

Nenad Hrnjak, Marija Jevtić, Andrej Zdravković
Ivo Lola Ribar, Bulevar Revolucije 84, Beograd

**STRUKTURA PROGRAMSKE PODRSKE SISTEMA ZA PRAĆENJE
TOKA IZVRŠENJA KORISNIČKOG PROGRAMA LPA-512**

**LPA-512 APPLICATION PROGRAM EXECUTION
MONITORING SYSTEM SOFTWARE STRUCTURE**

SADRŽAJ: U radu je prikazana programska podrška za praćenje toka izvršenja korisničkog programa na programabilnom automatu LPA-512. Posle napomena o ograničenjima nametnutim specifičnom hardverskom realizacijom sledi opis realizovanih monitorskih podsistema: praćenje izvršenja korisničkog programa po funkcionalnim blokovima, praćenje stanja elemenata po zadanom uslovu i upravljanje tokom izvršenja korisničkog programa. Razmatrane su varijante programabilnog automata sa korisničkim programom u ROM i RAM memoriji.

ABSTRACT: The paper presents programmable logic controller LPA-512 application program execution monitoring system software structure. Brief description of the dedicated hardware limitations is followed by realized monitor subsystems presentation: program functional blocks execution monitoring, trace monitoring and program flow controlling. Programmable controllers LPA, ROM and RAM application program memory versions are taken into account.

1. UVOD

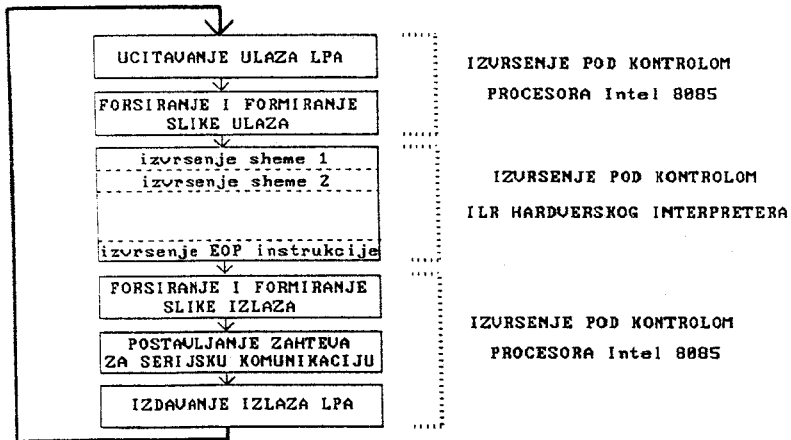
Bitne komponente programskog sistema za razvoj aplikativnog programa su alati za olakšavanje testiranja razvijenog programa. Program za programabilni automat razvija se na problemski orijentisanom jeziku lestvičastih dijagrama, standardnom za ovakav tip industrijskog računara opšte namene. Ovaj programski jezik, zajedno sa problemima upravljanja u realnom vremenu uslovljava veoma specifičnu realizaciju sistema za testiranje aplikativnih programa na programabilnom automatu. U programskom sistemu LPA-512 praćenje toka izvršenja programa omogućeno je preko uredjaja za programiranje LPROG koji je sa LPA-512 spregnut serijskom komunikacijom.

U radu je prikazana struktura programske podrške

programabilnog automata LPA-512 i uređaja za programiranje LPROG, orijentisane na omogućavanje testiranja korisničkog programa. Prvi deo rada pored osnovnih napomena o specifičnoj hardverskoj realizaciji objašnjava princip izvršenja korisničkog programa u LPA-512. Podsystem praćenja toka izvršenja korisničkog programa po shemama programa prikazan je u drugom delu rada. Treći deo rada je opis podsystema za praćenje stanja promenljivih u programu prema zadatom logičkom uslovu, dok četvrti deo rada prikazuje mogućnosti upravljanja tokom programa preko dodeljivanja vrednosti promenljivima.

2. OSNOVNI PRINCIPI IZVRŠENJA KORISNIČKOG PROGRAMA NA LPA-512

Osnovni tok izvršenja korisničkog programa na programabilnom automatu LPA-512 prikazan je slikom 1.



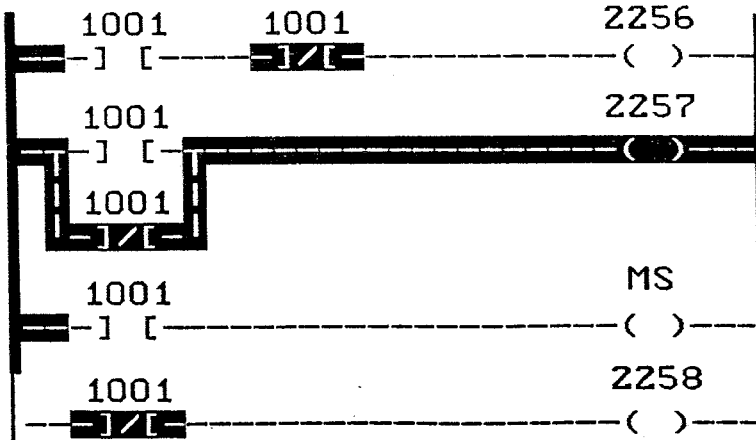
Slika 1. Osnovni tok izvršenja korisničkog programa na LPA 512

Program je podeljen na funkcionalne blokove koje nazivamo sheme. Izvršenje korisničkog programa je sekvencijalno, naime sheme, kao i programski redovi u okviru shema, se izvršavaju jedna za drugom. Specifičnost izvršenja programa u programabilnom automatu je i ciklično izvršenje korisničkog programa. Jedan ciklus izvršenja programa naziva se skanom programabilnog automata.

Radi ubrzanja izvršenja korisničkog programa na LPA-512 primenjeno je specifično rešenje: korisnički program se izvršava interpretacijom na hardverskom interpreteru. Ovakav koncept realizacije je povećao brzinu izvršenja korisničkog programa za red veličine, ali je uneo i značajne realizacione teškoće. Naime procesor nema kontrolu nad izvršenjem korisničkog programa osim u trenucima hardverskog prekida ili pri interpretaciji instrukcija programskog prekoda, tako da ne postoji ažurna informacija o toku korisničkog programa. [1]

3. PRAĆENJE IZVRŠENJA KORISNIČKOG PROGRAMA PO ŠEMAMA

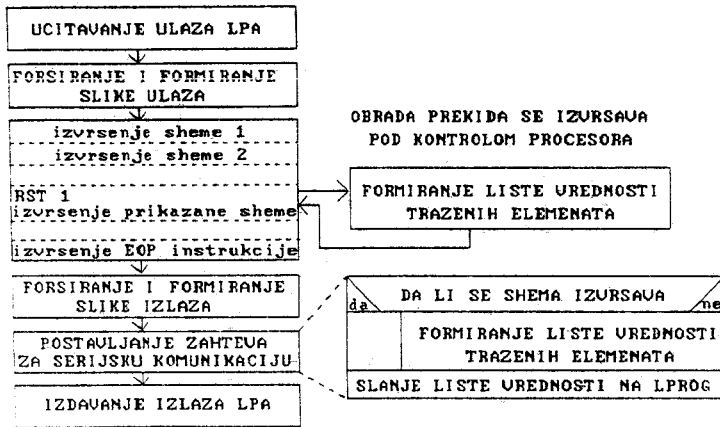
Organizacija korisničkog programa za LPA nameće potrebu za praćenjem promene stanja na funkcionalnom bloku programa - shemi. Monitorski režim koristi osnovnu karakteristiku lestvičastih dijagrama, grafički prikaz protoka struje kroz kontakte relea. Na slici 2. prikazan je izgled sheme u režimu monitor.



Slika 2. Izgled sheme u režimu monitor

Način izvršenja korisničkog programa i tip memorije za korisnički program LPA uslovljavaju dve varijante rada

monitorskog režima : ROM i RAM. Za prikazanu šemu na LPROG-u se formira adresa početka sheme u okviru korisničkog programa, kao i lista elemenata sheme koja se zajedno sa adresom šalje na LPA. Ova lista predstavlja zahtev za početak praćenja stanja elemenata prikazane sheme. Na osnovu primljene liste LPA u svakom skanu formira listu stanja traženih elemenata koju šalje na LPROG. U zavisnosti od tipa korisničke memorije postoje dva načina formiranja liste stanja i vrednosti po primljenoj listi elemenata.



Slika 3. Izvršenje jednog skana - RAM varijanta LPA 512

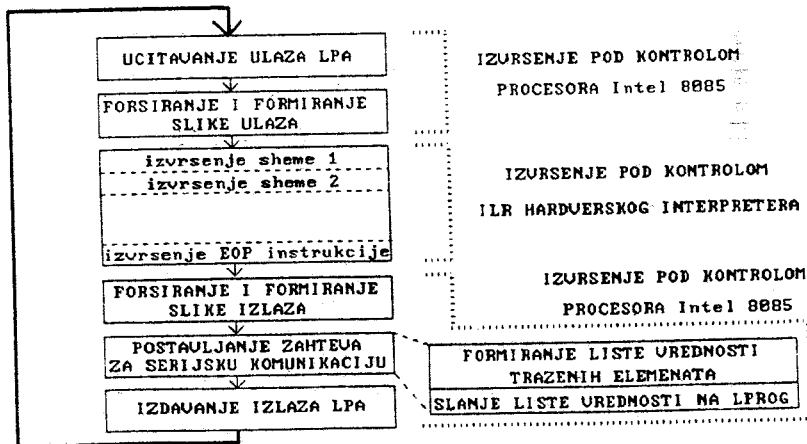
- LPA _ tip korisničke memorije RAM

Kod RAM varijante LPA se na osnovu primljene adrese vrši modifikacija korisničkog programa upisom prekidne tačke. Na mesto instrukcije koja se nalazi na toj adresi upisuje se kod koji hardverski interpreter prepoznaje kao softverski prekid RST 1. Rutina za obradu prekida prepoznaje prekidnu tačku nastalu promenom programa i na osnovu primljene liste adresa formira listu stanja i vrednosti promenljivih pre izvršenja prve instrukcije sheme. U listi se takodje nalazi i vrednost opšteg uslova aktiviranja izlaza za tu shemu. Takodje je uveden i poseban indikator na osnovu kojeg LPROG identifikuje da li se

shema izvršava. U slučaju kada se ne izvršava neće se izvršiti ni prekidna rutina već se lista formira kao i u ROM varijanti LPA. Zbog promene korisničkog programa takodje se vrši i softverska interpretacija prepisane instrukcije. Po završetku monitorisanja seme i pri havarijskim situacijama aktivira se posebna rutina za restauraciju programa izmenjenog upisom prekidne tačke. Princip izvršenja jednog skana programa u RAM varijanti LPA prikazan je slikom 3.

- LPA - tip korisničke memorije ROM

ROM varijanta LPA isključuje promenu korisničkog programa i upis prekidne tačke. Lista stanja i vrednosti promenljivih formira se po izvršenju korisničkog programa i izdavanja stanja izlaza na fizičke izlaze LPA. Korišćenje istog izlaza više puta u programu može dovesti do promene stanja izlaza tokom izvršenja programa što u nekim slučajevima rezultira pogrešnom listom vrednosti za datu shemu. Princip izvršenja jednog skana programa na ROM varijanti LPA prikazan je slikom 4.



Slika 4. Izvršenje jednog skana - ROM varijanta LPA 512

RAM varijanta LPA omogućuje da se u monitorskom režimu dobije realna slika izvršenja prikazane sheme. Zbog ovih karakteristika,

pri razvijanju, testiranju i praćenju programa RAM, kao tip korisničke memorije, je pogodniji za korišćenje.

Na osnovu primljene liste vrednosti ulaznih i izlaznih elemenata na LPROC-u se vrši potpuna simulacija rada LPA za prikazanu shemu programa, pri čemu se elementi sa aktivnim stanjem prikazuju inverzno.

- Praćenje stanja na relejnim shemama

Na relejnim shemama se koriste logički ekvivalenti AND, OR i NOT operacija Bulove algebre nad simbolima relejnih elemenata. Na shemi se simulira protok struje kroz normalno otvorene i normalno zatvorene kontakte, kao i kroz grananja, ukoliko postoje. U slučaju da su svi uslovi za aktiviranje izlaznog elementa ispunjeni i on je prikazan invertovano. Takođe se prate i izlazi kojima je vrednost dodeljena forsiranjem. Bez obzira na stanje dobijeno izračunavanjem vrednosti reda na ekranu se prikazuje forsirano stanje.

Tokom izvršenja sheme dolazi i do promene stanja izlaznih elemenata. U slučaju da se na shemi isti izlaz zatim javlja i kao kontakt u okviru te sheme za vrednost kontakta se uzima ne vrednost primljena u listi, već izračunata vrednost. Pored stanja tajmera i brojača na shemi se prikazuju i njihove tekuće vrednosti. Ukoliko tokom izvršenja sheme dodje do promene primljenih vrednosti i one se modifikuju.

- Praćenje stanja na numeričkim shemama

U okviru numeričkih shema vrši se i obrada osmobitnih numeričkih podataka. Definisane su operacije sabiranja i oduzimanja podataka, relacije poredjenja podataka i prenosa podataka. U monitorskom režimu se u potpunosti simuliraju ove operacije i relacije. Pored numeričkih elemenata se prikazuju njihove trenutne vrednosti. Takođe se simulira rad sistemskih ulaza/izlaza za množenje, deljenje, BCD <-> binarno prekodiranje i logičke operacije nad osmobitnim podacima (NOT, AND, OR, XOR).

RAM varijanta LPA omogućuje i da se na shemi koja se ne izvršava

elementi ne prikazuju inverzno. Na vrhu ekrana je poruka da se shema ne izvršava, a prikazuju se samo trenutne vrednosti numeričkih elemenata i tekuće vrednosti tajmera i brojača.

4. PRAĆENJE STANJA PROMENLJIVIH LPA-512 PO ZADATOM USLOVU

Kod testiranja i praćenja izvršenja programa korisna je i mogućnost praćenja stanja promenljivih po listi adresa u određenom vremenskom intervalu. Zadavanjem uslova određuje se trenutak početka ili kraja praćenja stanja.

Na sistemu LPROG postoji mogućnost praćenje stanja jednobitnih ulaza/izlaza i tajmera/brojača u vremenskom intervalu od 73 skana, od trenutka koji je zadat uslovima.

Maksimalan broj promenljivih čije se vrednosti prate je 19 pri čemu je broj logičkih AND operanada za formiranje logičkih uslova najviše 4. Ukupan broj uslova je limitiran veličinom ekrana. Na osnovu unetih uslova i liste adresa za praćenje formira se lista adresa koja se šalje na LPA.

I ovde postoje dve varijante rada u zavisnosti od tipa korisničke memorije. Kod ROM varijante LPA lista stanja se formira po izvršenju korisničkog programa, dok se kod RAM varijante lista formira pre izvršenja sheme koja je pre izbora ovog podrežima bila prikazana na ekranu. Ukoliko je potrebna lista stanja promenljivih po izvršenju korisničkog programa, ovaj podrežim treba izabrati sa poslednje sheme programa.

Po prijemu 73 liste stanja od zadanog trenutka, na ekranu LPROG-a se pored zadate adrese prikazuju primljene vrednosti. Aktivna stanja su data inverzno. Pored zadatih promenljivih uvek je prikazana i vrednost sistemskog izlaza 4005 koji predstavlja taktni impuls skaniranja.

5. UPRAVLJANJE IZVRŠENJEM KORISNIČKOG PROGRAMA DODELJIVANJEM VREDNOST PROMENLJIVIM LPA-512

Važna karakteristika monitorskog režima LPROG-a za testiranje korisničkog programa je i mogućnost simulacije pojedinih stanja procesa dodeljivanjem vrednosti promenljivama LPA tzv.

forsiranjem elemenata.

Pored forsiranja logičkih stanja relejnih elemenata, moguće je zadavati i vrednosti numeričkim ulazima i izlazima. Tekuće vrednosti tajmera i brojača se ne mogu na ovaj način zadati.

LPA forsiranje ulaznih elemenata vrši posle skaniranja ulaza a pre izvršenja korisničkog programa, dok se forsiranje izlaza vrši po izvršenju programa, a pre skaniranja izlaza. Na taj način forsirana vrednost izlaza ne utiče na izvršenje korisničkog programa već izlaz dobija tu vrednost tek po završetku programa. Ne postoji mogućnost forsiranja izlaznog elementa koji nije kao izlaz definisan u korisničkom programu. Pretpostavlja se da forsiranje takvog izlaza nema nekog posebnog značaja za testiranje programa.

Vrednost koja je jednom dodeljena forsiranjem ostaje važeća do ukidanja forsiranja, ili prestanka rada LPA. Moguće je ukinuti sva forsiranja zajedno ili ih ukinuti pojedinačno. Po ukidanju forsiranja, izlazni element ostaje u nepromenjenom stanju dok programskim putem ne dodje do njegove izmene. Na LPROG-u, u posebnom podržimu, postoji i mogućnost praćenja liste adresa svih elemenata forsiranih na "0", na "1" kao i liste adresa forsiranih numeričkih elemenata sa njihovim vrednostima.

6. Zaključak

U radu je prikazana programska podrška sistema za omogućavanje testiranja korisničkog programa LPA 512. Opisani sistem omogućava uvid u izvršenje korisničkog programa LPA 512 u realnom vremenu, kao i uticaj na tok izvršenja programa. Posebni podržim omogućava praćenje vrlo brzih promena stanja preko definisanja uslova praćenja. Ovim podsistemom omogućena je jednostavna kontrola ispravnosti korisničkog programa, čime je značajno skraćen i olakšan proces instalacije i verifikacije ispravnosti upravljačkog sistema na bazi LPA 512. Osnovni nedostatak uređaja za programiranje je nemogućnost simulacije celog korisničkog programa na uređaju za programiranje, bez učešća LPA 512. Ovaj nedostatak biće otklonjen u novoj familiji programabilnih automata kojase razvija u ILR.

7. Literatura

- [1] J. Medved, I. Latinović: Hardverski koncept programabilnog automata LPA 512, 1988, Lola Saopštenja 23