

primer preprostega mikroračunalnika z mikroprocesorjem motorola 6800

d. kodek

UDK 681.3-181.4

Fakulteta za elektrotehniko,
Ljubljana

Večina proizvajalcev mikroprocesorskih vezij podaja skupaj z opisom lastnosti vezij tudi tako imenovane minimalne sistemske konfiguracije. Te konfiguracije so običajno sestavljene iz 2 do 5 LSI vezij in postanejo uporabne šele, ko jih opremimo z ustreznim programom. V članku je opisan primer take konfiguracije na osnovi mikroprocesorja MOTOROLA 6800 in kontrolni program, ki omogoča njeno uporabo za različne namene.

AN EXAMPLE OF SIMPLE MICROCOMPUTER BASED ON MOTOROLA 6800: Most manufacturers of microprocessor circuits include so called minimum system configurations in their product description sheets. These configurations usually consist of 2 to 5 LSI circuits and are usable only if we support them with some kind of a program. The paper gives an example of such configuration based on MOTOROLA 6800 microprocessor and control program, which allows its use for different purposes.

1. UVOD

Danes je že povsem jasno, da predstavljajo mikroprocesorji veliko prelomnico v razvoju in uporabi digitalnih sistemov raznih vrst. Upoštevajoč vse prednosti programsko vodenega delovanja in današnje cene mikroprocesorskih elementov, je pravzaprav že težko najti primere, v katerih z mikroprocesorji tako ali drugačne vrste ne bi mogli zamenjati fiksno ožičene logike. To še posebej velja za počasnejše digitalne naprave, ki jih je mogoče zelo uspešno in hitro realizirati z uporabo MOS mikroprocesorjev.

Kljub veliki aparturni preprostosti z mikroprocesorji narejenih naprav in kljub dejstvu, da lahko isto napravo s preprosto zamenjavo programa uporabimo za povsem različne namene, pa so uporabniki kmalu spoznali, da uvajanje mikroprocesorjev ni vedno preprosto. Problemi, ki se pojavljajo, so predvsem posledica pomanjkanja izkušenj pri delu s programsko vodenimi sistemi in razmeroma skromne opreme, ki jo ima na voljo večina uporabnikov. Kot pomoč pri reševanju teh problemov je v tem članku opisan preprost mikroračunalnik s kontrolnim programom, ki omogoča prikaz in študij lastnosti mikroprocesorja MOTOROLA 6800 in ki je istočasno uporaben tudi pri realizaciji in preizkušanju različnih kontrolnih naprav.

2. KONFIGURACIJA

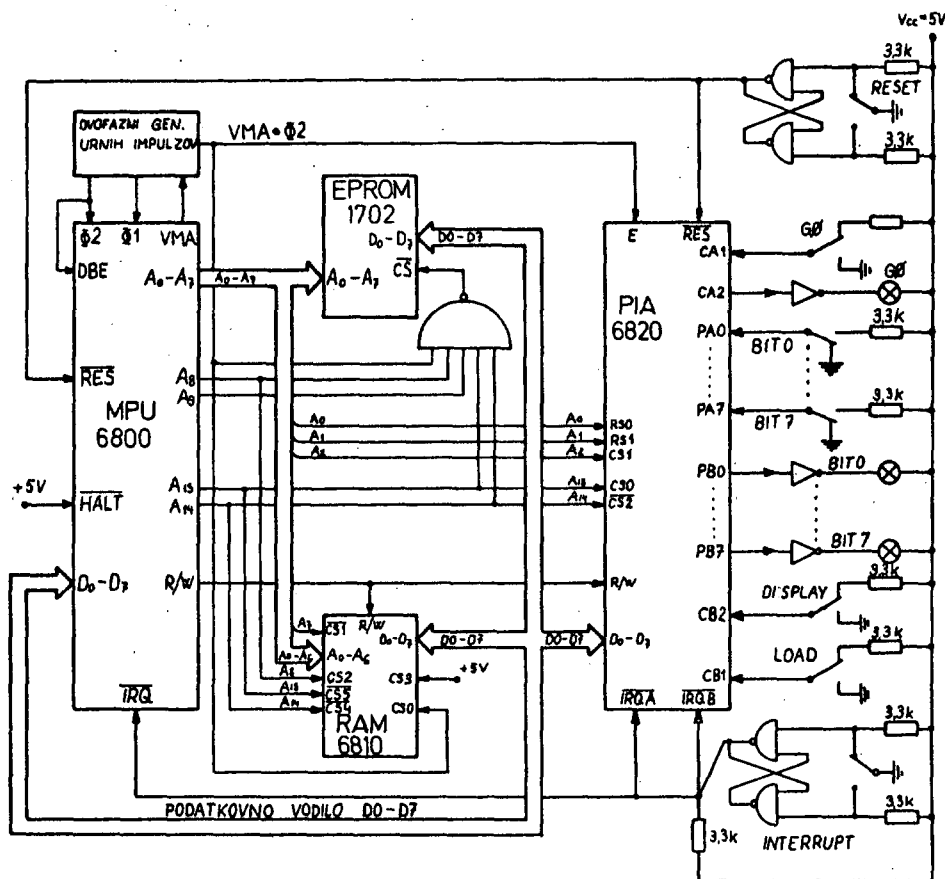
Pri izbiri mikroračunalniške konfiguracije so bili najpomembnejši kriteriji njena preprostost, cenenost in možnost za hitro realizacijo. Vse to ob zahtevi, da je omogočeno preprosto programiranje in da je po potrebi možno dodati še dodatne elemente za posebne namene.

Tem zahtevam je najlažje zadostiti z izpeljanko ene od minimalnih sistemskih konfiguracij, ki jih podaja proizvajalec / 1 /. Značilno za te konfiguracije je, da zahtevajo zelo malo vmesnih TTL vezij in so zato zelo primerne za naš namen.

Uporabljena konfiguracija je prikazana na sliki 1. Namesto težko dosegljivega in razmeroma dragega EPROM pomnilnika S6834, ki bi po proizvajalčevih napotkih sodil v to konfiguracijo, je bil uporabljen starejši in cenejši Intelov 1702 EPROM pomnilnik. Čeprav je kapaciteta tega pomnilnika samo 256 x 8 bitov, je to za naše namene dovolj. Tudi programatorji za te elemente pri nas niso več redkost in zato njihova uporaba ni problematična.

Kot je razvidno iz slike 1 vsebuje mikroračunalnik poleg pomožnih TTL vezij samo 4 LSI vezja. Pri današnjih cenah elementov znaša vrednost sestavnih delov nekaj več kot 1000 dinarjev, pri čemer odpade več kot 80% na LSI vezja. Uporabnik ima na voljo 128 8-bitnih besed v RAM pomnilniku (adrese 0 do 7F), 256 8-bitnih besed v EPROM pomnilniku (adrese FF00 do FFFF), 8 stikal kot vhodno enoto, 8 svetlečih diod kot izhodno enoto, 5 komandnih tipk (RESET, INTERRUPT, LOAD, DISPLAY in GO) in 1 svetlečo diodo kot kontrolni indikator. Potrebujemo napajalni napetosti 5V in -9V; pri tem je napetost -9V potrebna samo za EPROM pomnilnik.

Zijjemo tipk RESET in INTERRUPT, ki delujeta neposredno na vhode mikroprocesorja, so vse ostale tipke, stikala in svetleče diode priključene preko perifernega adapterja PIA (adrese 2004 do 2007). Način njihovega delovanja in s tem njihova funkcija, je zato povsem pod programsko kontrolo in jo lahko z zamenjavo programa spremenimo. Zaradi preprostosti tudi ni bilo realizirano popolno deko-



Slika 1: Konfiguracija

diranje adres in so ostali adresni biti A10, A11, A12 in A15 neuporabljeni. Pri razširjanju konfiguracije je zato potrebno upoštevati to zoženost adresnega prostora.

3. KONTROLNI PROGRAM

Omenili smo, da je z izjemo tipk RESET in INTERRUPT, delovanje vseh ostalih tipk in vseh stikal in svetlečih diod, pod programsko kontrolo. Brez ustreznega programa se bo mikroročunalnik s slike 1 odzval vedno samo na tipko RESET, na tipko INTERRUPT pa samo v primeru, ko maskirni bit I v mikroprocesorju ni postavljen. Ker je poleg brisanja registrov v mikroprocesorju edina posledica RESET signala ta, da se programski števec v mikroprocesorju napolni z vsebino adres FFFF in FFFE, je naš mikroročunalnik brez kontrolnega programa povsem neuporaben.

V našem primeru mora kontrolni program omogočati, da lahko uporabnik preko tipk, stikal in svetlečih diod vpisuje in izvaja poljubne programe. Razumljivo je, da bo pri tako skromni konfiguraciji uporabnik moral programirati v strojnem jeziku. Nujno je tudi, da se kontrolni program, ki to programiranje in izvajanje omogoča, nahaja v EPROM pomnilniku, saj bi se sicer ob izklopu napetosti izgubil. Uporabnik ima torej za svoje programe na voljo 128 8-bitnih besed v RAM pomnilniku, kar zadošča za spoznavanje mikroprocesorja, obenem pa omogoča realizacijo in preizkušanje raznih manjših kontrolnih naprav.

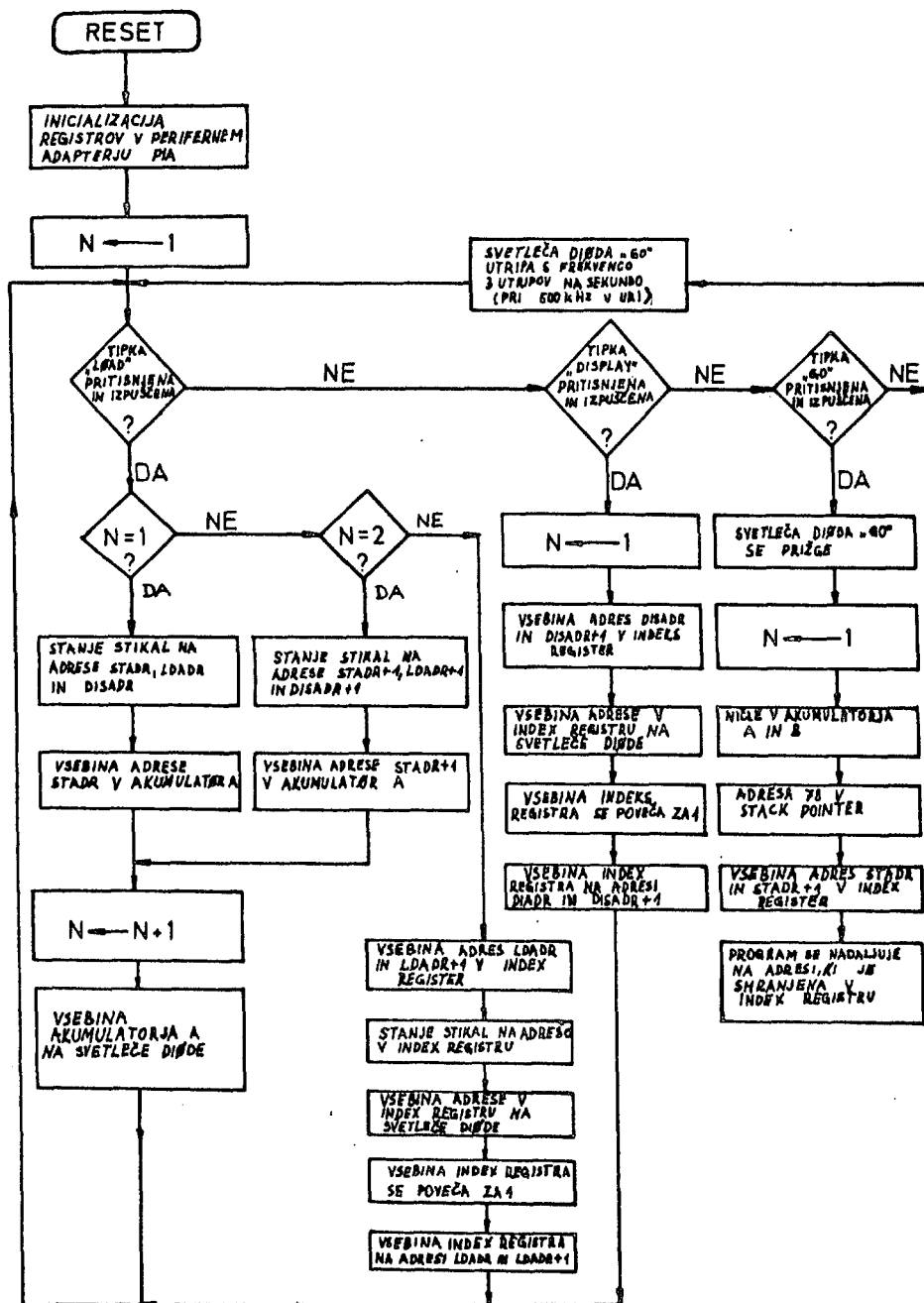
Princip delovanja kontrolnega programa je prikazan na sliki 2. Program poženemo s pritiskom na tipko RESET in se prične z inicializacijo registrov v perifernem adap-

terju PIA. Periferne priključke PA0 do PA7 konfiguriramo kot vhode (ničle v data direction registru A), priključke PB0 do PB7 pa kot izhode (enice v data direction registru B). Kontrolne priključke CA1, CB1 in CB2 konfiguriramo kot vhode, ki reagirajo na zadnjo fronto, (bit CRA1 = CRB1 = 0, CRA4 = CRB4 = 0, CRB5 = 0), medtem ko je priključek CA2 konfiguriran kot izhod (bit CRA5 = 1). Izhoda IRQA in IRQB onemogućimo (CRA0 = CRB0 = 0, CRA3 = CRB3 = 0), ker želimo imeti detekcijo aktiviranja tipk brez programskih prekinitev.

Po inicializaciji vpišemo v spremenljivko N vrednost 1 in pričnemo z otipavanjem stanja tipk. Ker so tipke LOAD, DISPLAY in GO priključene neposredno na vhode CB1, CB2 in CA1, je s posebno subrutino potrebno preprečiti, da se odskakovanje kontaktov v tipkah ne detektira kot večkratno aktiviranje tipke. Ker pride do odskakovanja tako takrat, ko pritisnemo na tipko, kot takrat, ko jo izpustimo, vsebuje subrutina po detekciji zadnje fronte (ob pritisku na tipko) najprej zakasnitev, ki izloči fronte zaradi odskakovanja, nato pa čaka na pozitivno fronto (ob izpustitvi tipke) in se šele po njeni detekciji in ponovni zakasnitvi zaradi odskakovanja, vrne v glavni program.

Aktivnosti, ki se sprožijo ob aktiviranju posameznih tipk so razvidne iz diagrama na sliki 2. S prvim in drugim aktiviranjem tipke LOAD vpišemo preko stikal 16-bitno adresu v spremenljivke STADR, LDADR in DISADR. Ob nadaljnjih aktiviranjih tipke LOAD se vsebina stikal prične vpisovati na adresu, ki se nahaja v spremenljivki LDADR in ki se po vsakem vpisu poveča za 1.

Ob aktiviranju tipke DISPLAY se na svetlečih diodah prikaže vsebina, ki se nahaja na adresi shranjeni v spre-



Slika 2: Princip delovanja kontrolnega programa

menljivki DISADR. Ob vsakem prikazu se vrednost adrese v spremenljivki DISADR poveča za 1. Ob aktiviranju tipke GO se prižge svetleča dioda GO, mikroročunalnik pa nadaljuje z izvajanjem programa na adresi, ki se nahaja v spremenljivki STADR. Ta adresa je običajno začetek uporabnikovega programa. Aktiviranje tipke GO zato pomeni konec delovanja kontrolnega programa in s tem možnost, da se stikala, tipke in svetleče diode z novim konfiguriranjem perifernega adapterja PIA uporabijo za povsem druge namene.

Kontrolni program je napisan tako, da za skoke v subrutine ne uporablja ukazov JSR ali BSR. S tem smo se izognili uporabi sklada (stack) in dosegli, da program teče tudi takrat, kadar v mikroročunalniku ni RAM pomnilnika. V tem primeru programiranje seveda ni možno, lahko pa preizkušamo pravilnost delovanja kontrolnega programa. Med izvajanjem kontrolnega programa utripa svetleča dioda GO (pri frekvenci generatorja urinih impulzov 500 kHz približno 3 krat v sekundi), kar služi za indikacijo pravilnosti delovanja kontrolnega programa in celotnega mikroročunalnika.

4. ZAKLJUČEK

Opisani mikroračunalnik je bil realiziran v ISKRI Elektromehanika, Kranj in se uspešno uporablja za študij in za preizkušanje različnih razvojnih hipotez. Avtor bi se rad ob tem zahvalil mag. A. Uratniku, ki je dal pobudo za izdelavo tega dela in diplomiranim inženirjem M. Rogaču,

J. Kožuhu in in L. Peternelju, ki so sodelovali pri realizaciji.

5. LITERATURA

/1/ Motorola Semiconductor Products Inc., "M6800 APPLICATION MANUAL", Motorola Inc., 1975.