Šolski center Novo mesto  
Srednja elektro šola in tehniška gimnazija  
Šegova ulica 112  
8000 Novo mesto

**NASLOV V ZBIRNIKU**(Seminarska naloga)

Predmet: Računalništvo  
  
Avtor: Andraž Svete, T2A  
Mentor: Albert Zorko, univ. dipl. inž. el.  
Mentor: Gregor Mede, univ. dipl. inž. rač. in inf.

Novo mesto, december 2019

**POVZETEK IN KLJUČNE BESEDE**

V tej seminarski nalogi bom predstavil Motorolo 6803 in postopek ustvarjanja programa v programu Simulator MC6803 po nalogi, ki nam je bila dodeljena.

Naloga se začne s teoretičnim delo, konča pa s postopkom realiziranja programa.   
Postopek in teoretični del sta opremljena tudi s slikami.

Ključne besede:

* Motorola
* Simulator
* Ukazi
* Program
* Postopek
* Programiranje
* Naloga

**KAZALA**

**KAZALO VSEBINE**

[1. UVOD 1](#_Toc27946690)

[2. MOTOROLA 6803 2](#_Toc27946691)

[2.1. LIHA PARITETA 3](#_Toc27946692)

[2.2. NALOGA 4](#_Toc27946693)

[2.3. POSTOPEK 4](#_Toc27946694)

[3. ZAKJUČEK 6](#_Toc27946695)

**KAZALO SLIK**

[Slika 1: Skica mikrokrmilnika Motorola 6803 2](file:///D:\Dokumenti\ŠOLA\SREDNJA%20ŠOLA\Informatika%20in%20računalništvo\Seminarska%20naloga%20-%20Naslov%20v%20zbirniku\SEM.docx#_Toc27946883)

[Slika 2: Začetek 4](file:///D:\Dokumenti\ŠOLA\SREDNJA%20ŠOLA\Informatika%20in%20računalništvo\Seminarska%20naloga%20-%20Naslov%20v%20zbirniku\SEM.docx#_Toc27946884)

[Slika 3: Razdelitev števil 4](file:///D:\Dokumenti\ŠOLA\SREDNJA%20ŠOLA\Informatika%20in%20računalništvo\Seminarska%20naloga%20-%20Naslov%20v%20zbirniku\SEM.docx#_Toc27946885)

[Slika 4: Seštevanje enic in ničel 4](file:///D:\Dokumenti\ŠOLA\SREDNJA%20ŠOLA\Informatika%20in%20računalništvo\Seminarska%20naloga%20-%20Naslov%20v%20zbirniku\SEM.docx#_Toc27946886)

[Slika 5: Ugotavljanje lihe paritete 5](file:///D:\Dokumenti\ŠOLA\SREDNJA%20ŠOLA\Informatika%20in%20računalništvo\Seminarska%20naloga%20-%20Naslov%20v%20zbirniku\SEM.docx#_Toc27946887)

[Slika 6: Seštevanje stolpcev 5](file:///D:\Dokumenti\ŠOLA\SREDNJA%20ŠOLA\Informatika%20in%20računalništvo\Seminarska%20naloga%20-%20Naslov%20v%20zbirniku\SEM.docx#_Toc27946888)

[Slika 7: Sestavljanje števil 5](file:///D:\Dokumenti\ŠOLA\SREDNJA%20ŠOLA\Informatika%20in%20računalništvo\Seminarska%20naloga%20-%20Naslov%20v%20zbirniku\SEM.docx#_Toc27946889)

[Slika 8: Ugotavljanje lihe paritete za števila, sestavljena iz stolpcev 5](file:///D:\Dokumenti\ŠOLA\SREDNJA%20ŠOLA\Informatika%20in%20računalništvo\Seminarska%20naloga%20-%20Naslov%20v%20zbirniku\SEM.docx#_Toc27946890)

# UVOD

Vsaka seminarska naloga se začne z načrtom, približnim planom za potek dela. Na začetku nisem točno vedel, kako začeti, a ko sem to ugotovil, je bil nadaljnji potek dela mala malica.

Pri delanju naloge sem se spopadal s kar nekaj problemi. Najprej nisem natančno vedel, kako začeti, a sem po nekaj branjih same naloge rešil ta problem. Veliko težav sem imel tudi s programom za programiranje (Motorola 6803). Težava je v tem, da program ni optimiziran za novejše verzije operacijskega sistema Windows, npr. Windows 10. Zaradi tega se velikokrat program sploh ne zažene ali pa se projekt, na katerem delaš, ne shrani pravilno. Ampak z malo srečen in ponovnimi zagoni računalnika sem uspel končati seminarsko nalogo.

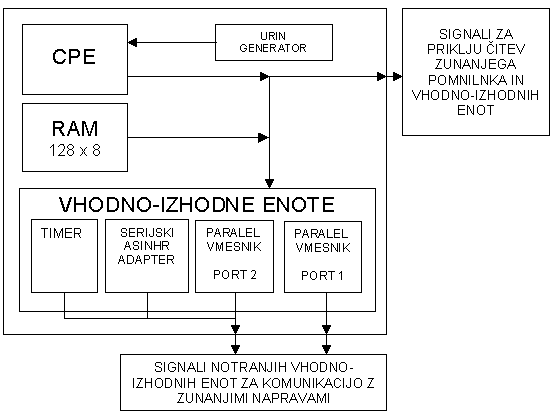
# MOTOROLA 6803

Motorola 6803 je mikrokrmilnik, ki poleg mikroprocesorja vsebuje še glavni pomnilnik in periferne enote. MC6803 uporablja osnovni nabor ukazov M6800, kar pomeni da vsi ukazi, ki smo jih napisali za M6800 delujejo na MC6803, obratno pa ne.

MC6803 vsebuje naslednje enote:

* CPE, ki je izboljšana verzija mikroprocesorja 6800. Dodani so mu nekateri novi ukazi;
* 128 zlogov RAM-pomnilnika. To je bralno pisalni pomnilnik, ki ohranja vsebino le, dokler je priključen na napajalno napetost;
* Večnamenski časovnik lahko uporabimo za generiranje raznih pravokotnih izhodnih signalov, merjenje časa oziroma frekvence, proženje periodičnih prekinitev…
* Serijski asinhronski komunikacijski adapter omogoča serijsko komunikacijo z zunanjimi napravami. Serijski adapter lahko hkrati sprejema in oddaja znake, prenos poteka časovno zaporedno (bit za bitom), za komunikacijo pa so potrebne tri linije (sprejem, oddaja in signalna masa;
* Paralelni vmesnik - port 1 je 8-bitni vmesnik, ki mu programsko določimo za vsako linijo, ali je vhodna in izhodna;
* Paralelni vmesnik - port 2 je samo 5-bitni vmesnik. Izhodne linije tega vmesnika si delijo 5 nožic mikrokrmilnika s signali serijskega vmesnika in časovnika. Programsko lahko določimo, katera od enot ima dostop do nožic mikrokrmilnika.

Slika 1: Skica mikrokrmilnika Motorola 6803



MC6803 sodi v Motorolino družino mikrokrmilnikov M6801. Deluje pri napajalni napetosti 5V, vsi njegovi signali, preko katerih komunicira z okolico, pa uporabljajo TTL-napetostne nivoje. MC6803 vsebuje dokaj zmogljive vhodno-izhodne enote in je uporaben v različnih aplikacijah. Ta mikrokrmilnik uporabljamo kot osnovo, ki ji dodamo še zunanji pomnilnik in nekatere zunanje vhodno-izhodne enote, kot jih narekuje dana aplikacija. Omogoča nam, da mikroračunalnik, ki je bil prej sestavljen iz mikroprocesorja in pripadajočih komponent, skrčimo na nekaj integriranih vezij (najmanj dve). Če želimo uporabljati mikrokrmilnik, ki deluje samostojno, pa se lahko odločimo za kakšnega drugega iz družine M6801, npr. MC68701. Ta vsebuje poleg vseh naštetih enot še 4 KB Eprom-pomnilnika, v katerega lahko sami vpišemo program (in po potrebi tudi zbrišemo). (SERŠ, 2015)

# LIHA PARITETA

V asinhronih komunikacijskih sistemih se liha pariteta nanaša na načine preverjanja paritete, kjer ima vsak niz poslanih bitov liho število le teh. Če je skupno število enot v podatkih + bit paritete liho število, je to liha pariteta. Če imajo podatki že liho število enakih, je vrednost bitja dodane parnosti enaka 0, sicer je 1.

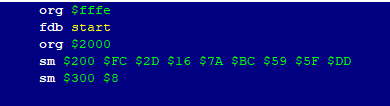
Biti paritete so najpreprostejša oblika odkrivanja napak. Preizkušanje parnosti se uporablja pri testiranju pomnilniških naprav. Pošiljatelj in prejemnik se morata strinjati z neparnim preverjanjem paritete. Brez tega uspešna komunikacija ni mogoča. Če se med prenosom preklopi liho število bitov, lahko preverjanje paritete zazna, da so podatki poškodovani. Vendar metoda ne bo zaznala napak, ki so bile vnesene, če je spremenjeno celo število bitov v isti podatkovni enoti, saj bo pariteta kljub korupciji še vedno neparna. (Technopedia, 2018)

# NALOGA

Za to seminarsko nalogo sem dobil naslednjo nalogo:

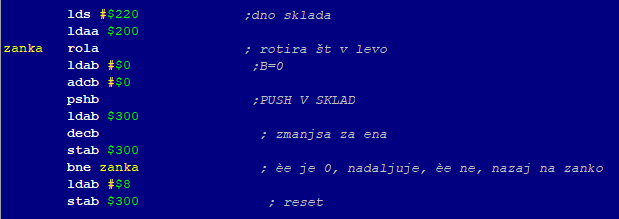
Preberi 8 števil iz pomnilnika in ugotovi ali ustrezajo lihi pariteti po vrsticah in stolpcih. Če ustrezajo, jih pošlji na izhod.

* 1. POSTOPEK



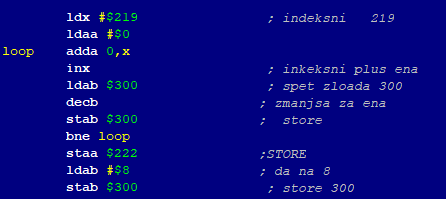
Slika 2: Začetek

Pričel sem z obvezni začetkom, prikazanim na sliki 2.



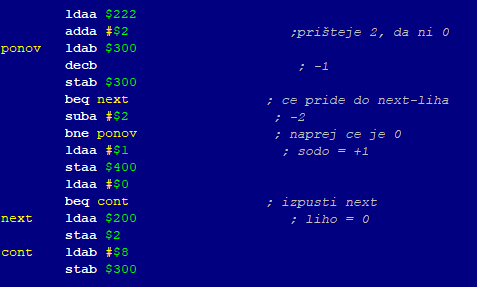
Slika 3: Razdelitev števil

Nato sem vsako število razdelil na enice in ničle. Postopek sem ponovil za vseh osem števil.



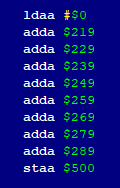
Slika 4: Seštevanje enic in ničel

Nadaljeval sem s seštevanjem enic in ničel. Postopek sem ponovil za vseh osem števil.

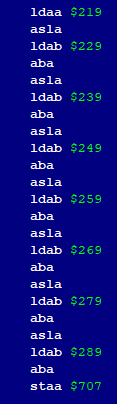


Slika 5: Ugotavljanje lihe paritete

Potem sem ugotovil, ali seštete vsote enic in ničel ustrezajo lihi pariteti. Če ustrezajo, zapišemo 0. To ponovimo za vseh osem števil.

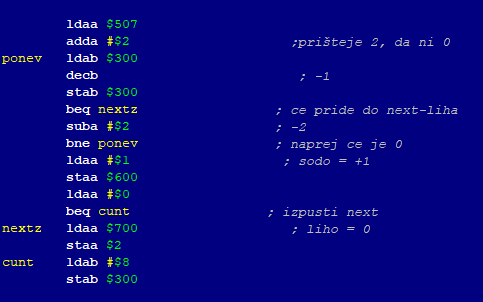


Slika 6: Seštevanje stolpcev

Nato sem sešteval stolpce, da sem lahko kasneje preveril, če ustreza lihi pariteti. Seštevanje sem ponovil za vseh osem stolpcev.

Slika 7: Sestavljanje števil

Nadaljeval sem s sestavljanje števil iz stolpcev, da sem jih lahko poslal na izhod. Sestavil sem števila iz vseh osmih stolpcev.



Slika 8: Ugotavljanje lihe paritete za števila, sestavljena iz stolpcev

Končal sem z ugotavljanjem lihe paritete za števila, sestavljena iz stolpcev. Postopek sem ponovil za vseh osem sestavljenih števil.

* 1. UKAZI

Pri programiranju sem uporabil ogromno različnih ukazov, tako da bom poskusil našteti vsaj nekaj izmed njih:

* start – začetek programa
* sm – nastavitev vrednosti izbranega naslova
* ldaa – v akumulator A naloži ali vrednost izbranega naslova ali samo vrednost
* rola – rotiranje v levo
* adda – vrednosti v akumuzlaotrju A doda ali samo vrednost ali vrednost izbranega naslova
* asla – premik bita v levo brez skoka v carry
* lds – nastavitev dna sklada
* end – konev programa

(Dickens, 2018)

# ZAKJUČEK

Program sem po vsakem narejenem koraku testiral in preveril, če vse stvari delujejo kot bi morale. Na srečo ni bilo težav, kar je zelo dobro, saj si noben ne želi popravljati programa, dolgega 773 vrstic.

Naloga, ki mi je bila zadana, se mi je sprva zdela težka in nisem si mislil, da mi bo uspelo. Je ena težjih in predvsem daljših nalog. Za pomoč sem naprej vprašal profesorja Gregorja Medeta, ki mi je z veseljem priskočil na pomoč in mu pomagal z idejo, kako začeti. Ko sem to ugotovil, je bila stvar mnogo lažja. Če sem v nadaljevanju naletel na kakšno težavo, sem jo skušal rešiti sam (s preizkušanjem) ali za pomoč vprašal sošolce.

Mogoče se mi je na prvi pogled naloga zdela malce čudna, smo jo delali in realizirali v simulatorju, starem več kot 30 let. Mogoče bi lahko poskusili kaj novejšega, a se strinjam s profesorjem Zorkom, ki pravi, da je program odličen za programerje začetnike, kot nas je večina dijakov v drugem letniku.

# VIRI IN LITERATURA

Dickens, T. (2018). *68HC11 Instruction Set*. Pridobljeno iz https://www.ele.uri.edu/Courses/ele205/6811-Instructions/index.html

SERŠ. (2015). *SERŠ*. Pridobljeno iz SERŠ: http://www.s-sers.mb.edus.si/gradiva/w3/sistemi/m6803.html

Technopedia. (2018). *Technopedia*. Pridobljeno iz Technopedia: https://www.techopedia.com/definition/1801/odd-parity

# STVARNO KAZALO

liha pariteta, 3

MC6803, 3, 2, 3

naloga, 3, 1, 4, 7

Windows 10, 1