flash memorijom ograničavajući faktor je bio RAM memorija. Da bi se obezbedio brži rad potrebno je memorijsku mapu stalno držati u RAM memoriji. Zbog potreba samog operativnog sistema za memorijsku mapu nije bilo moguće obezbediti više od 4096 bajta RAM memorije. Pošto se svaka adresa predstavlja pomoću dva bajta sledi da sistem nije mogao da obezbedi prostor za memorijsku mapu veću od 2048 blokova. Problem je rešen tako što rukovalac svaku NAND-flash memoriju koja ima više od 2048 blokova (4096, 8192...) interpretira kao NAND-flash memoriju koja ima  $n * 2048$  blokova predstavljajući jedan fizički uređaj kao  $n$  virtuelnih. U ovom slučaju proces formatiranja je isti kao u slučaju da sistem ima više uređaja NAND-flash memorije, samo što se svaki uređaj predstavlja pomoću  $n$  virtuelnih uređaja. To znači da ako sistem ima  $a$  fizičkih uređaja a oni se dele na  $b$  logičkih uređaja onda se formatiranje radi za  $a*b$  uređaja.

- Podršku za NAND-flash memoriju veličine 32, 64, 128, 256, 512 MB i više (samo 8-bitni uređaji)

Gore pomenute metode podrške za uređaje koji imaju različite veličine stranica, različit broj blokova omogućava veliku fleksibilnost u radu rukovaoca. Ako uređaj ima mali broj blokova zahtevi za RAM memorijom su mali a samim tim je njegov rad brži, u suprotnom ako uređaj ima veći kapacitet (veći broj blokova) javlja se potreba za njegovim deljenjem na virtuelne pa se performanse sistema smanjuju, ali u dozvoljenim granicama.

- Rad sa USB bibliotekom, pristup po fizičkoj adresi, rukovanje neispravnim blokovima

Rukovalac podržava rad sa USB bibliotekom preko rutina za čitanje i pisanje zasnovanim na pristupu po logičkim adresama. Ove rutine su vidljive za USB biblioteku, a jedino ograničenje je bilo da se čita i piše po blokovima podataka veličine 512 bajta. Ovo je zahtevalo izmene kod rutina čitanja i pisanja kod uređaja koji imaju stranice veličine 2048 bajta. Ovaj problem je rešen deljenjem stranice na virtuelne stranice od po 512 bajta tako da se podaci u jednu stranicu od 2048 bajta u stvari čitaju/upisuju kao  $4 * 512$  bajta. Isti princip je urađen i za prostor dodatnih informacija kod kojih se 64 bajta deli na  $4*16$  bajta i na tom nivou se upisuje i čita.

Jedan od primarnih ciljeva pri projektovanju rutina za čitanje i upis je da brzina sekvencijalnog pisanja preko USB veze mora da bude 5Mb/s. Ovaj maksimum performansi je postignut pisanjem rutina za čitanje i pisanje na asemblersu.

Rukovalac podržava pristup po fizičkim adresama. Rutine za pristup su pisane po specifikacijama različitih tipova NAND-falsh memorije kao što je pokazan primer čitanja koda prepoznavanja na slici 2. Čitanje po logičkim adresama se u stvari svodi na prevođenje logičke adrese u fizičku pomoću memorijske mape, a zatim se memoriji pristupa po fizičkoj adresi.

Prvi bajt polja dodatnih informacija nosi oznaku neispravnog bloka. Ako se u toku rada desi da je neki blok neispravan u polje za dodatne informacije tog bloka na poziciju prvog bajta se upisuje oznaka neispravnog bloka. Nakon toga se neispravan blok zamenjuje sa ispravnim blokom. Kada se radi formatiranje na najnižem nivou upravo ovo polje se proverava kako bi se i u memorijskoj mapi označio neispravan blok.

- Integritet podataka posle gubitka napajanja

Jedan od najvažnijih zahteva pri projektovanju rukovaoca je njegova pouzdanost rada pri nestanku napajanja uređaja. U ovom slučaju jako je bitno sačuvati podatke koji su se na memoriji nalazili pre nestanka napajanja. Ova situacija podrazumeva realno stanje sistema. U realnim uslovima kod prenosivih uređaja koji imaju NAND-flash memoriju može da dodje do nestanka napajanja usled smanjenja kapaciteta baterija. Ako nestane napajanje potrebno je sačuvati podatke koji su se prethodno nalazili na memoriji a eventualno i podatke koji su do tog momenta bili upisani na NAND-flash memoriju. Rukovalac podržava ovu opciju što se postiže na sledeći način: svaki put kada se uređaj startuje rekonstruiše se mapa na osnovu podataka koji su upisani u deo memorije za dodatne informacije, rekonstruisana mapa se poredi sa mapom koja se nalazi na memoriji, ako se utvrdi da su različite upotrebljava se nova rekonstruisana mapa koja garantuje da su svi do tada upisani podaci sačuvani.

#### 4. ISPITIVANJE I REZULTATI

Za ispitivanje rukovaoca su korišćene razvojne platforme sa PUC 3030A kontrolerom i NAND-flash memorijom. Sva ispitivanja su urađena za standardnu i poboljšanu NAND-flash memoriju čime je pokazano da rukovalac može da radi sa obe vrste memorije. Sva ispitivanja čitanja i pisanja su urađeni preko USB veze u Windows okruženju.

Formatiranje na najnižem nivou se ispituje tako što se briše nulti fizički blok datog uređaja a potom se uređaj restartuje. Posle restartovanja rukovalac otkriva nedostatak identifikacionog koda uređaja i pokreće proceduru formatiranja niskog nivoa, posle čega je NAND-flash memorija spremna za korišćenje. Za ovu proceduru je napisana aplikacija koja pristupa nultom fizičkom bloku a Windows okruženje i USB veza služi za proveru ispravnosti uređaja posle formatiranja.

Pristup po fizičkoj adresi je ispitivan na sledeći način: velika količina podataka se upisuje na slučajno odabrane fizičke adrese koristeći komandu pisanja sa pristupom po fizičkoj adresi, ti podaci se zatim čitaju sa istih adresa koristeći komandu čitanja sa pristupom po fizičkoj adresi a zatim se proverava ispravnost pročitanih sadržaja. Ukoliko su komande sa pristupom po fizičkoj adresi ispravno implementirane, kao što je ovde slučaj, pročitani podaci su ispravni odnosno jednaki onima koji su upisani.

Za testiranje sekvencijalnog i slučajnog čitanja/pisanja, kritične testove (pokretanje i zaustavljanje uređaja nekoliko puta) i višeblokovsko čitanje/pisanje je napisana aplikacija u programskom jeziku C. Ova aplikacija ujedno meri brzinu USB transfera. Sekvencijalno čitanje/pisanje se ispituje na isti način kao i ispitivanje pristupa po fizičkoj adresi. Upisuju se podaci u memoriju, zatim se čitaju a onda se proverava ispravnost. Kritični ispit podrazumeva sledeću sekvencu: startovanje uređaja, jedno čitanje, jedan upis, jedno čitanje, zaustavljanje uređaja. Ova sekvencija se ponavlja 30 puta. Ispit za višeblokovsko čitanje/pisanje služi za proveru ispravnosti implementacije komande za datu mogućnost. Gore pomenuta aplikacija, kao što je napomenuto, meri USB transfer merenjem vremena potrebnog da se 1MB podataka

upíše/pročita na NAND-flash memoriju koristeći različite načine pisanja/čitanja (slučajni ili sekvencijalni pristup) i različite veličine paketa (512 – 64K bajta). Tabela 1 prikazuje rezultate merenja USB transfera za NAND-flash memoriju tipa TOSHIBA TC58V64AFT. Tabela 2 prikazuje rezultate merenja koji su rađeni za poboljšanu NAND-flash memoriju tipa SAMSUNG K9K2G08Q0M.

Tabela 1. Rezultati merenja USB transfera za TOSHIBA TC58V64AFT NAND-flash memoriju [Mb/s]

<i>Paket</i>	<i>512</i>	<i>1024</i>	<i>2048</i>	<i>4096</i>	<i>8192</i>	<i>16k</i>	<i>32k</i>	<i>64k</i>
<b>Sekv. pis.</b>	0.95	1.54	2.50	3.47	4.17	4.63	4.86	5.00
<b>Sluč. pis.</b>	0.26	0.52	1.04	2.02	4.17	4.63	4.88	5.00
<b>Sekv. čit.</b>	0.98	1.57	2.24	3.14	3.91	4.31	4.55	4.72
<b>Sluč. čit.</b>	0.98	1.57	2.24	3.14	3.91	4.31	4.55	4.72

Tabela 2. Rezultati merenja USB transfera za SAMSUNG K9K2G08Q0M poboljšanu NAND-flash memoriju [Mb/s]

<i>Paket</i>	<i>512</i>	<i>1024</i>	<i>2048</i>	<i>4096</i>	<i>8192</i>	<i>16k</i>	<i>32k</i>	<i>64k</i>
<b>Sekv. pis.</b>	0.97	1.57	2.59	3.85	4.46	5.02	5.30	5.48
<b>Sluč. pis..</b>	0.03	0.05	0.11	0.21	0.41	0.83	1.44	2.78
<b>Sekv. čit.</b>	0.98	1.57	2.24	3.14	3.91	4.31	4.57	4.72
<b>Sluč. čit.</b>	0.98	1.57	2.24	3.13	3.91	4.32	4.55	4.72

Rukovanje neispravnim blokovima je ispitano koristeći aplikaciju koja upisuje podatke u n-ti blok i namerno vraća informaciju o neuspešnom upisu. Na ovaj način rukovalac dobija informaciju da je dati blok neispravan i pokreće mehanizam rukovanja neispravnim blokovima koji zamenjuje neispravni blok ispravnim i ažurira memorijsku mapu. Ispitivanje je urađeno preko USB veze i PC aplikacije koja omogućava uvid u stanje memorijske mape čime je verifikovana ispravna promena memorijske mape posle pojave novog neispravnog bloka.

Integritet podataka i posle gubitka napajanja ili prekida USB veze se testira NAND-flash memorijom u koju je već upisana velika količinom podataka. Ispituje se sledećom sekvencom: pokrene se upis podataka u NAND-flash memoriju preko USB veze, u toku prenosa se prekine USB veza ili se prekine napajanje uređaja a zatim se uspostavlja USB veza ili se uključuje napajanje uređaja. Ispiti su pokazali da su i posle neočekivanog gubitka USB veze ili napajanja uređaja svi podaci koji su se ranije nalazili na NAND-flash memoriji sačuvani kao i podaci koji su do trenutka prekida bili upisani u memoriju.

Svi do sad pomenuti ispiti su rađeni na platformi koja ima dva integrisana kola poboljšane ili standardne NAND-

flash memorije čime je ujedno ispitana funkcionalnost rukovaoca pri radu sa dva uređaja.

#### 4. ZAKLJUČAK

U ovom radu je opisano jedno rešenje implementacije rukovaoca poboljšanom NAND-flash memorijom na PUC 3030A mikrokontroleru. Kao što je pokazano svi ispiti koji su urađeni potvrđuju funkcionalnost implementiranog rukovaoca. Rukovalac je pisan na programskom jeziku C sa izuzetkom rutina za pisanje i čitanje na najnižem nivou koje su pisane na assembleru radi ubrzanja rada uređaja. Pisanjem ovih rutina na assembleru je postignuta brzina sekvencijalnog pisanja preko USB veze od 5Mb/s što je i bio jedan od prioriteta pri projektovanju prikazanog rukovaoca. Postignut je i integritet podataka posle gubitka napajanja što je takođe jedan najvažnijih zahteva koje svaki rukovalac memorijom treba da podrži.

#### LITERATURA

- [1] Intel Corporation, *Understanding the Flash Translation Layer (FTL) Specification*, 1998. <http://developer.intel.com/design/comp/applnots/297816.htm>
- [2] A. Kawaguchi, S. Nishioka, and H. Motoda, *A Flash Memory based File System*, Proceedings of the USENIX Technical Conference, 1995.

- [3] Samsung Electronics Company, *1Gb 1.8V NAND Flash Errata Datasheet*, March 2003.
- [4] Samsung Electronics Company, *Advance FLASH MEMORY Datasheet*, April 2003.
- [5] Toshiba Corporation, *TC58V64AFT Datasheet*, September 2000.
- [6] David Woodhouse, *JFFS: Journalling Flash File System*, <http://sources.redhat.com/jffs2/jffs2-html/jffs2-html.html>
- [7] Li-Pin Chang, Tei-Wei Kuo, *An Adaptive Striping Architecture for Flash Memory Storage Systems of Embedded Systems*, Department of Computer Science and Information Engineering National Taiwan University, Taipei, Taiwan, 2002.

**Abstract** – The article deals with the implementation of advanced NAND flash driver on PUC 3030A microcontroller. Main goal is to implement driver with many possibilities using minimal resources.

**ONE SOLUTION FOR ADVANCED NAND FLASH DRIVER ON PUC 3030A MICROCONTROLLER**  
Siniša Grgić, Zvonimir Kaprocki, Željko Lukač, Dragan Simić