

Emulacija računalnika Iskra Delta Partner

Matej Horvat, Jurij Mihelič

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija
E-pošta: matej.horvat@guest.arnes.si, jurij.mihelic@fri.uni-lj.si

Povzetek. V osemdesetih letih prejšnjega stoletja je bila slovenska računalniška industrija na višku svojega razvoja, dandanes pa njenih dosežkov tako rekoč ni nikjer več zaslediti. Ohranjanje slovenske računalniške dediščine ima velik pomen iz več zornih kotov: od izobraževalnega in raziskovalnega do ohranjanja nacionalne identitete. V tem članku predstavimo dejavnosti, ki smo jih izvedli za ohranjanje enega pomembnejših slovenskih računalnikov (Iskra Delta Partner). Pri tem se osredinimo predvsem na izdelavo emulatorja računalnika, kar vključuje emulacijo procesorja in več različnih pripadajočih naprav. Emulator smo izdelali v programskem jeziku C in s tem omogočili njegovo učinkovitost in prenosljivost, pri čemer ga je s pomočjo ustreznih orodij mogoče izvajati celo v spletnem brskalniku. V prvem delu članka predstavimo osnovne koncepte emulacije, nato sledi kratka predstavitev emuliranega računalnika. V glavnem delu pa predstavimo izdelavo emulatorja, pri čemer podrobno opišemo tudi emulacijo posameznih naprav.

Ključne besede: arhiviranje, CP/M, emulator, navidezni stroj, obratni inženiring, ohranitev, Z80

Emulation of the Iskra Delta Partner computer

In the 1980s, the computer industry in Slovenia was at its peak of development, and nowadays its achievements are practically nowhere to be found. Preservation of the Slovene computer heritage is of great importance from several perspectives, from educational and research to preservation of the national identity. In this paper we present various activities that we carried out for the conservation of one of the most important Slovenian computers, the Iskra Delta Partner. In doing so, we focus primarily on creating a computer emulator, which involves emulation of the processor and several related devices. We created the emulator in the C programming language, thus enabling its efficiency and portability, while with the help of appropriate tools, it can also be executed within a Web browser environment. In the first part of the paper we discuss the basic concepts of emulation and briefly examine the emulated computer. In the main part we present the emulator, its construction, and give a detailed description of the emulated devices.

1 UVOD

V današnjem času v svetu pretežno prevladujeta računalniška tehnologija in oprema, zasnovani v zahodnih in izdelani v vzhodnih državah. Verjetno si le malokdo predstavlja, da je nekoč bila tudi Slovenija (še kot del Jugoslavije) zelo aktivna pri tem razvoju in izdelavi. Med drugim je podjetje Iskra Delta v 80. letih prejšnjega stoletja razvijalo in izdelovalo strojno opremo – tako namizne kot velike računalnike –, namenjeno poslovni rabi, in programsko opremo zanje [1].

V primerjavi s hišnimi računalniki tistega časa je te računalnike uporabljalo relativno malo ljudi. Prav

tako niso bili izdelani v velikih količinah, zato je te računalnike in podrobne informacije o njih danes težko najti; enako velja za ljudi, ki imajo specifične informacije o njih. Poleg tega se strojna oprema in povezana tehnologija zelo hitro starata, zato na slovenske starodobne računalnike preži nevarnost pozabe.

Fizično ohranjanje starodobne strojne opreme lahko sčasoma postane zapleten izziv (npr. odpoved posameznih komponent zahteva dodatno elektrotehnično znanje in morebitne nadomestne dele). Ohraniti programsko opremo pa je relativno lahko, če za to poskrbimo pravočasno, da imamo na voljo še dovolj delujočo strojno opremo, dokumentacijo in seveda znanje s področja temeljnih ved o delovanju računalnikov, npr. področij računalniške arhitekture in sistemske programske opreme. Pri tem je treba izvesti prenos programske opreme na sodoben računalnik, kjer si lahko pomagamo z izdelavo bitnih slik starih pomnilnih medijev (npr. disket in tudi trajnega bralnega pomnilnika računalnika), s čimer zagotovimo popolno istovetnost podatkov. Področje, ki se ukvarja s tem, se imenuje digitalno ohranjanje (angl. digital preservation), emulacija pa je le ena od specifičnih tehnik [2], [3] tega področja.

Dobro programsko posnemanje strojne opreme lahko dosežemo tako, da ustvarimo program, ki izvaja funkcije in opravila procesorja in pripadajočih vhodno-izhodnih (V/I) naprav enako kot imitirani računalnik. Takemu programu pravimo *emulator* in omogoča izvajanje programske opreme na drugem računalniku (*gostitelj*) kot na tistem, za katerega je bila prvotno razvita (*gost*). Emulirati pomeni implementirati vmesnik in funkcionalnost nekega sistema na sistemu z drugačnim vmesnikom in funkcionalnostjo [4]. Poleg izvajanja programske

opreme na drugem procesorju (kar omogoča *navidezni stroj*) emulator poustvarja tudi V/I naprave; tako gostujoči program praviloma ne more ugotoviti, da se ne izvaja na prvotni strojni opremi. Emulacija se pogosto uporablja tudi v izobraževalne namene [5], [6], tako imenovana emulacija celotnega sistema [7] pa je pogosto edini dolgoročni in celoviti način za njegovo ohranitev [8], [9], [10].

V nadaljevanju članka predstavimo razvoj in implementacijo emulatorja za računalnik Partner podjetja Iskra Delta. Emulator kot gostitelja uporablja danes "standardni" PC (angl. personal computer) ali Mac računalnik. V naslednjem razdelku opišemo tehnične lastnosti računalnika s poudarkom na procesorju, pomnilniku in nekaterih V/I napravah. Poleg tega podamo še nekaj podrobnosti o operacijskem sistemu CP/M, ki ga uporablja Partner. Glavni prispevek članka pa je zbran v tretjem poglavju, kjer podrobno predstavimo izdelani emulator. Najprej opišemo njegovo splošno zgradbo, potem pa se osredinimo na posamezne komponente, kjer podrobno opišemo predvsem emulacijo pomembnejših V/I naprav. V četrtem poglavju podamo kratko ovrednotenje naše rešitve, opišemo zanimivejše težave in rešitve iz ozadja razvoja emulatorja in obratnega inženiringa, ki je bilo potrebno za pridobivanje pravih informacij o Partnerju.

2 ISKRA DELTA PARTNER

Namizni računalnik Iskra Delta Partner je bil prvič predstavljen leta 1983 [11] in mišljen kot mali poslovni oz. razvojni sistem [1]. Pozneje so nastali modeli, ki so lahko prikazovali tudi grafiko. Znani so vsaj naslednji trije modeli:

- 1F/G z enim disketnim pogonom,
- 2F/G z dvema disketnima pogonoma in
- WF/G s trdim diskom in enim disketnim pogonom.

Zadnji, ki smo ga imeli tudi na voljo za potrebe razvoja emulatorja, je prikazan na sliki 1. Sestavljen je iz enote, ki vključuje zaslon, trdi disk, disketni pogon in sam računalnik, tipkovnica pa je ločena in z osnovno enoto povezana s kablom.

V nadaljevanju tega razdelka na kratko opišemo osnovne lastnosti računalnika Partner, pri čemer smo številne od njih ugotovili preko obratnega inženiringa.

2.1 Procesor in pomnilnik

V računalnik Partner je vgrajen 8-bitni mikroprocesor Z80A podjetja Zilog, ki je bil predstavljen leta 1976 [12]. Poganja ga ura s frekvenco 4 MHz [13].

Nekateri 8-bitni registri so združeni v pare, da sestavljajo 16-bitne različice, tako so na voljo naslednji registrski pari: AF, BC, DE in HL, pri čemer je A akumulator, F zastavice, preostali B, C, D, E, H in L pa so splošnonamenski registri. Obstajajo še 16-bitni registri: programski števec PC, skladovni kazalec SP in indeksna registra IX in IY.

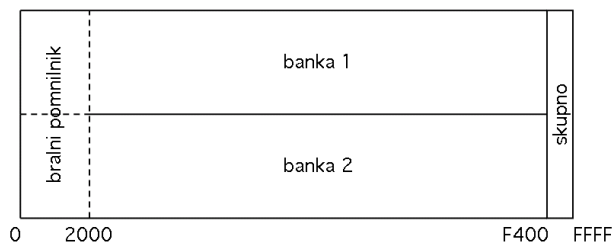


Slika 1: Iskra Delta Partner WF/G, na zaslonu je prikazan program URASET

Procesor podpira različne odzive na prekinitve, pri čemer Partner uporablja naslednji način. Najprej shrani PC na sklad, nato s podatkovnega vodila prebere 8-bitni podatek (*data*) in iz njega sestavi 16-bitni pomnilniški naslov, kjer zgornjih 8 bitov dobi iz posebnega registra I (gre za bazni register prekinitvenih vektorjev), zatem prebere 16-bitno besedo s tako sestavljenega naslova; ta beseda je naslov *isrAddr* prekinitveno servisne rutine naprave, t.j.

$$isrAddr = M_{16}[2^8 \cdot I + data].$$

Procesor podpira dva ločena naslovna prostora: 16-bitni pomnilniški naslovni prostor (prikazuje ga slika 2), ki ga zasedata delovni pomnilnik (RAM), in bralni pomnilnik (ROM), ter 8-bitni vhodno-izhodni naslovni prostor, ki ga zasedajo naprave. Računalnik Partner ima 125 kB delovnega pomnilnika, procesor Z80A pa lahko naslavlja le 64 kB, zato je dostop do celotnega pomnilnika urejen preko dveh t. i. *bank*. Vsaka banka je velika 61 kB, poleg tega pa je še 3 kB skupnega stalnega pomnilnika; skupaj tako dobimo $2 \cdot 61 \text{ kB} + 3 \text{ kB} = 125 \text{ kB}$. Prva banka in skupni del pomnilnika sta rezervirana za operacijski sistem, druga banka pa je namenjena uporabniški programski opremi.



Slika 2: Organizacija pomnilnika v Iskra Delta Partnerju

ROM ob zagonu računalnika zaseda spodnjih 8 kB. Njegova glavna naloga je nalaganje operacijskega sistema, ki ga nato izključi in prostor zasedeta banki.

2.2 V/I naprave

Računalnik Partner ima vgrajen disketni pogon za 5,25-palčne diskete, pri čemer je format dvostranski, na

vsaki strani je 73 sledi, v vsaki sledi je 18 sektorjev in v vsakem sektorju je 256 bajtov. Skupna kapaciteta diskete je tako 657 kB, sektorji pa so prepleteni v razmerju 2:1, vendar BIOS (angl. basic input/output system) to prikrije. Za branje teh disket na sodobnem računalniku je treba uporabiti poseben krmilnik, npr. Kryoflux [14].

Model Partner WF/G vsebuje tudi trdi disk Seagate ST-412 s kapaciteto skoraj 10 MB [13], pri čemer ima ta 1224 sledi, vsaka sled vsebuje 32 sektorjev in vsak sektor 256 bajtov.

2.3 Operacijski sistem

Partner za svoj operacijski sistem uporablja CP/M (angl. Control Program for Microcomputers), ki ga je razvilo ameriško podjetje Digital Research. To je enoopravilni diskovni operacijski sistem za mikroročunalnike s procesorjem Intel 8080 ali združljivim, kar velja za Zilogov Z80A. CP/M programom nudi osnovne storitve za delo z datotekami in osnovne funkcije za vhodno-izhodne naprave [15].

Operacijski sistem je sestavljen iz več delov. Prvi del je BIOS, ki se razlikuje glede na računalnik in ga priskrbi izdelovalec računalnika. Partnerjev BIOS poleg zahtevanih funkcij za CP/M vsebuje še funkcije za risanje grafičnih primitivov [13]. Njegove funkcije kliče drugi del, ki se imenuje BDOS (angl. basic disk operating system) in je neodvisen od strojne opreme ter implementira višjenivojske funkcije, ki jih kličejo uporabniški programi. Kot tretji del CP/M vsebuje konzolski ukazni interpreter CCP (angl. console command processor), ki podpira zaganjanje programov, delo z datotekami in druge sistemske programe.

3 EMULATOR

3.1 Zgradba emulatorja

Kot v pravem računalniku tudi v emulatorju navidezna centralna procesna enota (CPE) nadzira in usklajuje delovanje preostalih naprav. Emulator na najvišjem nivoju izvaja ukaze v emuliranem procesorju. Ko ta zahteva vhodno-izhodno operacijo, jo izvede in nadaljuje.

Naprave, s katerimi dela človek (tipkovnica, miška, zaslon), zahtevajo posebno obravnavo. Na pravem Partnerju se slika na zaslonu osvežuje s frekvenco 50 Hz sočasno z delovanjem procesorja, v emulatorju pa se zaradi enostavnosti emulacija procesorja in izrisovanje izvajata izmenično.

Natančneje, v vsaki iteraciji glavne zanke se izvede $4 \text{ MHz}/50 \text{ Hz} = 80000$ procesorskih ciklov, nato se izriše zaslonska slika, če se je spremenila. Sprožijo se tudi morebitne čakajoče prekinitve. Potem emulator čaka 20 ms ($1/50$ s) ali manj (odvisno od tega, kako dolgo je trajalo izrisovanje slike) in zanka se ponovi.

Tako emulirani procesor doseže hitrost, ki je približno enaka kot v pravem Partnerju (morebitne zakasnitve, ki jih zahteva DRAM (angl. dynamic RAM), se ne

upoštevajo). Hkrati emulator preverja, ali je bila pritisnjena tipka na tipkovnici, ali je bil pritisnjen ali spuščен gumb na miški in ali je bila miška premaknjena. Te dogodke posreduje ustreznemu modulu, ki ustrezno posodobi stanje naprave, ki bo programski opremi vidno ob naslednjem branju, in v primeru tipkovnice sproži prekinitve, da obvesti programsko opremo o pritisku tipke.

3.2 Centralna procesna enota

Za emulacijo CPE smo uporabili knjižnico Zymosis [16]. Uporaba knjižnice je tako rekoč poljubna, ker je avtor knjižnice deklariral kot javno dobro (angl. public domain). Knjižnica natančno implementira emulacijo procesorja Z80, vključno z nedokumentiranimi inštrukcijami in drugimi podrobnostmi. Za večjo prenosljivost smo knjižnico še nekoliko prilagodili.

Emulirani procesor se z napravami povezuje preko mehanizma povratnih klicev (angl. callback). To je mehanizem, kjer knjižnici podamo neki del kode (npr. preko naslova funkcije), ki se nato izvede ob vnaprej predvideni situaciji. Takšna situacija nastopi, ko npr. emulirani procesor potrebuje dostop do pomnilnika ali V/I naprave, zato pokliče ustrezno funkcijo za izvedbo bralne oz. pisalne operacije. Tako za pomnilnik (RAM in ROM) skrbi en par funkcij, za naprave pa se bralno-pisalna zahteva preusmeri na ustrezni modul, ki skrbi za napravo.

Povezavo naprav (vključno z glavnim pomnilnikom) s CPE izvedemo tako, da za vsako napravo implementiramo vsaj naslednje funkcije:

- Ponastavitev naprave: Funkcijo za ponastavitev naprave knjižnica pokliče ob začetku emulacije in vsakič ob resetiranju procesorja.
- Branje podatkov: Funkcija kot argument prejme naslov vrat, prebere vrednost in jo vrne.
- Pisanje podatkov: Funkcija prejme naslov vrat in vrednost, ki jo nato zapiše v napravo. Ta nato glede na zapisano vrednost izvede ustrezno operacijo.

Naprave lahko tudi sprožajo prekinitve, vendar ker je vsaka naprava drugačne narave, za to ni enotnega mehanizma. V nadaljevanju opišemo izvedbo emulacije nekaterih pomembnejših naprav.

3.3 Pomnilnik

Izvedba pomnilnika kot navidezne naprave je morda najbolj preprosta. V emulatorju je pomnilnik (tako RAM kot ROM) le polje, do katerega emulirani procesor dostopa preko povratnih klicev. Ob vsakem dostopu se pomnilniški naslov prevede iz naslovnega prostora emuliranega procesorja v naslovni prostor gostitelja, kjer se nato izvede še bralna oz. pisalna operacija.

Naslovni prostor Partnerja je grafično predstavljen na sliki 2. Polje, ki predstavlja pomnilnik, je treba na začetku emulacije inicializirati, tako da se v začetek polja naloži vnaprej pripravljena slika bralnega pomnilnika.

3.4 Tekstovni prikaz

Za krmiljenje zaslona in tekstovni prikaz v Partnerju skrbi napredni zaslonski krmilnik (angl. advanced video display controller, AVDC), ki temelji na čipu Signetics SCN2674 [17]. Čip lahko deluje na različne načine, v Partnerju pa se uporablja t. i. samostojni način, kjer ima AVDC svoj pomnilnik, ki ni neposredno dostopen, temveč ga procesor upravlja s pošiljanjem posebnih ukazov.

Pomnilnik je sestavljen iz dveh delov velikosti 4 kB, pri čemer prvi del vsebuje kode znakov, drugi pa njihove attribute [13]. Primer nekaterih podprtih ukazov:

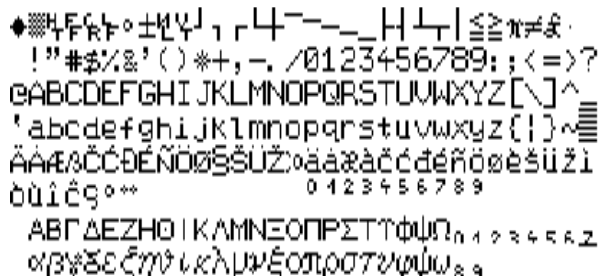
- izbira inicializacijskega registra,
- vklop oz. izklop kurzorja,
- vklop oz. izklop prekinitve,
- zapis znaka in atributov na naslov kurzorja,
- premik kurzorja na naslednji naslov,
- zapis več enakih znakov in atributov.

Več podrobnosti o uporabi čipa v računalniku Partner je predstavljenih v [18].

Kot je razvidno iz [17], čip AVDC za generiranje slike uporablja še generator zaslonskih znakov (vsebuje bitne slike znakov) in krmilnik grafičnih atributov (nadzoruje utripanje, podčrtavanje, osvetlitev in barvo znakov), ki pa procesorju nista neposredno dostopna, zato ju ni treba posebej emulirati. Njune funkcije, pomembne za delovanje AVDC, zato izvedemo kar v istem modulu.

Tekstovna slika je sestavljena iz več vrstic, vsaka od njih pa naprej iz več vrstic pikslov. V tem vrstnem redu poteka tudi izris slike v čipu AVDC, zato se vsaka vrstica v njegovem pomnilniku prebere večkrat. Emulator pa nima neposrednega dostopa do strojne opreme gostitelja, zato sliko najprej v celoti izriše v vmesni pomnilnik in jo nato posreduje na gostiteljev zaslon.

Partner podpira trinajst naborov znakov: poleg jugoslovskega in ameriškega nabora še nekatere druge (od teh so trije nedokumentirani). Žal procesor do slik znakov ne more dostopati, zato smo jih za potrebe emulatorja preprosto prerisali.



Slika 3: Nabor znakov generatorja zaslonskih znakov v 80-stolpcičnem načinu

Primer nabora znakov je na sliki 3. Partner podpira 80- in 132-stolpcični način prikaza, pri čemer piksele v prvem načinu horizontalno odebeli. Oba načina imata 26 vrstic. Emulator čipa sliko izriše po znakih, pri tem

se za vsak znak glede na njegove attribute določi barva ospredja (barva znaka) in barva ozadja, nato se znak (oz. njegova ustrežna polovica ali četrtina, če gre za dvojno širino ali višino) izriše na ustreznem mestu. Pri tem se tudi ustrezno skalira in odebeli, če je treba.

Čip AVDC lahko tudi sproži različne prekinitve, pri čemer Partnerjev BIOS omogoči le prekinitve ob intervalu navpičnega vračanja (angl. vertical blanking interval). Posledično smo se odločili le za emulacijo te prekinitve, preostalih pa ne (npr. prekinitve, ki se proži ob začetku izrisa prve vrstice pikslov vsake vrstice).

3.5 Grafični prikaz

Grafična različica Partnerja poleg čipa AVDC vsebuje še grafični procesor (angl. graphic display processor, GDP) Thomson EF9367 [19].

GDP ima svoj pomnilnik, velik 128 kB, ki vsebuje dve t. i. strani, katerih ločljivost je lahko 1024×512 ali 1024×256 pikslov, pri čemer vsak piksel zavzame en bit. Medtem ko programska oprema riše na eno stran, lahko GDP prikazuje drugo (ali isto) stran. Grafika na zaslonu prekrije besedilo, ki ga prikazuje AVDC.

Tudi do grafičnega pomnilnika CPE nima neposrednega dostopa. Bralne in pisalne operacije sproža s pošiljanjem ukazov, npr.:

- risanje črt z Bresenhamovim algoritmom v različnih slogih,
- risanje besedila v pisavi, vgrajeni v ROM čipa GDP (različna od pisave AVDC), pokončno ali ležeče, vodoravno ali navpično, v poljubni velikosti,
- brisanje celotne slike,
- določanje, ali se pikseli "rišejo" (prižigajo) ali "brišejo" (ugašajo),
- branje vrednosti piksla.

Več podrobnosti o čipu in njegovi rabi v Partnerju je obrazloženih v [18].

Emulator ukaze za GDP izvaja tako, da spreminja sliko v polju, kjer vsak piksel zaseda en bit. Ko je treba vsebino zaslona osvežiti, se vsebina vidne strani izriše; za prižgane piksele to pomeni, da se svetlosti tistega piksla na zaslonu prišteje neka konstanta. Tako se slika združi s tisto, ki jo je prej narisal AVDC.

3.6 Združitev tekstovnega in grafičnega prikaza

Sliki, ki ju generirata AVDC in GDP, sta neodvisni. Emulator mora za prikaz sliko celotnega zaslona izrisati v polje kvadratnih pikslov enakih velikosti. Na vprašanje, kaksno ločljivost slike in gostoto pikslov uporablja Partner, ni enotnega odgovora, saj:

- AVDC generira sliko 80×26 ali 132×26 znakov. Vsak znak je velik 8×11 pikslov. To je skupaj 640×286 oz. 1056×286 pikslov, pri čemer so v prvem primeru pikseli še dodatno raztegnjeni v širino.
- GDP generira sliko velikosti 1024×256 ali 1024×512 pikslov; le v drugem primeru so pikseli kvadratni (gostota je približno 128 dpi v obe dimenziji).

Maksimum obeh dimenzij je 1056×512. V resnici pa je slika čipa AVDC nekoliko večja od slike čipa GDP (približno za višino ene vrstice besedila v vsako smer). Treba jo je torej skalirati, vendar dovolj hitro in tako, da je besedilo še vedno dovolj estetsko, saj je treba sliko izrisati do 50-krat na sekundo. Za sliko čipa GDP to ni problem, ker so pikslji ali kvadratni ali pa podvojeni v višino.

Odločili smo se za ločljivost 1056×572 oz. 1024×572, če je zaslon gostitelja ožji od 1056 pikslov. Pri tem velja:

- Višina slike AVDC se v vsakem primeru podvoji (zato višina 572 pikslov).
- Če AVDC prikazuje 80 stolpcev, se vsak drugi piksel podvoji v širino; slika AVDC je potem široka 960 pikslov, (kar je manj kot slika GDP, vendar videz večine aplikacij zaradi tega ne trpi) in je vodoravno centrirana.
- Če AVDC prikazuje 132 stolpcev in je zaslon ožji od 1056 pikslov, je vsak četrti znak širok samo 7 pikslov namesto 8. Slika AVDC je potem široka 1023 pikslov.
- Slika GDP se podvoji v višino, če je izbrana ločljivost 1024×256, drugače pa se ne skalira. V vsakem primeru se navpično centrirata.
- Če je zaslon širši od 1024 pikslov, se slika GDP vodoravno centrirata.

3.7 Serijski vmesnik

Partner ima lahko ena ali troja serijska vrata, ki ustrezajo standardu RS-232-C in uporabljajo konektor DB-25 [13]. Zanje skrbita dva krmilnika, od katerih ima vsak dva kanala, pri čemer vsak kanal zaseda dva vhodno-izhodna naslova [13], [20]. Prvi je podatkovni naslov, s pisanjem na katerega se sproži pošiljanje, če je krmilnik na to pripravljen, z branjem pa se dobi čakajoči podatek, če je na voljo. Drugi je naslov, ki se uporablja za branje statusnih informacij (npr. ali je prejel podatek, ki ga mora aplikacija prebrati) in spreminjanje nastavitvev (npr. hitrosti komuniciranja). Prejemanje in pošiljanje podatkov poteka sinhrono. Emulator samih vrat posebej ne emulira, emulira pa nekatere naprave, priključene nanje.

3.8 Tipkovnica

Partnerjevo tipkovnico (prikazana je na sliki 4) je izdelalo podjetje Gorenje in se priključi na serijska vrata, vendar s konektorjem DIN 5. Z računalnikom komunicira s hitrostjo 300 bitov na sekundo [13].

Tipkovnica ima poleg alfanumeričnih tipk in numeričnega dela še nekaj posebnih tipk, ki jih danes standardne tipkovnice nimajo. Emulator jih zato preslika na ustrezne tipke na čim bolj podobnih mestih.

Kode, ki jih generirajo tipke, so v večini primerov enake kodam znakov ASCII (angl. American standard code for information interchange), ki jih predstavljajo. Pri tem posebne tipke, kot so SHIFT, CTL in CAPS, vplivajo na generirane kode, same pa kod ne generirajo.



Slika 4: Partnerjeva tipkovnica

Posledično računalnik neposredno ne zazna teh tipk in med njimi tudi ne loči, npr. med kombinacijo SHIFT+A in tipko A v načinu velikih črk (CAPS).

Nadalje je računalnik s prekinitvijo obveščen le o pritisku (vendar ne spustu) tipke, če pa je tipka pritisnjena dalj časa, se ustrezna koda pošilja ponavljajoče.

Tipkovnica ima tudi osem lučk in zvočnik, ki se lahko oglašata z dolgim, kratkim ali zelo kratkim piskom, kjer se slednji opcijsko proži ob pritisku tipk. Oboje je mogoče upravljati programsko s pošiljanjem 8-bitnih vrednosti, pri čemer nekatere vrednosti spreminjajo način delovanja tipkovnice, npr.:

- ali tipke ob pritisku povzročijo pisk,
- ali daljši pritisk tipke sproži ponavljanje kode,
- ali je v rabi jugoslovanski (QWERTZ) ali ameriški (QWERTY) razpored tipk.

Uporabnik do teh in tudi drugih nastavitvev pride preko programa, vgrajenega v BIOS, ki ga priključi s tipko SET UP.

Med razvojem smo ugotovili, da obstajata vsaj dva različna modela tipkovnice, saj se je tista, ki smo jo imeli na voljo, odzivala na drugačne ukaze, kot jih je pošiljal BIOS. Zaradi boljše uporabniške izkušnje smo emulator prilagodili modelu, kot ga predvideva BIOS. Večina aplikacij tipkovnico le bere (preko funkcij BDOS), zato razlika v modelih ne pomeni velike težave.

Ko emulator zazna pritisk prave tipke, jo preslika v ustrezno kodo in sproži prekinitvev serijskega vmesnika ter pri tem predvaja še ustrezen pisk, če je ta opcija vklopljena. BIOS se potem odzove z branjem podatkovnega registra serijskih vrat, preko katerega mu emulator poda preslikano kodo tipke. Ob daljšem pritisku tipke in vklopljenem ponavljanju tipk emulator nato še naprej periodično sproža prekinitve, dokler tipka ni spuščena.

3.9 Disketni pogon in trdi disk

Partnerjev ROM omogoča zaganjanje operacijskega sistema tako z disketnega pogona kot s trdega diska, ki ga ima le model WF/G. Tega upravlja krmilnik Xebec S1410, o krmilniku prvega pa skorajda nismo imeli nobenih informacij.

Zato in tudi iz želje po čim prejšnjem delovanju CP/M smo se odločili za emulacijo na višjem nivoju, tako imenovano virtualizacijo na nivoju gonilnika [4], kjer

smo v emulatorju implementirali prestrezanje diskovnih funkcij. Ta pristop bi se dalo uporabiti tudi za druge naprave, vendar smo se ga zaradi želje po čim boljši emulaciji izogibali.

Na srečo CP/M vedno kliče BIOS skozi dokumentirane vstopne točke, zato emulator pred izvedbo vsakega strojnega ukaza preveri vrednost v PC in če se ujema s katero od vstopnih točk za diskovne funkcije, emulator izvede želeno operacijo in vrne ukaz RET, ki povzroči, da se prava funkcija ne izvede. Natančneje, emulator to stori za funkcije HOME, SELDSK, SETTRK, SETSEC, SETDMA, READ, WRITE in SETBNK.

Ta metoda dobro deluje, ker je CP/M edini operacijski sistem za Partnerja in ker večina aplikacij ne dostopa do teh naprav neposredno, ampak preko operacijskega sistema. Nam znani izjemi sta predvsem programa WF in DISKETTE, ki sta namenjena prav temu, da preverita delovanje trdega diska in njegovega krmilnika oz. trenutno vstavljene diskete, in program FORMAT, ki formatira diskete. Tudi ROM do teh naprav dostopa neposredno (saj drugače ne more začeti nalaganja operacijskega sistema), zato emulator prestreže tudi nekatere njegove funkcije.

3.10 Ura realnega časa

Poznavanje datuma in ure Partnerju omogoča ura realnega časa, ki jo napaja baterija. Procesorju je dostopna kot množica registrov, ki vsebujejo časovne komponente (tisočinke, sekunde, minute, ure, dan, mesec, zadnji dve števki leta) v obliki "packed BCD". Vsak register ima svoj V/I naslov, ki je namenjen za branje in pisanje.

Emulator dobi datum in čas od gostiteljevega operacijskega sistema. Ker Partnerjev BIOS in CP/M ne podpirata datumov po letu 1999 (zaradi hroščev se ne izpišejo pravilno; to bi se sicer dalo popraviti), emulator od trenutnega leta odšteje večkratnik 28, da je leto manjše od 2000 (npr. 2018 – 28 = 1990). Gregorijanski koledar se namreč ponovi na vsakih 28 let; npr. 1. julij je bila sobota tako leta 2017 kot 1989, 31. julij pa ponedeljek (ta rešitev deluje do 28. februarja 2100).

K uri realnega časa spada tudi majhen pomnilnik, namenjen za shranjevanje uporabniških nastavitvev. Emulator vsebino tega pomnilnika ob zagonu prebere iz datoteke (oz. uporabi privzete vrednosti), ob izhodu pa jo shrani, če so se nastavitve spremenile.

4 RAZPRAVA

Preden zaključimo, podajmo še kratko ovrednotenje naše rešitve v smislu smernic ovrednotenja emulacije kot metode digitalnega ohranjanja [21], pri čemer se osredinimo predvsem na natančnost emulacije, prenosljivost emulatorja in na uporabniško izkušnjo.

Naš glavni cilj pri razvoju je bila kar se da natančna emulacija. Kljub temu se delovanje emulatorja v nekaterih podrobnostih razlikuje od delovanja pravega Partnerja.

Večina razlik se navezuje na sočasnost delovanja pravih naprav, ki so v emulatorju izvedene zaporedno. To se odraža pri emulaciji AVDC in GDP, kjer se spremembe v njunih registrih oz. pomnilniku uveljavijo le periodično namesto v realnem času, saj emulator ne poustvari delovanja zaslona s katodno cevjo. Druga stvar, povezana s časom, so razne zakasnitve strojne opreme kot posledica njenega sočasnega delovanja. Primer je pomnilnik (DRAM), do katerega je dostop v emulatorju z vidika gosta takojšen.

Prenosljivost emulatorja smo dosegli z uporabo programskega jezika C in knjižnice Simple DirectMedia Layer [22], ki priskrbi enoten vmesnik za upravljanje V/I naprav na številnih operacijskih sistemih (preizkušeno na Windows, Mac OS, OS/2, Haiku). Preko okolja Emcripten [23], ki kodo iz jezika C prevede v JavaScript, pa je emulator mogoče (omejeno, brez zvočnih učinkov) izvajati tudi v spletnem brskalniku.

Uporabniško izkušnjo emulatorja pa smo nadgradili z uporabo zvočnih posnetkov trdega diska, tipkovnice in stikal za vklop oz. izklop in resetiranje. Tudi slika na zaslonu je takoj po "vklopu" črna, šele nato doseže polno svetlost, medtem pa se predvaja zvok zagona trdega diska.

Med Partnerjevo programsko opremo, ki zahteva natančnejše delovanje emulatorja, najdemo tudi grafično aplikacijo VIGRED, katere zaslonska slika je prikazana na sliki 5.



Slika 5: Zaslonska slika aplikacije VIGRED v emulatorju

5 SKLEP

Na začetku projekta zaradi težav z disketnim pogonom in serijskimi vrati ter pomanjkanjem dokumentacije nismo vedeli, ali nam bo uspelo narediti delujoč emulator. Težave smo uspešno prebrodili in z nekaj znanja, sreče ter obratnega inženiringa nam je vseeno uspelo narediti emulator, ki se z vidika uporabnika obnaša skoraj enako kot pravi Partner. Večina zbrane dokumentacije, kakor tudi končna izvedba emulatorja, je na voljo v okviru spletne strani <http://matejhorvat.si/sl/slorac/delta/partner/>.

Projekt je trajal okrog štiri mesece, sama programska koda (brez knjižnic) obsega okrog 3500 vrstic. Poleg

tega smo napisali še okrog 1000 vrstic dodatne kode v različnih jezikih, npr. kodo za izdelavo bitne slike trdega diska in bralnega pomnilnika itd.

Emulatorju bi lahko dodali še emulacijo nekaterih naprav, kot so npr. tiskalnik, risalnik in grafična tablica, ter tudi funkcije, ki bi izboljšale uporabniško izkušnjo, kot so npr. integracija z datotečnim sistemom gostitelja, skupno odložišče med emulatorjem in gostiteljem itd.

Za konec omenimo še, da trenutno na Fakulteti za računalništvo in informatiko Univerze v Ljubljani izvajamo študentski projekt, s katerim skušam pridobiti dokumentacijo in druge vire še o drugih starodobnih slovenskih računalnikih, ki pa so bili morda manj uporabljani in izdelani v še manjših količinah kot Partner. Na podlagi teh rezultatov upamo, da v prihodnosti lahko izdelamo še kakšen emulator slovenskega starodobnega računalnika.

ZAHVALA

Zahvaljujemo se Janezu Kožuhu, ki je muzeju Fakultete za računalništvo in informatiko Univerze v Ljubljani doniral delujoč računalnik Iskra Delta Partner, in izr. prof. dr. Veselku Guštinu za vso pomoč in trud, ki ga vlaga v fakultetni računalniški muzej. Za pomoč pri izdelavi bitnih slik Partnerjevih disket in iskanju ustreznih disketnih pogonov se zahvaljujemo Tadeju Pečarju in doc. dr. Tomažu Dobravcu. Za fotografiji Partnerja in razne nasvete se zahvaljujemo dr. Gašperju Fele Žoržu in Službi za komuniciranje FRI za izposojlo fotoaparata.

LITERATURA

- [1] A. Vilfan, J. Vilfan. "Iskra Delta med zahodom in vzhodom", *Bit* 1, 1984.
- [2] David S. H. Rosenthal. *Emulation & Virtualization as Preservation Strategies*, University of North Texas Libraries, Digital Library, 2015.
- [3] Stephan Strodl, Christoph Becker, Robert Neumayer, Andreas Rauber. *How to Choose a Digital Preservation Strategy: Evaluating a Preservation Planning Procedure*, Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries, Vancouver, BC, Canada, 2007.
- [4] J. E. Smith, R. Nair. *Virtual Machines – Versatile Platforms for Systems and Processes*. Morgan Kaufmann Publishers, 2005.
- [5] Jurij Mihelič, Tomaž Dobravec. "SicSim: A simulator of the educational SIC/XE computer for a system-software course", *Computer Applications in Engineering Education* 23 (1), 2015.
- [6] Klemen Kloboves, Jurij Mihelič, Patricio Bulić, Tomaž Dobravec. "FPGA-Based SIC/XE Processor and Supporting Tool-chain", *The International Journal of Engineering Education* 33 (6A), 2017.
- [7] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt and B. Werner. "Simics: A full system simulation platform", *Computer* 35 (2), 2002.
- [8] Raymond A. Lorie. "Long Term Preservation of Digital Information", *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries*, Roanoke, Virginia, USA, 2001.
- [9] Jeff Rothenberg. "Ensuring the Longevity of Digital Documents", *Scientific American* 272 (1), 1995.
- [10] Jeff Rothenberg. *Using Emulation to Preserve Digital Documents*, 2000.
- [11] A. P. Železnikar. "Evropska mikroročunalniška industrija", *Informatika* 1, 1984.
- [12] J. C. Nichols, E. A. Nichols, P. R. Rony. *Z-80 Microprocessor Programming & Interfacing: Book 2*. Howard W. Sams & Co, 1980.
- [13] Iskra Delta. *Partner WFG, 2FG, 1FG – priručnik za uporabnike*. 1987.
- [14] *KryoFlux*. Dosegljivo: <https://www.kryoflux.com/>. [Dostopano: 9. 3. 2018]
- [15] Digital Research. *An introduction to CP/M features and facilities*. 1978.
- [16] Ketmar. *Zymosis*. Dosegljivo: <http://repo.or.cz/zymosis>. [Dostopano: 9. 3. 2018]
- [17] Signetics. *SCN2674/SCN2674T Advanced Video Display Controller (AVDC)*. Dosegljivo: <http://www.datasheet4u.com/datasheet-pdf/Signetics/SCN2674/pdf.php?id=524408>. [Dostopano: 9. 3. 2018]
- [18] Matej Horvat. *Emulacija računalnika Iskra Delta Partner*, diplomska naloga, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2017.
- [19] SGS-Thomson Microelectronics. *EF9367 MOS graphic display processor (GDP)*. Dosegljivo: <http://www.datasheet4u.com/datasheet-pdf/SGS-Thomson/EF9367/pdf.php?id=604475>. [Dostopano: 9. 3. 2018]
- [20] L. A. Leventhal. *Z80 assembly language programming*. Osborne/McGraw-Hill, 1979.
- [21] Mark Guttentbrunner and Andreas Rauber. "A Measurement Framework for Evaluating Emulators for Digital Preservation", *ACM Transactions on Information Systems* 30 (2), 2012.
- [22] Simple DirectMedia Layer. Dosegljivo: <https://www.libsdl.org/>. [Dostopano: 9. 3. 2018]
- [23] Emscripten. Dosegljivo: <https://kripken.github.io/emscripten-site/>. [Dostopano: 9. 3. 2018]

Matej Horvat je diplomiral leta 2017 na Fakulteti za računalništvo in informatiko v Ljubljani. Trenutno je magistrski študent študija računalništva istoimenske fakultete. Ukvarja se z vgrajenimi sistemi in zgodovino računalništva.

Jurij Mihelič deluje kot docent v okviru Laboratorija za Algoritmiko Fakultete za računalništvo in informatiko Univerze v Ljubljani, kjer je leta 2006 tudi doktoriral iz računalniških znanosti. Njegova raziskovalna področja zajemajo inženiring in optimizacijo algoritmov, kombinatorično optimizacijo ter sistemsko programsko opremo in virtualizacijo.