

Univerza v Ljubljani
Fakulteta za elektrotehniko

Tomaž Šolc

Replika mikroračunalnika Galaksija

Univerzitetno diplomsko delo

Mentor
prof. dr. Tadej Tuma

Ljubljana, 2007

Povzetek

Zaradi različnih dejavnikov je danes običajno, da je za starejše mikroprocesorske sisteme na voljo le zelo omejena dokumentacija. V takem primeru se pogosto uporabi reverzni inženiring, za uspešnost katerega pa je pomembno poznavanje načrtovalskih postopkov in prijemov iz obdobja nastanka sistema, ki ga raziskujemo.

Galaksija je jugoslovanski hišni mikroračunalnik iz 80-ih let 20. stoletja z zgodovinsko vrednostjo. Zahvaljujoč javni objavi se je do danes ohranil del dokumentacije o strojni opremi in obsežna zbirka programske opreme, vključno z operacijskim sistemom. Kot tipičen in hkrati relativno preprost mikroprocesorski sistem iz tega obdobja Galaksija omogoča enostaven vpogled v takratne načrtovalske prijeme.

V okviru te naloge je bila najprej na podlagi ohranjenih načrtov in reverznega inženiringa programske opreme izdelana dokumentacija strojne arhitekture. Na podlagi te pa je bila načrtana in izdelana replika Galaksije, to je elektronsko vezje iz sodobnih komponent, ki je iz programskega vidika kompatibilno z izvirnim mikroračunalnikom in na katerem se izvaja izvirni operacijski sistem.

Izdelava replike je bila osnova za dokumentiranje posebnosti v implementaciji izvirnega elektronskega vezja, delujoča Galaksija pa je služila tudi kot orodje pri raziskovanju programerskih prijemov avtorjev operacijskega sistema. Namen te dokumentacije je olajšati podobne projekte za obnovo mikroprocesorskih sistemov iz tega obdobja.

V okviru naloge je nastalo tudi programsko orodje za reverzni inženiring strojne kode za mikroprocesor Z80 ter razvojno okolje za Galaksijino arhitekturo.

Ključne besede: Galaksija, mikroračunalniki, mikroprocesorski sistemi, operacijski sistem, reverzni inženiring

Abstract

Due to various reasons a shortage or even absence of detailed documentation for older microprocessor systems is common today. Because of this we often have to resort to reverse engineering for which a knowledge of practices and patterns used at the time of design of the system in question is of great importance.

Galaksija is a Yugoslavian home microcomputer from the 1980s with some historical value. Because of publication in magazines a part of hardware documentation and a large collection of software, including the operating system, has been preserved to this day. As a typical and at the same time relatively simple microprocessor system from that time Galaksija offers an insight into the design practices in the 1980s.

The first part of this thesis includes the study of the preserved schematics and reverse engineering of the software in order to construct a complete documentation of Galaksija's hardware architecture. This documentation is then used to design and construct a replica of Galaksija, that is a modern electronic circuit that is software compatible with the original microcomputer and which is capable of running the original operating system.

Design of the replica was the basis for the second part of this thesis which documents unusual and interesting hardware design considerations. The working replica also served as a tool for research into original operating system design. The purpose of this part is to aid any future projects for redesigning hardware from this time period.

A number of software tools were created during this work, including a Z80 disassembler and a software developers kit for Galaksija architecture.

Keywords: Galaksija, microcomputers, microprocessor systems, operating system, reverse engineering

Zahvala

Mentorju prof. dr. Tadeju Tumi bi se rad zahvalil za podporo pri izdelavi te naloge.

Zahvaljujem se staršema in sestri Barbari za podporo in pomoč v kritičnih trenutkih. Mami za znanstveni pogled na svet in očitju, ki mi je bil zgled pri študiju elektronike.

Za pomoč pri zbiranju dokumentacije in ohranjenih člankov o Galaksiji, izdelavo tiskanega vezja in nenazadnje tudi za zanimive pogovore o stari računalniški in elektronski opremi bi se rad zahvalil g. Jožetu Stepanu. Ravno tako se zahvaljujem Tomažu Kacu za njegovo zbirko ohranjene programske opreme za Galaksijo ter občasne namige pri raziskovanju delovanja Galaksije.

Prof. Mateji Cvelbar bi se rad zahvalil za lektoriranje tega dela.

Na koncu bi se rad zahvalil tudi ostalim članom ekipe Kiberpipinega računalniškega muzeja, ki mi je omogočal dostop do skritih delov starih računalnikov in katerega zanimanje za Galaksijo je bil pravzaprav povod za nastanek tega diplomskega dela.

Kazalo

1	Uvod	12
1.1	Galaksija	12
1.2	Motivacija	16
1.3	Cilji	17
2	Strojna arhitektura	18
2.1	Uporaba naslovnega prostora	18
2.2	Periferija	19
2.2.1	Tipkovnica	20
2.2.2	Vmesnik za magnetofon	20
2.2.3	Latch	21
2.3	Nadzor linije A7 za RAM	21
2.4	Generiranje video signala	22
2.4.1	Sinhronizacija	23
2.4.2	Generator znakov	25
2.4.3	Video gonilnik	28
2.4.4	Shift register	30
3	Nova Galaksija	32
3.1	Digitalni del	32
3.1.1	Mikroprocesor in pomnilnik	32
3.1.2	Naslovni dekodler	35
3.1.3	Tipkovnica	35
3.1.4	Delilnik ure	35
3.1.5	Shift register	36
3.1.6	Sinhronizacija prekinitev	36
3.2	Analogni del	38
3.2.1	Napajalnik	38
3.2.2	Oscilator	38
3.2.3	Reset vezje	38
3.2.4	Kompozitni video	38
3.2.5	Vmesnik za magnetofon	40
4	Posebnosti vezja originalne Galaksije	42
4.1	Uporaba nedokumentiranih lastnosti mikroprocesorja	42
4.1.1	Prva pomnilna celica	42
4.1.2	Druga pomnilna celica	43
4.2	Vezava mikroprocesorjevega vodila	43
4.3	Vezava tipkovnice	45

KAZALO

4.4	Uporaba R registra	45
5	Posebnosti operacijskega sistema	46
5.1	Večličnost programske kode	48
5.2	Večličnost podatkovnih struktur	50
5.3	Organizacija programske kode	52
5.4	Uporaba procesorskih registrov	52
5.5	Procesorski sklad	54
5.6	Shranjevanje podatkov na magnetofon	55
6	Zaključek	58
7	Literatura	60
8	Priloge	62
A	Napetostni inverter	63
B	Izračun impulznega ojačevalnika	65
C	Števila s plavajočo vejico	67

Slike

1	Ročno izdelana Galaksija	14
2	Tovarniško izdelana Galaksija	14
3	Zgornja stran tiskanine	15
4	Spodnja stran tiskanine	15
5	Blok shema	18
6	Uporaba naslovnega prostora mikroprocesorja Z80	19
7	Naslovi posameznih tipk in izhoda iz komparatorja	20
8	Oznake posameznih bitov v registru latch	22
9	Vpliv bita A7CLMP	23
10	Časovni okvirji za generiranje video signala	24
11	Časovni diagram delovanja mikroprocesorja takoj po prekinitvi	24
12	Zapis ASCII znaka A v ROM-u generatorja znakov	26
13	Tabela znakov, shranjenih v ROM-u generatorja znakov	27
14	Poenostavljen diagram izvajanja video gonilnika	29
15	Delovanje video shift registra	31
16	Matična plošča nove Galaksije	33
17	Tipkovnica nove Galaksije	33
18	Nova Galaksija v ohišju	34
19	Časovni diagram delovanja vezja za nadzor shift registra	37
20	Časovni diagram delovanja vezja za sinhronizacijo prekinitev	37
21	Bodejev diagrama video ojačevalnika	39
22	Izhodni signal vmesnika za magnetofon	41
23	Signal impulznega ojačevalnika	41
24	Časovni diagrama prve D pomnilne celice	43
25	Časovni diagram druge D pomnilne celice	44
26	Organizacija delovnega pomnilnika	47
27	Tipična razporeditev funkcij v operacijskem sistemu	53
28	Primer spreminjanja povratnega naslova funkcije na skladu	54
29	Primer uporabe sklada za prenos kazalca na argumente	55
30	Časovni diagram modulacije	56
31	Rezultat meritev napetosti pred in po stabilizatorju	64

TABELE

Tabele

1	Primerjava s tujimi mikroračunalniki	13
2	Primerjava z domačimi mikroračunalniki	13
3	Izhodna napetost analognega izhoda	21
4	Pomeni posameznih bitov v registru latch	21
5	Primerjava karakteristik video signala	36
6	Primer večličnosti strojne kode	49
7	Strojna koda kot ASCII niz	49
8	Strojna koda kot konstanta s plavajočo vejico	50
9	Primer večličnosti podatkovnih struktur	51
10	Tipična uporaba registrov v operacijskem sistemu	53
11	Vrednosti simbolov (k sliki 30)	56
12	Pomen bytov pri shranjevanju na magnetofon	57
13	Shranjevanje BASIC programa	57
14	5-bytni zapis s plavajočo vejico	68
15	4-bytni zapis s plavajočo vejico	68

Uporabljene oznake

0x2800	Za zapis števil v heksadecimalnem sistemu je uporabljena predpona 0x iz programskega jezika C. Vsi pomnilniški naslovi so zapisani v heksadecimalnem sistemu.
VIDEO_INT	Imena funkcij, spremenljivk in ostalih simbolov se nanašajo na dokumentirano izvorno kodo ROM-a A, ki se nahaja v prilogi.
U11	Oznake elektronskih komponent se v poglavju 2 nanašajo na originalni vezalni načrt Galaksije [2], v preostalem besedilu pa na vezalni načrt matične plošče, ki se nahaja v prilogi.
ld a,b	Strojna koda v zbirniku uporablja notacijo podjetja Zilog. Rezultat operacije se zapiše v prvi operand.

1 Uvod

1.1 Galaksija

Galaksija je hišni mikroračunalnik, ki ga je leta 1983 zasnoval Voja Antonić. Podobno kot ostali mikroračunalniki, zasnovani v tistem času na področju bivše Jugoslavije, je bila Galaksija zamišljena predvsem kot domača konkurenca za tuje mikroračunalnike podjetij Sinclair Research (Sinclair ZX81, Sinclair ZX Spectrum) in Commodore Business Machines (Commodore 64). Ti so bili v tistem času kljub povpraševanju za zasebnike težko dostopni legalno zaradi strogih omejitev uvoza zahodne tehnologije in tudi relativno visokih cen [1].

Da bi imel računalnik kar se da nizko ceno in tako bil dostopen najširšemu krogu ljudi, je bilo njegovo elektronsko vezje zasnovano kar najbolj preprosto in je vključevalo le dele, ki so bili takrat ali prosto dostopni v trgovinah z elektronskimi komponentami ali pa jih je bilo mogoče brez težav uvoziti iz sosednjih držav. Ker je imel ta cilj pri načrtovanju najvišjo prioriteto, so bili pri strojni opremi sprejeti precejšnji kompromisi. Delovanje računalnika je tako na primer odvisno od nekaterih nedokumentiranih lastnosti uporabljenega mikroprocesorja in podobnih razvojnih prijemov, ki bi bili pri načrtovanju sodobnih sistemov nesprejemljivi. Tudi razvoj sistemske programske opreme se je v veliki meri podrejal želji po nizki ceni strojne opreme.

Primerjava zmogljivosti Galaksije s podobnimi tujimi in domačimi mikroračunalniki iz tistega časa je prikazana v tabelah 1 in 2.

Računalnik je bil sprva namenjen samogradnji, zato je avtor ob pomoči urednika Dejana Ristanovića njegovo zgradbo v celoti opisal v seriji člankov v posebni izdaji revije Galaksija (po kateri je računalnik tudi dobil ime), ki je izšla januarja 1984 [2]. Predvsem zaradi objave velikega dela razvojne dokumentacije se je le-ta v uporabni obliki ohranila do danes. Zaradi velikega povpraševanja se je računalnik kasneje izdeloval tudi serijsko.

1.1 Galaksija

	ZX81 [3]	Galaksija [2]	ZX Spectrum [4]	Commodore 64 [5]
Leto	1981	1984	1982	1982
CPU	Zilog Z80A	Zilog Z80A	Zilog Z80A	MOS 6502
Ura	3250 kHz	3072 kHz	3500 kHz	985 kHz (PAL)
ROM	8 kB	4 kB (ROM A)	16 kB	20 kB
RAM	1 kB	2, 4 ali 6 kB	16 ali 48 kB	64 kB
Video	Programsko	Programsko	Lastni IC (ULA)	MOS VIC-II
Ločljivost	32x24 znakov 64x48 pik	32x16 znakov 64x48 pik	32x24 znakov 256192 pik	40x25 znakov 320x200 pik
Barve	Ne	Ne	15 barv	16 barv
Audio	Ne	Ne	Programsko	MOS SID

Tabela 1: Primerjava zmogljivosti Galaksije s tujimi mikroračunalniki

	Galaksija	Orao	Lola-8	Galeb
CPU	Zilog Z80A	MOS 6502	Intel 8085A	MOS 6502
Ura	3072 kHz	1000 kHz	5000 kHz	Ni podatka
ROM	4 kB (ROM A)	16 kB	16 kB	16 kB
RAM	2, 4 ali 6 kB	8 kB	6 kB	9 kB
Ločljivost	32x16 znakov 64x48 pik	32x16 znakov 256x128 pik	40x25 znakov 80x75 pik	48x16 znakov 96x48 pik
Barve	Ne	Ne	Ne	Ne
Audio	Ne	Da	Da	Da

Tabela 2: Primerjava zmogljivosti Galaksije z domačimi mikroračunalniki [1]

1 UVOD



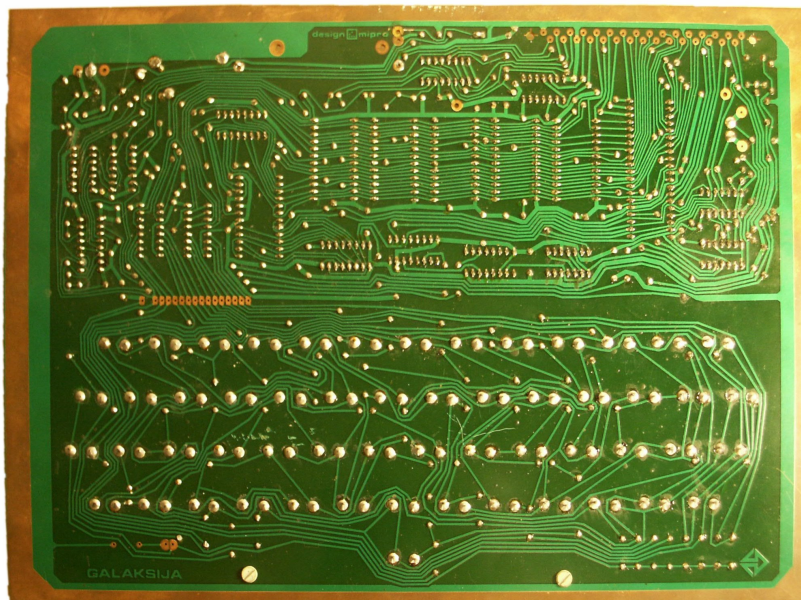
Slika 1: Ročno izdelana Galaksija (zasebna zbirka g. Damjana Lenarčiča)



Slika 2: Galaksija, kot so jo izdelovali v *Zavodu za udžbenike i nastavna sredstva*



Slika 3: Zgornja stran tiskanine originalne Galaksije (zasebna zbirka g. Iztoka Pušnarja)



Slika 4: Spodnja stran tiskanine originalne Galaksije (zasebna zbirka g. Iztoka Pušnarja)

1.2 Motivacija

Danes se zaradi izredno hitrega razvoja računalniških sistemov vedno pogostejše dogaja, da se znanje, potrebno za vzdrževanje starejših izdelkov, izgubi. Še posebno je ta pojav očiten pri potrošniški elektroniki, kjer lahko določen izdelek zastara in popolnoma izgine iz javno dostopnih arhivov dokumentacije proizvajalca že v manj kot letu dni. Pogosta združevanja in reorganizacije podjetij situacijo le še poslabšajo, saj pri iskanju arhivske dokumentacije pomenijo dodatne korake.

Po drugi strani pa potrošniška elektronika s svojo nizko ceno in široko dostopnostjo danes na vedno več področjih izriva dosti dražje namenske industrijske naprave. Največ primerov lahko najdemo ravno na področju osebnih računalnikov, kjer masovna proizvodnja omogoča proizvajalcem velika vlaganja v razvoj. Posledica tega je na primer, da so zmogljivosti arhitekture, prvotno namenjenih manj zmogljivim osebnim računalnikom, prehiteli določene namenske arhitekture.

V praksi se pogosto pojavi problem, ko ima programska oprema dosti višjo vrednost kot strojna oprema, na kateri teče. Take primere lahko na primer najdemo pri visoko zanesljivih sistemih, kjer je programska oprema z leti uporabe in popravkov postala dosti bolj zanesljiva kot oprema, napisana na novo. V nasprotju s programsko opremo, ki se v strogem smislu ne stara in obrablja, pa strojna oprema potrebuje občasno vzdrževanje. Ko stroški takega vzdrževanja narastejo do neke meje, je smiselno izdelati novo, združljivo strojno opremo. Pri tem zaradi zgoraj omenjenega pojava velikokrat ni na voljo potrebne dokumentacije za originalno strojno opremo. V tem primeru je potrebno to dokumentacijo izdelati na novo s pomočjo reverznega inženiringa.

V mikroračunalniku Galaksija lahko najdemo tipične prijeme razvijalcev strojne in programske opreme, ki so bili v uporabi v obdobju 80-ih let 20. stoletja. Medtem ko za samo Galaksijo niso znani primeri, da bi se uporabljala za profesionalne namene, pa so izkušnje, pridobljene pri njenem reverznem inženiringu, uporabne pri podobnih projektih, ki imajo večjo uporabno vrednost.

Pri izdelavi replike Galaksije tudi ne moremo zanemariti pomena ohranjanja tehnične dediščine. Zaradi razvoja mikroelektronike danes integriranih vezij, ki bi jih potrebovali za izgradnjo nove originalne Galaksije, ni več na trgu. Tako je za poganjanje arhivske programske opreme na voljo le še majhno število ohranjenih računalnikov v delujočem stanju. Galaksija in programska oprema zanjo je pomembna iz zgodovinskega vidika, saj je ve-

liko prispevala k širjenju znanja računalništva in informatike na področju takratne Jugoslavije.

V poučevanju osnov mikroprocesorskih sistemov se danes prav tako pogosto pojavlja težava, da so sodobni sistemi preprosto prezapleteni, da bi njihovo delovanje lahko uporabili kot primer za razlago osnov delovanja. Sodobni osebni računalnik je na primer tako zapleten, da je za študenta praktično nemogoče, da bi v celoti razumel delovanje vseh njegovih delov. Zaradi strahu pred konkurenco pa proizvajalci dostikrat celo namenoma uporabljajo prijeme, ki otežujejo spremljanje in razumevanje dogajanja znotraj naprav. Težave povzroča tudi visoka stopnja integracije sodobne elektronike, ki onemogoča vpogled v delovanje vezij z merilnimi instrumenti, ki so na razpolago v tipičnem laboratoriju za delo z elektronskimi vezji.

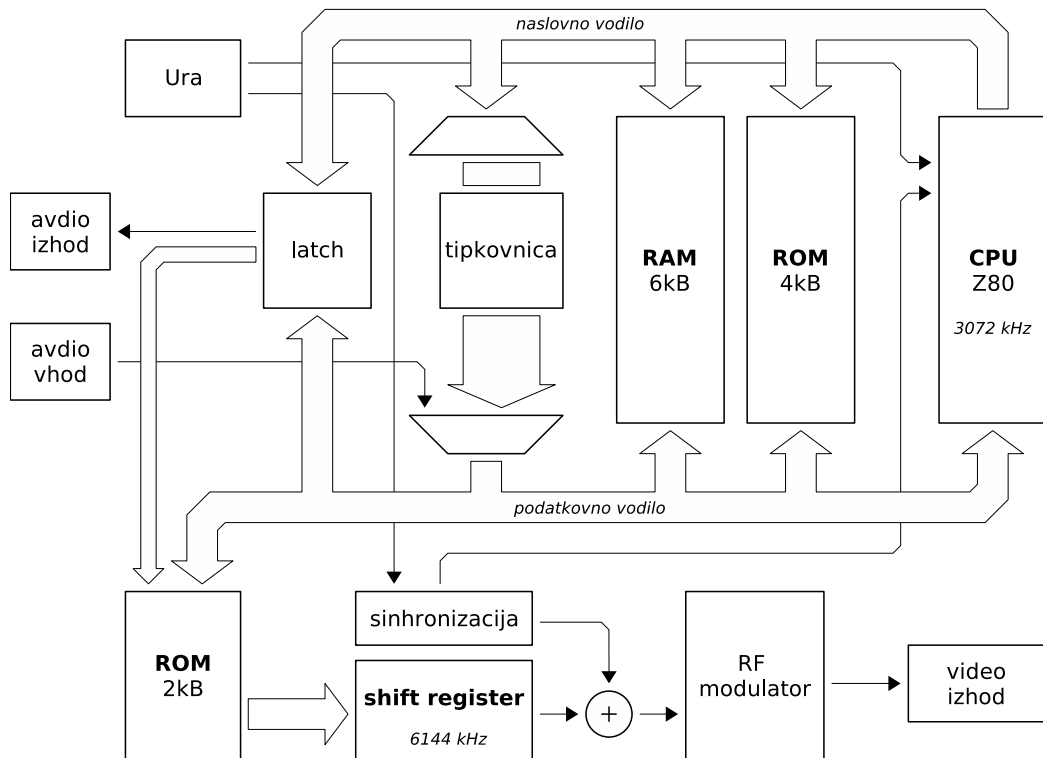
Galaksijo po drugi strani sestavlja razmeroma preprosto elektronsko vezje, sestavljeno izključno iz integriranih vezij z nizko ali srednjo stopnjo integracije (izjema je mikroprocesor Z80, za katerega pa je na voljo vsa dokumentacija, od nivoja logičnih vrat dalje). Tudi hitrost delovanja je dovolj nizka, da so za analizo delovanja dovolj najcenejši osciloskopi na tržišču. Hkrati pa Galaksija predstavlja celoten delujoč mikroprocesorski sistem, ki vsebuje enake glavne sestavine (mikroprocesor, pomnilnik, vodilo, periferne enote, itd.) kot sodobni sistemi.

1.3 Cilji

Cilji dela so:

- Opisati delovanje strojne opreme (ter delov programske opreme, katerih delovanje je tesno povezano s strojno opremo) mikroračunalnika Galaksija tako podrobno, da ta opis omogoča izdelavo računalnika, ki je v največji možni meri združljiv z izvirno programsko opremo.
- Po zgoraj omenjenem opisu izdelati računalnik, združljiv z izvirno Galaksijo, iz elektronskih komponent, ki so danes na voljo na trgu. Pri tem naj bo tudi elektronsko vezje podobno izvirnemu, kjer je to praktično.
- Dokumentirati posebne načrtovalske prijeme, ki so bili uporabljeni v izvirnem računalniku, in tako olajšati podobne projekte obnavljanja strojne opreme iz 80-ih let 20. stoletja.

2 STROJNA ARHITEKTURA



Slika 5: Blok shema mikroračunalnika Galaksija

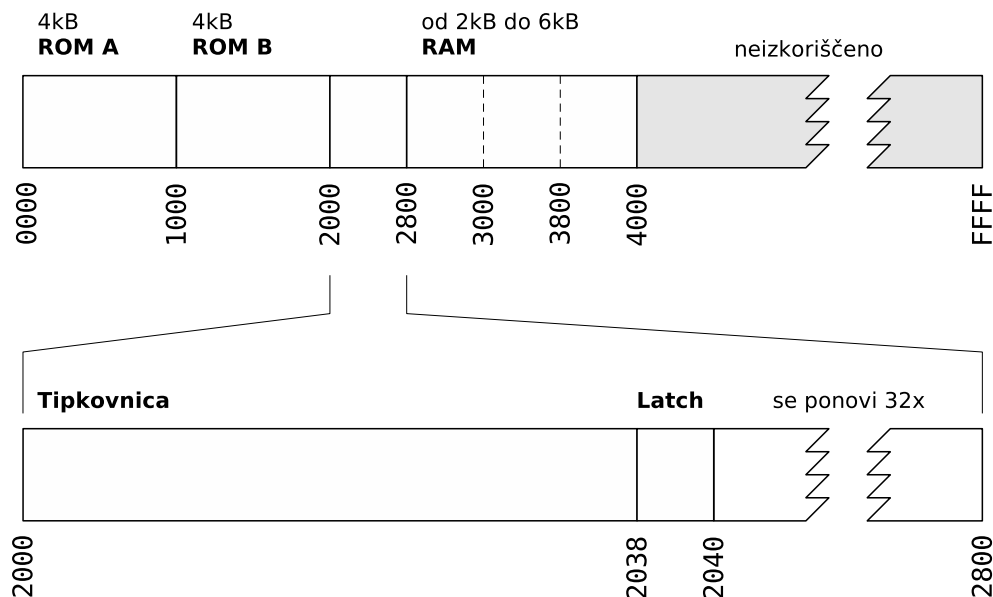
2 Strojna arhitektura

Blok shema vezja matične plošče je prikazana na sliki 5. Faksimile originalnega vezalnega načrta [2] se nahaja v prilogi.

2.1 Uporaba naslovnega prostora

Mikroprocesor Z80 [6] ima dva ločena naslovna prostora: 16-bitnega za pomnilnik in 8-bitnega za vhodno/izhodne operacije (IO). Galaksija na matični plošči uporablja samo pomnilniški naslovni prostor, tako za dostop do pomnilnika kot tudi za komunikacijo s periferijo (t.i. memory mapped IO).

Na sliki 6 ja prikazana razporeditev pomnilnika in strojne opreme po pomnilniškem naslovnem prostoru (angl. memory map). Prikazana razporeditev je določena z vezjem naslovnega dekoderja in je ni mogoče programsko spreminjati (z izjemo nadzora A7 naslovne linije RAM pomnilnika,



Slika 6: Uporaba naslovnega prostora mikroprocesorja Z80

slika 9). Neizkoriščeni del naslovnega prostora ni uporabljen in je rezerviran za razširitve. Vrednosti, ki jih dobimo z branjem iz teh pomnilniških naslovov, so nedefinirane.

Bralni pomnilnik velikosti 4 kB s sistemsko programsko opremo (ROM A) se nahaja na naslovu 0x0000 in zaseda vse prekinitvene vektorje mikroprocesorja. Na naslovu 0x1000 je predviden prostor za še en bralni pomnilnik velikosti 4 kB, namenjen razširitvam (ROM B).

Pisanje na naslove, ki jih uporablja ROM, ni dovoljeno. Na računalnikih brez ROM-a B je rezultat branja iz naslovov, uporabljenih zanj, nedefiniran.

Na naslovu 0x2800 se nahaja od 2 kB do 6 kB statičnega RAM pomnilnika, ki ne potrebuje osveževanja s števcem za osveževanje pomnilnika (memory refresh counter) mikroprocesorja Z80.

Celoten vhodno/izhodni naslovni prostor je rezerviran za razširitve. Vrednosti, ki jih dobimo z vhodnimi operacijami (na primer `in`), so nedefinirane.

2.2 Periferija

Procesor dostopa do perifernih naprav preko 2 kB velikega dela pomnilniškega naslovnega prostora na naslovu 0x2000.

2 STROJNA ARHITEKTURA

	0	1	2	3	4	5	6	7
0000		A	B	C	D	E	F	G
0008	H	I	J	K	L	M	N	O
0010	P	Q	R	S	T	U	V	W
0018	X	Y	Z	↑	↓	←	→	space
0020	0	1	2	3	4	5	6	7
0028	8	9	;	:	,	=	.	/
0030	return	break	repeat	delete	list	shift		

Slika 7: Naslovi posameznih tipk in izhoda iz komparatorja

2.2.1 Tipkovnica

Tipkovnica ima 54 tipk. Vsaka posamezna tipka ima dodeljen svoj naslov v pomnilniku, kot je prikazano na sliki 7¹. Prikazan je odmik od osnovnega naslova. Zaradi nepopolnega dekodiranja se namreč lahko uporablja katerikoli od 32 osnovnih naslovov (0x2000, 0x2040 ... 0x27C0). Potemnjena polja so neizkoriščena in se vedno obnašajo kot nepritisnjene tipke.

Če je najnižji bit (LSB) na določenem naslovu enak 0, potem je tipka pritisnjena. Nasprotno, če je bit enak 1, tipka ni pritisnjena. Ostali biti imajo nedefinirane vrednosti.

Pisanje na naslove, ki jih uporablja tipkovnica, ni dovoljeno.

2.2.2 Vmesnik za magnetofon

Vmesnik za magnetofon je sestavljen iz analognega vhoda in izhoda.

Analogni vhod je realiziran kot preprost komparator s high-pass filtrom na vhodu. Če se na vhodu pojavi impulz amplitude več kot približno 700 mV (V_{be} tranzistorja T2), se na naslovih komparatorja, označenih na sliki 7, najnižji bit (LSB) postavi na 0 za čas trajanja impulza. Največja dolžina impulza je določena s filtrom, ki ga tvorita C2 in R14².

¹Izjema sta obe tipki *SHIFT*, ki se iz vidika programske opreme obnašata kot ena sama tipka, ki je pritisnjena, če je pritisnjena katerakoli od fizičnih tipk *SHIFT*.

²Mejna frekvenca filtra glede na objavljene podatke [2] je približno 16 kHz, kar je glede na frekvenčni odziv povprečnega avdio magnetofona previsoka vrednost. Sklepam, da gre za napako v tisku.

2.3 Nadzor linije A7 za RAM

AOUT0	AOUT1	U_a [V]
0	0	0.0
1	0	0.5
0	1	0.5
1	1	1.0

Tabela 3: Izhodna napetost analognega izhoda U_a v odvisnosti od bitov AOUT0 in AOUT1

- A7CLMP Če je ta bit postavljen na 0, se naslovna linija A7 za RAM pomnilnik postavi na 1, ne glede na vrednost na naslovnem vodilu procesorja. Slika 9.
- AOUT0-1 Nadzor digitalno-analognega pretvornika za magnetofonski vmesnik
- CHR0-3 Nadzor trenutne vrstice, ki jo izpisuje generator znakov (character generator).

Tabela 4: Pomeni posameznih bitov v registru latch

Analogni izhod se krmili preko dveh bitov v registru latch (slika 8, AOUT0, AOUT1), ki tvorita enostaven digitalno-analogni pretvornik. Izhodna napetost v odvisnosti od stanja teh dveh bitov je prikazana v tabeli 3.

2.2.3 Latch

Mikroprocesor lahko na naslovih oblike 0x0010 0xxx xx11 1xxx (najnižji tak naslov je 0x2038, najvišji pa 0x27FF) dostopa do 6-bitnega registra (v izvirni dokumentaciji imenovanega „latch“), ki se uporablja za krmiljenje generatorja znakov, analognega izhoda in A7 naslovne linije RAM pomnilnika.

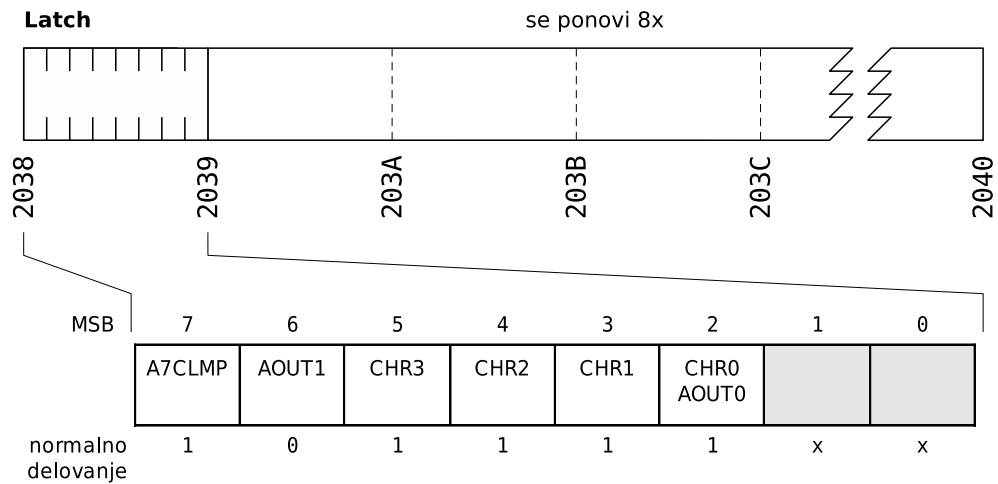
V zgornje naslove je mogoče le zapisovati 8-bitno vrednost. Branje vrne nedefinirano vrednost.

Pomeni posameznih bitov (po oznakah s slike 8) so podani v tabeli 4.

2.3 Nadzor linije A7 za RAM

Z nastavljanjem bita A7CLMP je mogoče naslovno linijo A7 RAM pomnilnika postaviti na 1, ne glede na vrednost na naslovnem vodilu procesorja. Če RAM

2 STROJNA ARHITEKTURA



Slika 8: Oznake posameznih bitov v registru latch

razdelimo na bloke, velike 128 bytov, lahko z uporabo tega bita dosežemo, da mikroprocesor bere vrednosti, zapisane v lihih blokih, tudi na naslovih, ki jih sicer zasedajo sodi bloki.

Ta funkcija se uporablja pri generiranju video signala zaradi posebnosti mikroprocesorja Z80, ki pri osveževanju pomnilnika samodejno ne povečuje naslovne linije A7 [6].

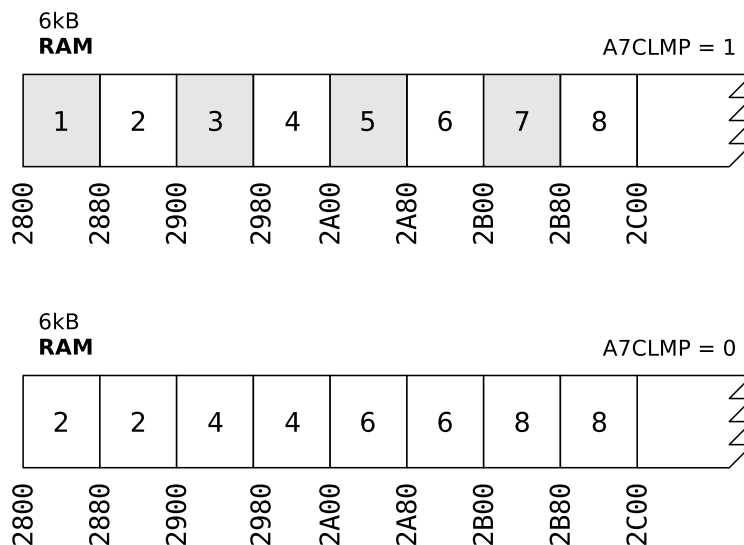
2.4 Generiranje video signala

Galaksija je namenjena priklopu na računalniški monitor s kompozitnim video vhodom ali televizijski sprejemnik (preko UHF modulatorja). V obeh primerih gre za prikazovalnik s katodno cevjo, zato se v nadaljevanju opis delovanja opira na pot elektronskega žarka preko zaslona Braunove elektronke.

Zaradi preproste strojne opreme je za generiranje video signala potrebno sodelovanje mikroprocesorja. Pri tem je bistvenega pomena natančna sinhronizacija delovanja programske opreme z odklonom elektronskega žarka.

Video gonilnik, ki je odgovoren za generacijo video signala, se nahaja v ROM-u A na naslovu 0x0038, to je prekinitvenem vektorju za prekinitev INT (software maskable interrupt) v prekinitvenem načinu IM 1.

2.4 Generiranje video signala



Slika 9: Vpliv bita A7CLMP na razporeditev RAM pomnilnika v naslovnem prostoru mikroprocesorja

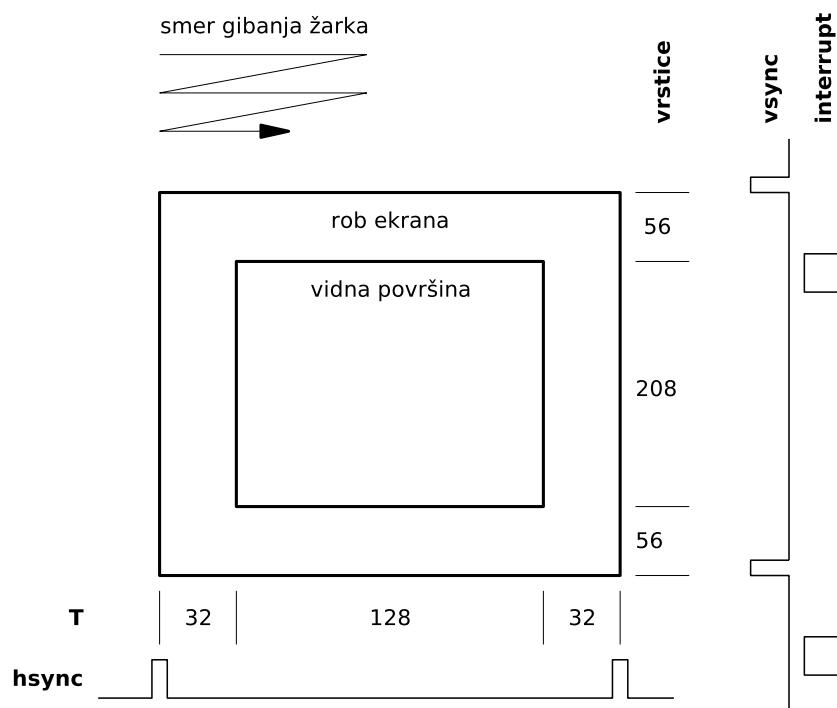
2.4.1 Sinhronizacija

Osnovni časovni intervali, pomembni za generiranje video signala, so prikazani na sliki 10. Urina takta mikroprocesorja (3072 kHz) in vezja za video signal (6144 kHz) sta sinhronizirana. Površina zaslona je razdeljena na črn rob, ki ni uporaben za prikaz slike, in uporabno površino v sredini. Izrisovanje ene vrstice na zaslon vedno traja 128 urinih ciklov mikroprocesorja, celotna uporabna površina pa je sestavljena iz 208 vrstic. Horizontalni in vertikalni sinhronizacijski impulzi se generirajo strojno.

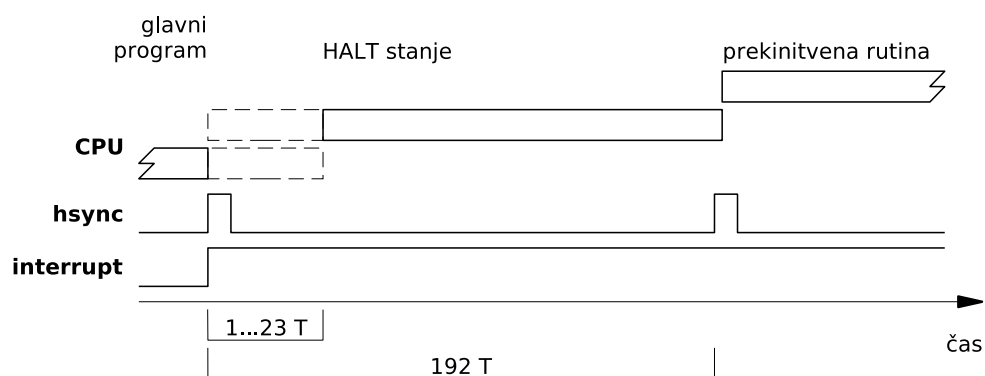
Ko se elektronski žarek ob 56. horizontalnem sinhronizacijskem impulzu začne premikati na začetek prve vrstice vidne površine ekrana, video vezje prekine izvajanje uporabnikovega programa preko INT prekinitve.

Zakasnitev pri odzivu mikroprocesorja na prekinitev ni deterministična. Odvisna je namreč od časa, ki ga procesor potrebuje, da konča izvajanje trenutnega ukaza, kar pomeni od 1 do 23 urinih taktov. Ker pa se mora koda video gonilnika izvajati popolnoma sinhrono s prehodom žarka čez površino zaslona, se izvajanje mikroprocesorja po njegovem odzivu na prekinitev zaustavi za čas ene zaslonske vrstice (192 urinih ciklov). To je izvedeno z vnašanjem streznega števila HALT stanj v delovanje procesorja, kar je izvedeno z vezjem, priključenim na WAIT signal procesorja. Slika 11.

2 STROJNA ARHITEKTURA



Slika 10: Časovni okvirji, pomembni za pravilno delovanje programske opreme za generiranje video signala



Slika 11: Časovni diagram delovanja mikroprocesorja takoj po prekinitvi

2.4.2 Generator znakov

Informacija o zaslonski sliki je shranjena v 512 bytov dolgem delu RAM pomnilnika na naslovu 0x2800. V tem delu so shranjene ASCII kode znakov (16 vrstic po 32 znakov), ki se ob vsaki osvežitvi slike izrišejo na zaslon.

Naloga video gonilnika, ki se med osveževanjem izvaja na mikroprocesorju, je zaporedno branje 8-bitnih ASCII kod iz tega dela pomnilnika in prenos le-teh do generatorja znakov.

Generator znakov predstavlja 2 kB ROM pomnilnik (angl. character generator ROM), v katerem je shranjena preslikava iz ASCII kode³ znaka v bitno sliko (angl. bitmap) točk, ki ustrezajo prikazu tega znaka na zaslonu. Način zapisa znaka je prikazan na sliki 12. Bit 0 v bitni sliki pomeni svetlo točko na zaslonu, bit 1 pa temno točko.

Spodnjih 7 naslovnih linij ROM-a je povezanih na podatkovno vodilo mikroprocesorja, zgornje 4 linije pa na bite od CHR0 do CHR3 v latchu. Tako lahko procesor z nastavljanjem latcha in postavljanjem vrednosti na podatkovno vodilo (na primer z branjem vrednosti iz RAM pomnilnika) iz generatorja znakov priključijo poljubno slikovno vrstico (angl. scanline) poljubnega znaka.

Na sliki 13 je prikazanih vseh 128 znakov, ki jih lahko na zaslonu prikaže generator znakov. Ker linija D6 podatkovnega vodila ni povezana, vsakemu shranjenemu znaku ustrezata dve 8-bitni ASCII kodi. Razporeditev znakov je izbrana tako, da se kode prvih 64 znakov pokrivajo z osrednjo polovico osnovnega ASCII nabora znakov (ASCII kode od 20 do 5F). Ostalih 64 znakov pa sestavlja nabor psevdografičnih znakov, ki omogočajo grafični prikaz ločljivosti 64x48 pik.

Na zaslonu je vsak znak sestavljen iz 13 vrstic ($13 \cdot 16 = 208$). Video gonilnik v ROM-u uporablja le prvih 12 slikovnih vrstic od 16, ki so sicer shranjene v ROM-u. 13. slikovna vrstica vsakega znaka je na zaslonu vedno prazna (črna), ker se v času, ko jo preletava elektronski žarek, video gonilnik pripravlja na risanje naslednje vrstice znakov, generator znakov pa je izključen.

Pomembno je tudi dejstvo, da je zadnja slikovna vrstica (naslov 0x0F) vseh zapisanih znakov prazna (vsebuje same enice). Ta vrstica je v latchu

³Izraz ASCII koda se tukaj uporablja z namenom razlikovanja kode znaka, zapisanega v pomnilniku n kode, pod katero je znak shranjen v ROM-u generatorja znakov. Standardu American Standard Code for Information Interchange sicer ustreza le del Galaksijinega nabora znakov.

2 STROJNA ARHITEKTURA

Naslov												Vsebina								
Vrstica				Znak																
CPU					D7	D6	D5	D4	D3	D2	D1	D0								
					0	1	0	0	0	0	0	1	= hex 41 (ASCII Majuscule A)							
ROM	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	
	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	
	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	
	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	
	0	0	1	1	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	
	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	0	1	
	0	1	0	1	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	
	0	1	1	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	
	0	1	1	1	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	
	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	
	1	0	0	1	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	
	1	0	1	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	
	1	0	1	1	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	
	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	
	1	1	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	
	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	
	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	

Slika 12: Zapis ASCII znaka A v ROM-u generatorja znakov

2.4 Generiranje video signala

ASCII naslov	ASCII naslov	ASCII naslov	ASCII naslov
20 20	30 30 0	40 00 €	50 10 P
21 21 !	31 31 1	41 01 A	51 11 Q
22 22 "	32 32 2	42 02 B	52 12 R
23 23 #	33 33 3	43 03 C	53 13 S
24 24 \$	34 34 4	44 04 D	54 14 T
25 25 %	35 35 5	45 05 E	55 15 U
26 26 &	36 36 6	46 06 F	56 16 V
27 27 >	37 37 7	47 07 G	57 17 W
28 28 (38 38 8	48 08 H	58 18 X
29 29)	39 39 9	49 09 I	59 19 Y
2A 2A *	3A 3A :	4A 0A J	5A 1A Z
2B 2B +	3B 3B ;	4B 0B K	5B 1B Ć
2C 2C ,	3C 3C <	4C 0C L	5C 1C Č
2D 2D -	3D 3D =	4D 0D M	5D 1D Ž
2E 2E .	3E 3E >	4E 0E N	5E 1E Š
2F 2F /	3F 3F ?	4F 0F O	5F 1F _

(se ponovi pri ASCII 60-7F) (se ponovi pri ASCII 00-1F)

ASCII naslov	ASCII naslov	ASCII naslov	ASCII naslov
80 40	90 50	A0 60	B0 70
81 41	91 51	A1 61	B1 71
82 42	92 52	A2 62	B2 72
83 43	93 53	A3 63	B3 73
84 44	94 54	A4 64	B4 74
85 45	95 55	A5 65	BB 75
86 46	96 56	A6 66	B6 76
87 47	97 57	A7 67	B7 77
88 48	98 58	A8 68	B8 78
89 49	99 59	A9 69	B9 79
8A 4A	9A 5A	AA 6A	BA 7A
8B 4B	9B 5B	AB 6B	BB 7B
8C 4C	9C 5C	AC 6C	BC 7C
8D 4D	9D 5D	AD 6D	BD 7D
8E 4E	9E 5E	AE 6E	BE 7E
8F 4F	9F 5F	AF 6F	BF 7F

(se ponovi pri ASCII C0-DF) (se ponovi pri ASCII E0-FF)

Slika 13: Tabela znakov, shranjenih v ROM-u generatorja znakov

2 STROJNA ARHITEKTURA

nastavljena med tem, ko se izvaja uporabnikov program. Na ta način je generator znakov izključen v času, ko vsebina na podatkovnem vodilu mikroprocesorja ni pod neposrednim nadzorom video gonilnika. S tem je poskrbljeno tudi, da je intenziteta žarka na minimumu v času, ko ta preletava črni rob zaslona in med vračanjem žarka po diagonali na izhodišče.

2.4.3 Video gonilnik

Video gonilnik mora med prehodom elektronskega žarka preko uporabne površine zaslona poskrbeti, da se vsake 4 takte ure procesorja (3072 kHz) na podatkovnem vodilu pojavi byte, ki ustreza ASCII kodi znaka, kateremu pripada naslednjih 8 pik na zaslonu.

Zaporedno branje video dela pomnilnika je izvedeno s funkcijo Z80 mikroprocesorja [6], ki je sicer namenjena osveževanju dinamičnih pomnilnikov. Gonilnik neposredno pred začetkom risanja zaslonske slikovne vrstice nastavi registra I in R mikroprocesorja tako, da sestavljata kazalec, ki kaže na pomnilniško lokacijo, kjer se nahaja začetek vrstice 32 znakov. Interna logika mikroprocesorja nato sama poskrbi, da se v nadaljevanju izvajanja programa ob vsakem M1 ciklu procesorja prebere 1 byte iz pomnilnika in inkrementira register R.

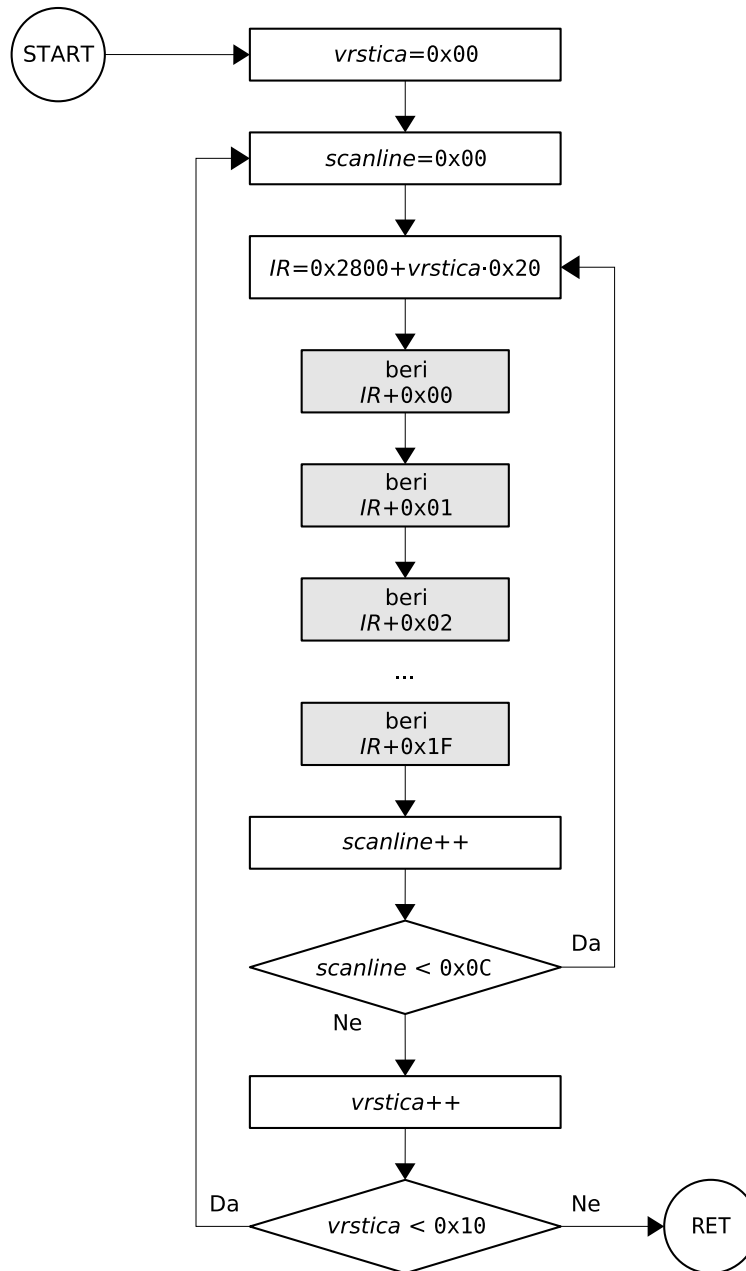
Da se zadosti pogoju prenosa enega byta na 4 takte ure, mora video gonilnik po nastavitvi I in R registrov izvajati (poljubnih) 32 procesorskih ukazov, ki imajo le en M cikel (torej so sestavljeni le iz M1 cikla in trajajo 4 urine takte). Po preteku 32 M ciklov je trenutna vrstica izrisana in video gonilnik se v času, ko se žarek pomika čez rob zaslona, lahko pripravi na risanje nove vrstice.

Pred risanjem vsake slikovne vrstice je potrebno tudi pravilno nastaviti CHRx bite v latchu in s tem v generatorju znakov nastaviti številko trenutne slikovne vrstice, po risanju pa se generator znakov izključi z izbiro vrstice 0x0F.

Mikroprocesor sam ne inkrementira registra I in najvišjega bita registra R. Register I gonilnik ustrezno nastavi programsko v času, ko se pripravlja na risanje nove vrstice, najvišji bit registra R pa se nastavlja skupaj z izbiro slikovne vrstice preko bita A7CLMP v latchu.

Na sliki 14 je prikazan poenostavljen diagram izvajanja video gonilnika. Diagram ne prikazuje kode gonilnika, potrebne za sinhronizacijo branja pomnilnika z elektronskim žarkom, nastavljanje linije A7 in zapisovanje številke trenutne slikovne vrstice v latch.

2.4 Generiranje video signala



Slika 14: Poenostavljen diagram izvajanja video gonilnika

2 STROJNA ARHITEKTURA

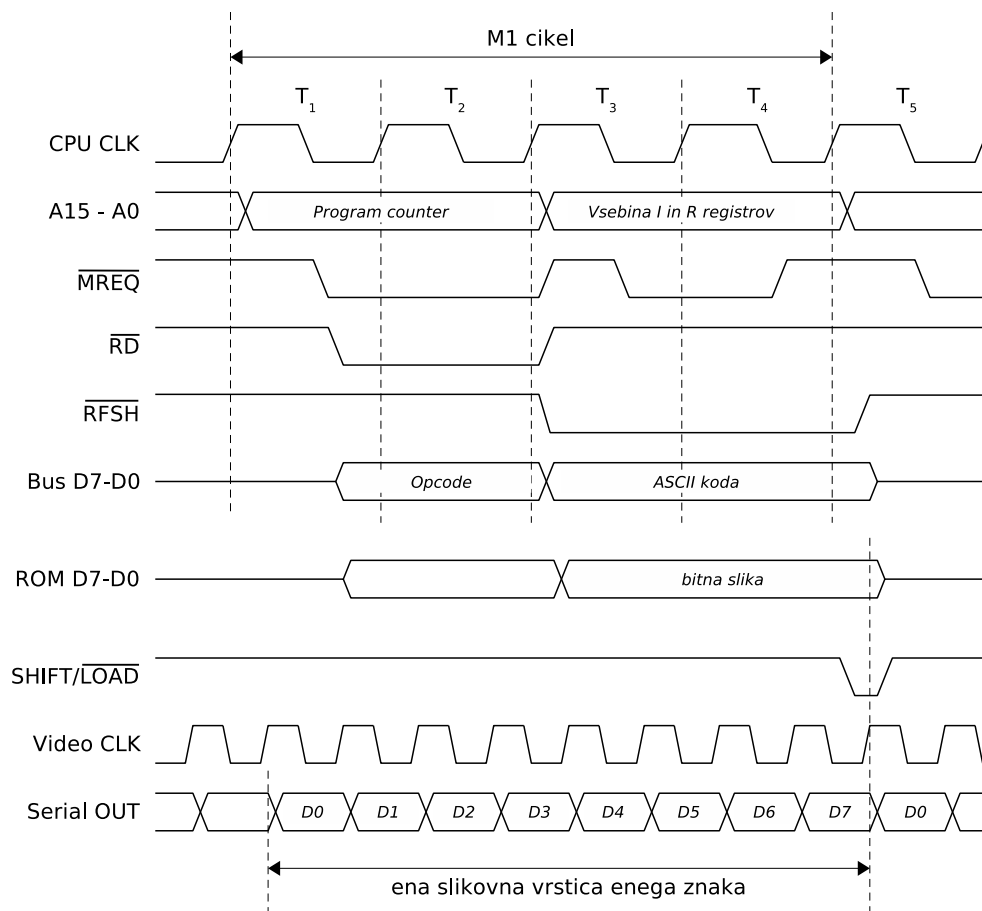
2.4.4 Shift register

Izhod generatorja znakov je priključen na paralelno/serijski pomikalni register. Ta v pravem trenutku vzporedno naloži 8 bitov iz generatorja znakov in jih tekom 8 ciklov video ure (6144 kHz) zaporedno pošilja na video izhod.

Zaporedno nalaganje se izvrši po koncu vsakega M1 cikla mikroprocesorja (slika 15). V času, ko deluje video gonilnik in ko žarek preletava uporabno površino zaslona, se ob M1 ciklih v shift register po vrsti nalagajo bitne slike znakov. Ko pa se izvaja uporabniški program, se v register nalagajo prazne (temne) vrstice 0x0F. S tem je doseženo, da se v času sinhronizacijskih impulzov in prehoda žarka čez črni rob zaslona v video signalu ne pojavljajo naključne vrednosti iz podatkovnega vodila.

Ko se izvajajo ukazi, ki imajo več M ciklov (in torej trajajo več kot 4 takte procesorske ure), se shift register popolnoma izprazni (saj med dvema vzporednima nalaganjima mine več kot 8 ciklov video ure). Serijski vhod registra je zaradi tega povezan na logično 1. Tako se tudi v primeru izvajanja daljših procesorskih ukazov v uporabnikovem programu na zaslonu ne izrisuje nič.

2.4 Generiranje video signala



Slika 15: Delovanje video shift registra

3 Nova Galaksija

Elektronsko vezje nove Galaksije je v glavnih značilnostih podobno originalu. Spremenjeni so bili tisti deli, ki jih zaradi nedosegljivosti ekvivalentnih komponent ni bilo mogoče implementirati po originalnem načrtu. Pri načrtovanju replike je imela zanesljivost delovanja prednost pred ceno izdelave, zaradi česar so bile na račun kompleksnejšega vezja odpravljene tudi vse znane pomanjkljivosti izvirnega vezja, pod pogojem da takega popravka ne bi bilo mogoče zaznati s programskega vidika. Novo vezje je tudi prilagojeno za lažji priklop na sodobne televizijske sprejemnike oziroma monitorje.

Vezalni načrt vezja in shema tiskanine se nahajata v prilogi.

3.1 Digitalni del

Digitalni del vezja je implementiran z integriranimi vezji v high-speed CMOS tehnologiji (družina logičnih vezij 74HC).

3.1.1 Mikroprocesor in pomnilnik

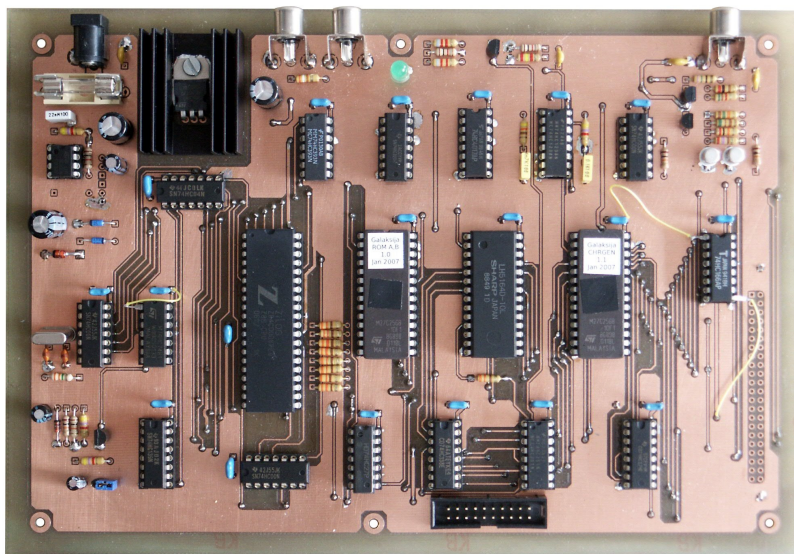
Vlogo centralne procesne enote (U1) ima v novem vezju integrirano vezje Z84C0008 podjetja Zilog. Gre za mikroprocesor, izdelan v CMOS tehnologiji v DIP40 ohišju, ki je strojno in programsko popolnoma kompatibilen z mikroprocesorjem Z80, ki se je uporabljal v originalni Galaksiji.

Bralni pomnilnik (U2) je implementiran z enim samim 32 kB EPROM vezjem 27C256 v CMOS tehnologiji. Uporabljeno sodobno vezje ima dovolj veliko kapaciteto, da je bilo vanj mogoče shraniti tako osnovni operacijski sistem, kot tudi prvo razširitev zanj (tako to integrirano vezje v naslovnem prostoru mikroprocesorja zavzema ROM A in ROM B poziciji na sliki 6). Zgornjih 24 kB pomnilnika je neizkoriščenih.

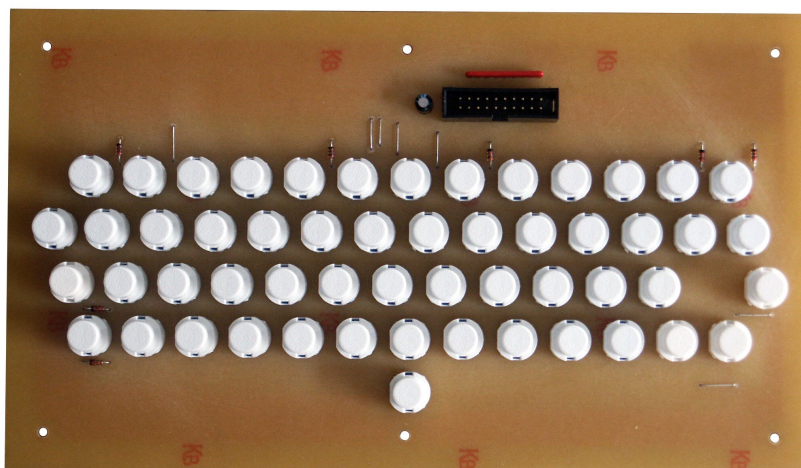
Zaradi izvedbe naslovnega dekodeja je EPROM vezje vključeno (izhodi v stanju nizke impedance) tudi, ko mikroprocesor poizkusi pisati v del naslovnega prostora, ki ga ta zavzema. Ker sta v tem primeru na podatkovnem vodilu dve napravi z izhodi v stanju nizke impedance, bi po vodilu stekel kratkostični tok, ki bi lahko poškodoval mikroprocesor ali EPROM. Za omejevanje toka v tem primeru skrbijo upori R10-R17 med podatkovnimi linijami EPROM-a in podatkovnim vodilom.

Delovni pomnilnik (U3) sestavlja 8 kB SRAM vezje LH5164D. Spodnjih 2 kB pomnilnika je neizkoriščenih (ta del naslovnega prostora zaseda periferija).

3.1 Digitalni del



Slika 16: Matična plošča nove Galaksije



Slika 17: Tipkovnica nove Galaksije

3 NOVA GALAKSIJA



Slika 18: Nova Galaksija v ohišju

3.1.2 Naslovni dekodler

Naslovni dekodler sestavljata dvojni 2/4 demultiplekser U10 in logična vrata U18 in implementira naslavljanje EPROM in SRAM vezij ter periferije v skladu s sliko 6.

Naslovi tipk, komparatorja in latcha v naslovnem prostoru so posledica vezave matrike tipkovnice, demultipleksorja U7 in multipleksorja U6.

3.1.3 Tipkovnica

Nova Galaksija ima tipkovnico na ločeni tiskanini, ki je priključena na matično ploščo preko 20-polnega konektorja.

Tipkovnica originalne Galaksije je za svoje delovanje potrebovala izhode demultipleksorja v izvedbi z odprtim kolektorjem. Ker uporabljeni demultipleksor U7 nima takšnih izhodov, so le-ti simulirani s priključitvijo diod D1 do D7.

Izhod multipleksorja je pred kratkim stikom zaščitem z uporom R30 iz podobnega razloga kot izhodi EPROM-a.

3.1.4 Delilnik ure

Delilnik ure sestavljajo 4 4-bitni števeci (ripple counter) v vezjih U12 in U13. Ti iz signala oscilatorja s frekvenco 6144 kHz po naslednjih enačbah pridobijo signale, potrebne za generiranje kompozitnega videa.

$$\begin{aligned}f_{video} &= f_{osc} = 6144\text{kHz} \\f_{cpu} &= \frac{f_{osc}}{2} = 3072\text{kHz} \\f_{hsync} &= \frac{f_{osc}}{12 \cdot 16 \cdot 2} = 16\text{kHz} \\10 \cdot f_o &= \frac{f_{osc}}{12 \cdot 16 \cdot 16 \cdot 4} = 500\text{Hz}\end{aligned}$$

S pomočjo signala $10 \cdot f_o$ Johnsonov števec U14 generira dva fazno zamaknjena signala frekvence $f_o = 50\text{Hz}$ (signal za prekinitev procesorja in proženje vertikalne sinhronizacije) v skladu s potrebami generacije kompozitnega video signala (slika 11).

3 NOVA GALAKSIJA

Karakteristika	PAL B,G [7]	Galaksija
Št. vrstic na okvir	625	320
Št. polj na sekundo	50	50
Horizontalna frekvenca	15,625 \pm 0.02% Hz	16,000 Hz
Prepletenost	2/1	1/1
Črni nivo	0 IRE	0 IRE
Beli nivo	100 IRE	100 IRE
Sinhronizacijski nivo	-43 IRE	-43 IRE
Nazivna pasovna širina	5 MHz	3 MHz

Tabela 5: Primerjava karakteristik video signala Galaksije s PAL B,G standardom

Primerjava karakteristik tako pridobljenega video signala s televizijskim standardom, ki se uporablja v Sloveniji, je prikazana v tabeli 5. Kljub odstopanjem v horizontalni frekvenci, številu vrstic in neprepletenosti slike večina televizijskih sprejemnikov signal pravilno prikaže na zaslonu.

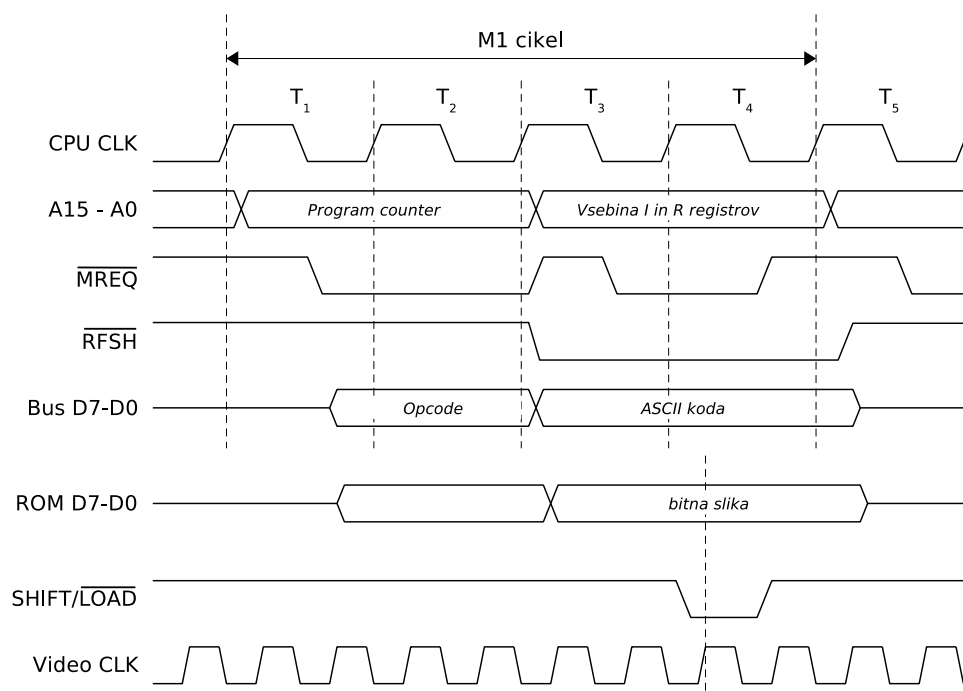
3.1.5 Shift register

Logično vezje, ki ga sestavljajo vrata U17, detektira četrto T stanje cikla M1 in proži vzporedno nalaganje podatkov v shift register U6 (slika 19).

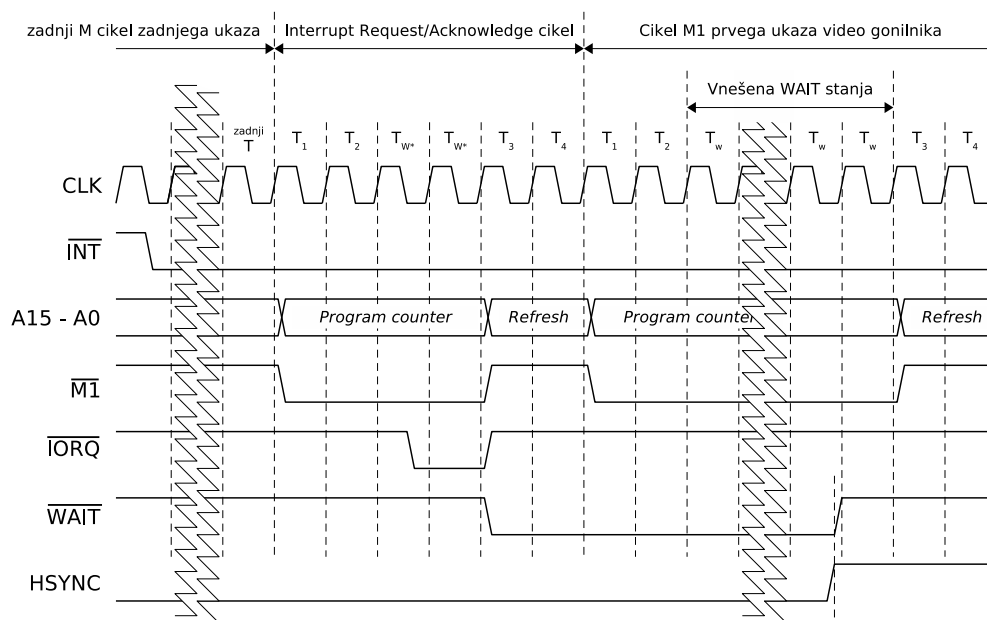
Nalaganje podatkov v shift register se v novem vezju izvrši dva cikla video ure prej kot pri originalnem vezju (primerjaj s sliko 15). Razlog za to spremembo je večja zanesljivost delovanja (stran 43), posledica pa, da je slika v primerjavi z originalno Galaksijo premaknjena dve piki v levo.

3.1.6 Sinhronizacija prekinitov

Sekvenčno logično vezje, sestavljeno iz spominske celice U21 in vrat U19, detektira odziv mikroprocesorja na prekinitov in zaustavi izvajanje prvega ukaza video gonilnika do naslednjega horizontalnega sinhronizacijskega impulza (slika 20).



Slika 19: Časovni diagram delovanja vezja za nadzor shift registra



Slika 20: Časovni diagram delovanja vezja za sinhronizacijo prekinitev

3.2 Analogni del

3.2.1 Napajalnik

Za napajanje računalnika iz omrežne izmenične napetosti 240V je uporabljen zunanji 12V usmernik moči 5W. Na matični plošči se 12V enosmerna napetost dalje pretvarja v +5V in -5V napajalni napetosti.

Napetost +5V se uporablja za napajanje vseh digitalnih delov in analognih delov in se pridobiva s stabilizatorjem napetosti U9.

Napetost -5V se uporablja za napajanje video ojačevalnika in se pridobiva s preklopnim napetostnim inverterjem in stabilizatorjem napetosti U22 (dodatek, stran 63).

3.2.2 Oscilator

Kristalni oscilator generira osnovni urin signal frekvence 6144 kHz. Kot aktivni del so v oscilatorju uporabljena logična vrata U19.

3.2.3 Reset vezje

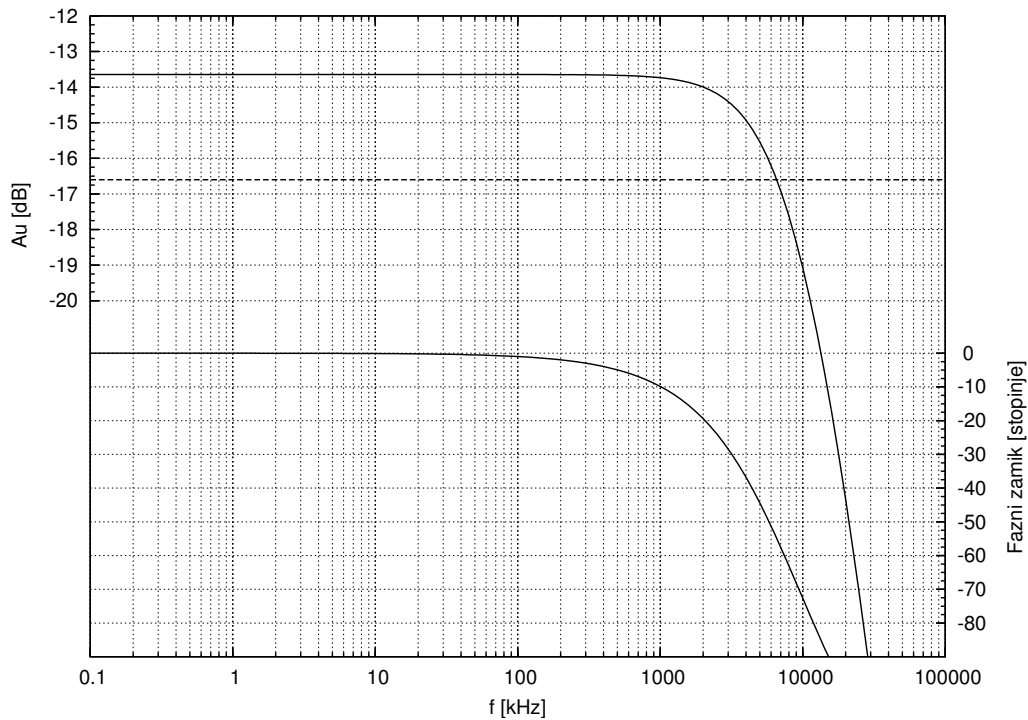
Reset vezje še kratek čas po vklopu napajanja drži procesor v reset stanju. Na ta način se zagotovi, da so ob zagonu operacijskega sistema minili vsi morebitni prehodni pojavi v napajalnih napetostih, ki bi sicer lahko povzročili napake v delovanju.

3.2.4 Kompozitni video

Iz urinih signalov, ki jih generira delilnik ure, se z dvema monostabilnima multivibratorjema (vezje U15) generirajo horizontalni in vertikalni sinhronizacijski impulzi. Ti impulzi se nato združijo v kompozitni sinhronizacijski signal z XOR logično funkcijo, sestavljeno iz logičnih vrat U16.

Sinhronizacijski signal in video signal iz shift registra se nato vodita v mešalnik (Q4, Q5). Z uporoma R21 in R22 je določeno razmerje med sinhronizacijskim in črnim nivojem.

Tako ustvarjeni kompozitni video signal nato vodimo najprej na uporovni delilnik (R24, R27), ki prilagodi amplitudo signala, nato pa v dvostopenjski ojačevalnik v konfiguraciji skupni kolektor (Q6, Q8). Generirani signal ima tako $V_{pp} = 2V$ pri izhodni upornosti 75Ω .



Slika 21: Fazni in amplitudni Bodejev diagram video ojačevalnika (SPICE simulacija)

Izračun pasovne širine video signala [8][9]:

$$\begin{aligned}
 BW &= \frac{1}{2} \cdot K \cdot N_{ht} \cdot N_{vt} \cdot f_o \cdot K_h \dot{V}_h = \\
 &= \frac{1}{2} \cdot 0.7 \cdot 256 \cdot 208 \cdot 50\text{Hz} \cdot \frac{64 + 256 + 64}{256} \cdot \frac{56 + 208 + 56}{208} = \\
 &= 2.2\text{MHz}
 \end{aligned}$$

kjer je K Kell faktor, N_{ht} in N_{vt} horizontalna in vertikalna ločljivost zaslona ter K_h in K_v delež časa, ki ga zasedata horizontalna in vertikalna sinhronizacija.

Iz grafa na sliki 21 vidimo, da ima ojačevalnik zadostno pasovno širino in da ima v frekvenčnem pasu, ki ga obsega video signal, okoli 0.5 dB linearnega popačenja.

3 NOVA GALAKSIJA

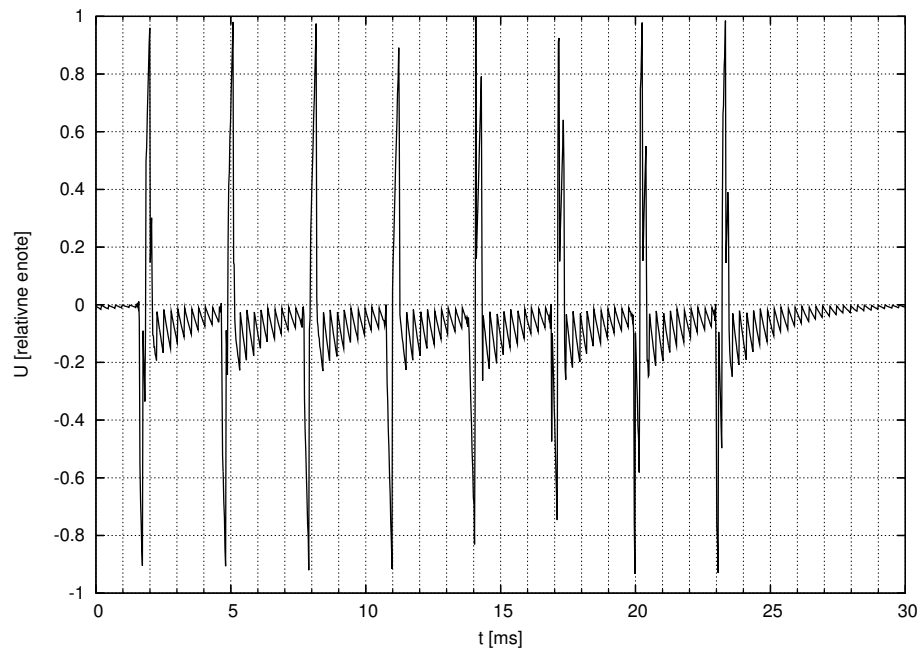
3.2.5 Vmesnik za magnetofon

Kot analogni izhod vmesnika za magnetofon je podobno kot pri originalni Galaksiji uporabljen preprost digitalno-analogni pretvornik. Izhodne napetosti so odvisne od uporov R25, R26 in R28.

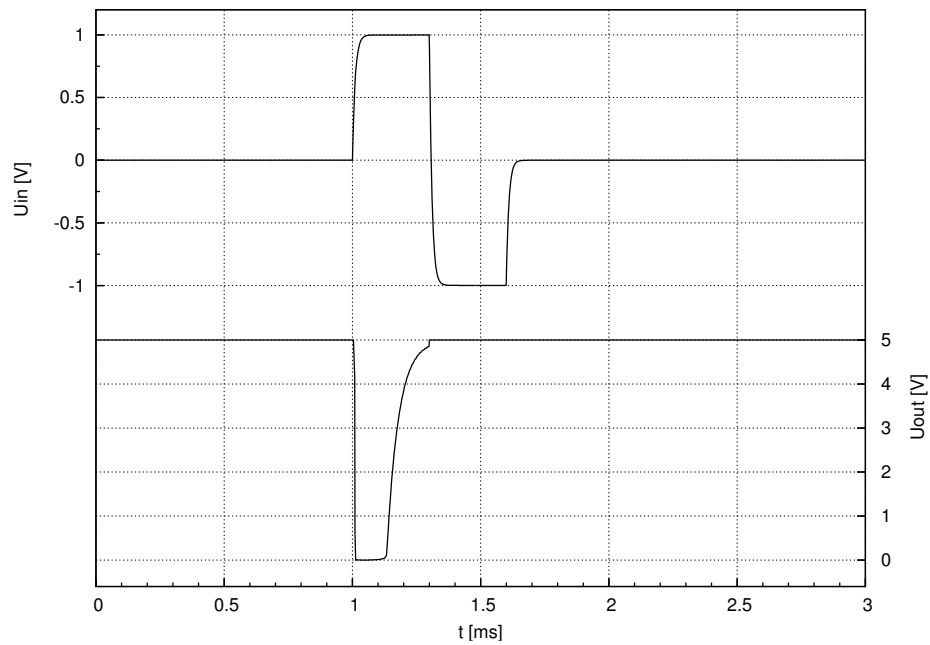
Na sliki 22 je prikazan posnetek izhodnega signala (v tem primeru enega sinhronizacijskega byta), narejen z analogno-digitalnim pretvornikom z vzorčevalno frekvenco 44100 kHz.

Analogni vhod je sestavljen iz impulznega ojačevalnika, ki deluje kot preprost analogno-digitalni pretvornik. Topologija vezja je enaka kot pri vezju originalne Galaksije, vrednosti komponent pa so bile ponovno izračunane (dodatek, stran 65), saj je tukaj pri originalnem načrtu napaka.

Primer vhodnega in izhodnega signala ojačevalnika za en vhodni impulz je prikazan na sliki 23.



Slika 22: Izhodni signal vmesnika za magnetofon (meritev)



Slika 23: Vhodni in izhodni signal impulznega ojačevalnika (SPICE simulacija)

4 Posebnosti vezja originalne Galaksije

4.1 Uporaba nedokumentiranih lastnosti mikroprocesorja

Originalna Galaksija ima v elektronskem vezju dve D pomnilni celici (v integriranem vezju 74LS74), ki se uporabljata za prepoznavanje stanja mikroprocesorja: prva celica prepozna interrupt request/acknowledge cikel in sodeluje v vezju za video sinhronizacijo (stran 23), druga pa prepozna zaključek M1 cikla in nadzira nalaganje podatkov v video shift register (stran 30). V obeh primerih je pravilno delovanje odvisno od minimalnih zakasnitev prehodov določenih izhodnih signalov mikroprocesorja Z80, ki jih proizvajalec v dokumentaciji ne zagotavlja. Tak pristop je bil najverjetneje izbran zaradi prihranka pri številu komponent na tiskanem vezju, saj bi bolj zanesljive izvedbe zahtevale najmanj eno dodatno integrirano vezje.

Zakasnitve signalov, ki so kritične za pravilno delovanje originalnega Galaksijinega vezja, so odvisne od odstopanj pri izdelavi integriranega vezja mikroprocesorja. Pri tem so navadno zakasnitve posameznih vezij v eni seriji podobne, med serijami pa se lahko (znotraj toleranc, ki jih v dokumentaciji zagotavlja proizvajalec) precej razlikujejo.

Ta lastnost vezja je eden od glavnih dejavnikov, ki preprečujejo gradnjo Galaksije po originalnih načrtih. S sodobnimi CMOS integriranimi vezji, ki so po specifikacijah proizvajalca sicer ekvivalentna originalnim NMOS vezjem, vezje ne deluje pravilno, saj so zakasnitve pri sodobnih vezjih bistveno manjše od originalnih (specifikacije v večini primerov podajajo le maksimalne vrednosti za določene zakasnitve).

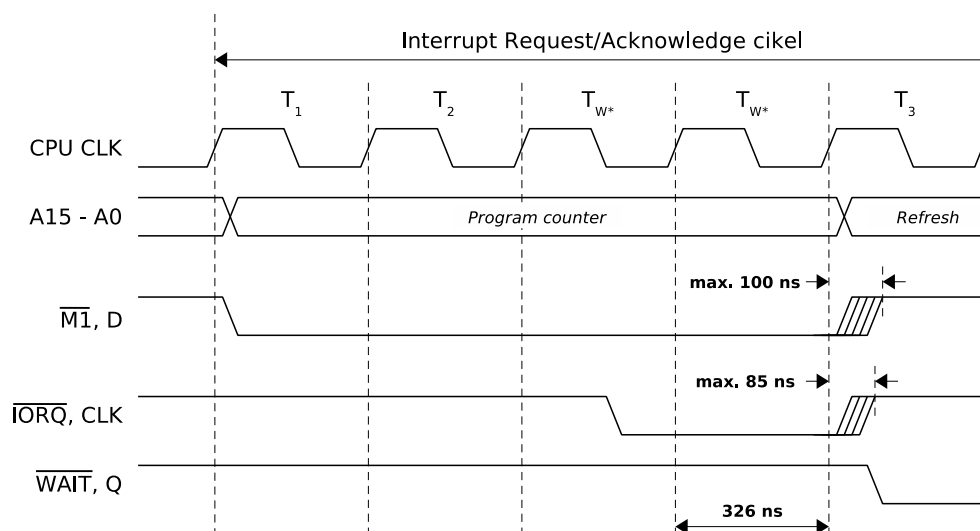
Replika Galaksije ima to pomanjkljivost rešeno na račun kompleksnejšega vezja.

4.1.1 Prva pomnilna celica

V to pomnilno celico se mora v tretjem T stanju prekinitvenega cikla zapisati logična 0. V ta namen je D vhod celice povezan na linijo $\overline{M1}$ mikroprocesorja, urin signal celice pa na \overline{IORQ} . Iz slike 24 je razvidno, da mora za pravilno delovanje imeti urin signal celice pozitivno fronto v času, ko je D signal še na logični 0⁴. Proizvajalec mikroprocesorja po drugi strani zagotavlja le

⁴Signal D mora biti še za čas t_{hold} pomnilne celice po pozitivni fronti ure na logični 0 za zanesljivo delovanje. V praksi so zakasnilni časi logičnih vezij velikostnega reda nekaj nanosekund in jih tukaj ne upoštevamo.

4.2 Vezava mikroprocesorjevega vodila



Slika 24: Časovni diagram delovanja prve D pomnilne celice iz vidika signalov mikroprocesorja (prikazani odzivni časi so za 4 MHz NMOS različico integriranega vezja Z8400 [10])

maksimalne časovne intervale med pozitivnim prehodom procesorske ure in prehodom signalov $\overline{M1}$ in \overline{IORQ} .

4.1.2 Druga pomnilna celica

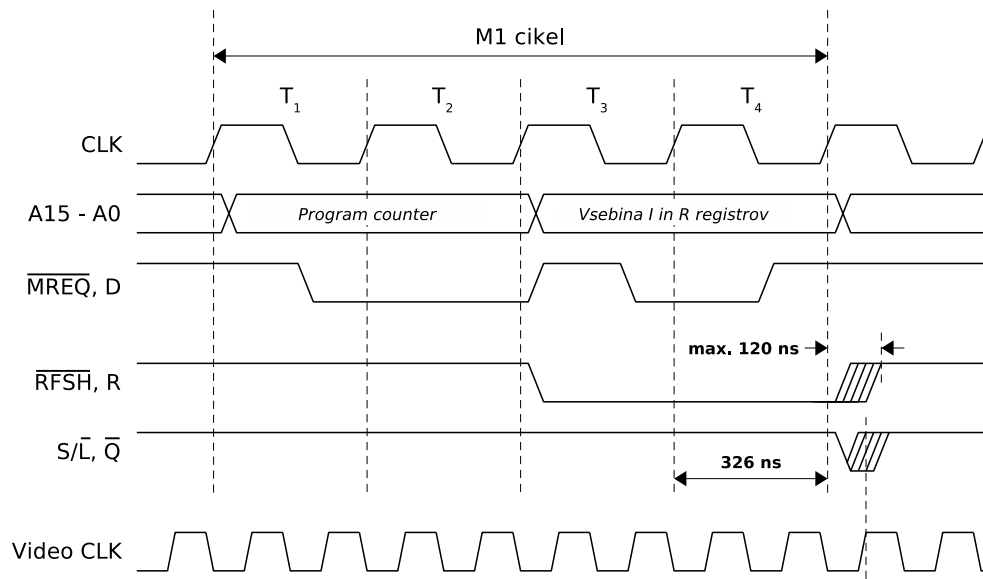
Pomnilna celica za detektiranje drugega dela M1 cikla mora imeti na izhodu logično 0 točno v času enega pozitivnega prehoda video ure. To je doseženo s priklopom procesorskega signala \overline{MREQ} na D vhod in \overline{RFSH} na reset vhod pomnilne celice. Procesor in pomnilna celica si urin signal v tem primeru delita.

Iz slike 25 je razvidno, da je pravilno delovanje vezja odvisno od zakasnitve med pozitivnim prehodom procesorske ure in \overline{RFSH} signala. Tudi tukaj proizvajalec zagotavlja le maksimalno zakasnitev, medtem ko je za delovanje kritična tudi minimalna zakasnitev.

4.2 Vezava mikroprocesorjevega vodila

Originalno vezje ima zelo preprost naslovni dekode. Poleg nepopolnega dekodiranja naslovov za latch in tipkovnico, ki je že bilo omenjeno, ima

4 POSEBNOSTI VEZJA ORIGINALNE GALAKSIJE



Slika 25: Časovni diagram delovanja druge D pomnilne celice iz vidika signalov mikroprocesorja (prikazani odzivni časi so za 4 MHz NMOS različico Z8400 integriranega vezja [10])

tudi to lastnost, da ne upošteva \overline{RD} in \overline{WR} signalov, ki se uporabljajo za razločevanje mikroprocesorjevih zahtev po pisanju oziroma branju iz naslovnega prostora.

Če program, ki se izvaja na mikroprocesorju, poizkusi pisati v del naslovnega prostora, ki ga zaseda naprava, ki ne podpira pisanja (ROM ali tipkovnica), sta na podatkovno vodilo priključeni dve napravi z izhodi v stanju nizke impedance. Zaradi tehnologije integriranih vezij, uporabljenih v originalnem vezju (NMOS za mikroprocesor in EPROM, ostalo low-power Schottky), ta pomanjkljivost ni predstavljala nevarnosti za življenjsko dobo vezja. Izhodi integriranih vezij v omenjenih tehnologijah imajo namreč v stanju logične enice relativno visoko izhodno upornost (velikostnega reda kiloohmov), kar omejuje kratkostične tokove, ki lahko v tem primeru tečejo po vodilu.

Izhodi sodobnih vezij v tehnologiji CMOS pa imajo po drugi strani nizko izhodno upornost v obeh definiranih stanjih. Replika Galaksije ima zaradi tega v podatkovno vodilo vezane izolacijske upore (R10-R17 in R30), ki v omenjenih primerih omejijo tok po vodilu.

4.3 Vezava tipkovnice

Tipke Galaksijine tipkovnice so razporejene po matriki 8 x 7 tako, da njihovi naslovi v naslovnem prostoru mikroprocesorja čim bolj ustrezajo razporeditvi ustreznih znakov v ASCII kodni tabeli (slika 7). Na ta način je lahko programska koda v operacijskem sistemu, ki pretvarja kodo tipke (angl. scan code) v ASCII kodo znaka⁵, ki ustreza tipki, znatno manjša, saj za pretvorbo niso potrebne relativno obsežne tabele.

Ker fizična razporeditev tipk v splošnem ne ustreza razporeditvi znakov po ASCII tabeli, je ta lastnost bistveno povečala kompleksnost tiskanega vezja tipkovnice. To je bilo v prvotni različici enostransko in je zaradi tega potrebovalo veliko število ročno narejenih prevezav.

Bralni pomnilnik EPROM je poleg mikroprocesorja in delovnega pomnilnika edino vezje z veliko stopnjo integracije in kot tak eden dražjih delov Galaksije. Iz tega vidika postane razumljivo, da je bila povečana kompleksnost izdelave dober kompromis za manjši obseg operacijskega sistema, ki je omogočil uporabo cenejšega EPROM vezja z manjšo kapaciteto. Poleg tega potrebna ročna izdelava prevezav ni vplivala na prodajno ceno kompletov za samogradnjo računalnika, kar je bil ob načrtovanju računalnika edini predvideni način prodaje.

4.4 Uporaba R registra

Uporaba mikroprocesorja za izris slike na zaslon za zmanjševanje kompleksnosti strojne opreme je bil relativno pogost pristop pri prvih hišnih računalnikih. Podobne pristope uporabljajo na primer tudi računalniki Sinclair ZX80 in ZX81. Posebnost Galaksije je, da za ta namen uporablja funkcijo osveževanja dinamičnih pomnilnikov (druga polovica M1 cikla, R register). Sinclairjevi računalniki po drugi strani za prenos podatkov iz delovnega pomnilnika v shift register uporabljajo prvo polovico M1 cikla [11].

⁵Funkcija KEY na naslovih od 0x0ce6 do 0x0d99

5 Posebnosti operacijskega sistema

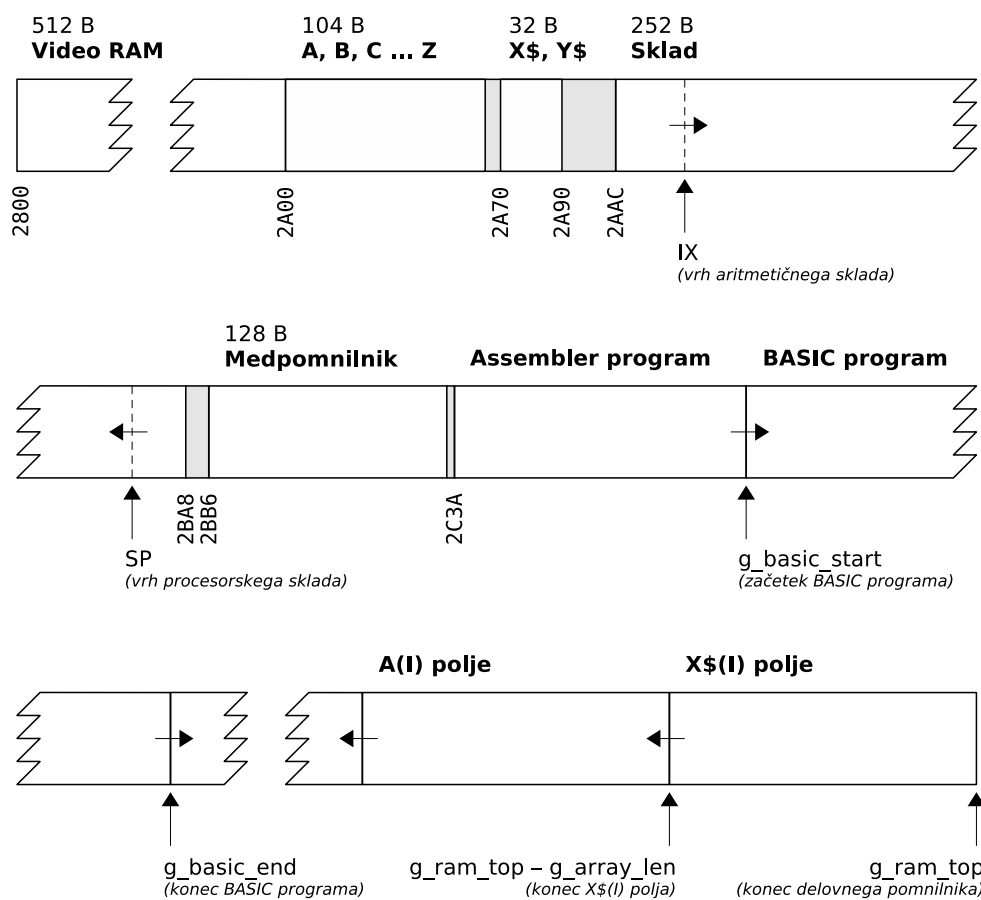
Kot je že bilo omenjeno, je bilo zmanjševanje obsega operacijskega sistema, shranjenega v EPROM pomnilniku, ena od učinkovitih metod nižanja cene celotnega sistema. Zaradi tega vsebuje operacijski sistem veliko optimizacij, ki zmanjšujejo obseg kode, po drugi strani pa zelo otežujejo reverzni inženiring in zmanjšujejo preglednosti kode. Predvsem je zelo zmanjšana uporabnost avtomatskih disasemblerskih programov, saj nekateri od opisanih pristopov povzročijo izgubo sinhronizacije disasemblerja s kodo, ki jo izvaja mikroprocesor. V tem primeru je potrebno ročno preverjanje rezultatov in disasembliranje strojne kode po delih, za katere je znana vstopna točka mikroprocesorja.

Operacijski sistem Galaksije (ROM A) lahko razdelimo na naslednje komponente:

- rutine za inicializacijo strojne opreme,
- video gonilnik,
- gonilnik za tipkovnico in osnovno terminalske emulacijo,
- koda za modulacijo in demodulacijo signala magnetofona,
- rutine za aritmetiko s plavajočo vejico ter
- BASIC interpreter.

Organizacija delovnega pomnilnika je prikazana na sliki 26.

V literaturi najdemo podatek, da je operacijski sistem osnovan na Microsoft Level 1 BASIC-u [12]. Pri tem gre najverjetneje za napako. Primerjava z ednim BASIC interpreterjem podjetja Microsoft velikosti 4 kB (BASIC za mikroračunalnik Altair) pokaže, da sta si programski kodi zelo različni: Altair BASIC uporablja le ukaze za mikroprocesor Intel 8080, medtem ko bistveni deli Galaksijinega operacijskega sistema uporabljajo tudi ukaze, ki so specifični za mikroprocesor Zilog Z80. Poleg tega koda Altair BASIC-a [13] izkorišča tudi dejstvo, da se izvaja v RAM-u, zaradi česar je na določenih mestih njeno delovanje odvisno od sposobnosti spreminjanja kode med delovanjem programa. Bistveno se razlikujejo tudi uporabljene podatkovne strukture (na primer zapis števil s plavajočo vejico, zapis BASIC programa, itd.). Edina podobnost med Galaksijino in Altairjevo kodo je restart `0x08` (na Galaksiji `0x10`), ki opravlja podobno funkcijo in je podobno implementiran.



Slika 26: Organizacija delovnega pomnilnika (potemnjena polja zasedajo sistemske spremenljivke)

5 POSEBNOSTI OPERACIJSKEGA SISTEMA

Večja verjetnost je, da je Galaksijin operacijski sistem osnovan na operacijskem sistemu mikroračunalnika Tandy TRS-80 Model I. Prva različica operacijskega sistema (prirejeni Tiny Basic avtorja Li-Chen Wanga) je uporabljala 4 kB velik ROM in ima zelo podobne lastnosti kot Galaksijin operacijski sistem: enaka sporočila o napakah („HOW?“, „WHAT?“, „SORRY“), podobno modulacijo za shranjevanje podatkov na magnetofon, mestoma identično strojno kodo za izvajanje operacij s plavajočo vejico, itd. Tudi sicer so strojne zmogljivosti TRS-80 podobne Galaksiji (enak psevdografični način, enaka vezava tipkovnice na procesorsko vodilo, itd.).

Za potrebe reverznega inženiringa Galaksijinega operacijskega sistema je bil izdelan nov disasemblerski program *z80dasm*, ki se nahaja v prilogi. Vsebina ROM-a A v obliki asemblerske izvirne kode, izdelane z *z80dasm*, se nahaja v prilogi.

V nadaljevanju so opisani nekateri od uporabljenih optimizacijskih pristopov s primeri iz vsebine ROM-a A.

5.1 Večličnost programske kode

Nekatere dele strojne kode, zapisane v EPROM pomnilniku, mikroprocesor ob različnih priložnostih interpretira različno.

Določeni nizi strojnih ukazov se na primer interpretirajo različno, glede na naslov skoka. V tabeli 6 je prikazan primer takega dela kode, ki se v treh različnih primerih interpretira na tri različne načine. V primeru skoka na naslov 0x0390 se v register HL zapiše vrednost 0x0f0e, pri skoku na naslov 0x0393 vrednosti 0x0f9b in pri skoku 0x0396 vrednost 0x0feeh.

Na podoben način je na več mestih v kodi operacijskega sistema skrajšana koda. V večini primerov gre za nadomestilo skoka čez nezaželeni ukaz z ukazom za nalaganje vrednosti v neuporabljeni register. Na tak način se za vsako vejo izvajanja prihrani en byte programske kode v primerjavi z uporabo procesorskega ukaza *jr*.

V določenih primerih se strojna koda mikroprocesorja Z80 interpretira tudi kot podatkovna struktura.

Prvi tak primer najdemo na naslovu 0x0098 (tabela 7), kjer se koda interpretira tudi kot ASCII niz. Znaki iz zgornje polovice kodne tabele ASCII se interpretirajo izključno kot 8-bitni load (1d) ukazi z registrskimi operandi [14], kar poenostavi izbiro tekstovnega niza, ki ustreza mikroprocesorski kodi.

5.1 Večličnost programske kode

Adr.	Hex	Procesorjeva interpretacija ukazov		
0x0390	0x2e	1 → ld 1,0eh		
0x0391	0x0e			
0x0392	0x01	ld bc,9b2eh		
0x0393	0x2e	2 → ld 1,9bh		
0x0394	0x9b			
0x0395	0x01	ld bc,ee2eh	ld bc,ee2eh	
0x0396	0x2e	3 → ld 1,eeh		
0x0397	0xee			
0x0398	0x26	ld h,0fh	ld h,0fh	ld h,0fh
0x0399	0x0f			

Tabela 6: Primer večličnosti strojne kode v Galaksijinem operacijskem sistemu. Register HL se nastavi na različne vrednosti glede na vstopno točko (→) mikroprocesorja

Adr.	Hex	ASCII	Procesorjeva interpretacija ukazov
0x0098	0x42	B	ld b,d
0x0099	0x52	R	ld d,d
0x009a	0x45	E	ld b,l
0x009b	0x41	A	ld b,c
0x009c	0x4b	K	ld c,e
0x009d	0x00	NUL	nop

Tabela 7: Primer uporabe strojne kode kot ASCII niza „BREAK“

5 POSEBNOSTI OPERACIJSKEGA SISTEMA

Adr.	Hex	Podatki	Procesorjeva interpretacija ukazov
0x00a0	0x00	M=0x800000	nop
0x00a1	0x00		nop
0x00a2	0x80	S=+1	add a,b
0x00a3	0x00	E=0x01	nop

Tabela 8: Primer uporabe strojne kode kot 4-bytnega zapisa konstantne vrednosti s plavajočo vejico (dodatek, stran 67)

Drugi primer se nahaja na naslovu 0x00a0 (tabela 8). Tukaj se strojna koda interpretira tudi kot zapis števila s plavajočo vejico $+1 \cdot 0x800000 \cdot 2^{0x01-24} = 1.0$.

5.2 Večličnost podatkovnih struktur

V EPROM pomnilniku se nahaja vrsta konstantnih podatkovnih struktur, ki so večinoma pomešane med strojno kodo. Nekatere od njih operacijski sistem interpretira na več različnih načinov.

Tak primer najdemo v tabeli ASCII kod za tipkovnico na naslovih od 0x0d70 do 0x0d99 (tabela 9).

Za pretvorbo kode tipke v ASCII kodo znaka se uporablja tabela `KEY_SHIFT_SYM_TABLE` na naslovu 0x0d70. V njej so shranjene ASCII kode znakov v vrstnem redu, ki ustreza naslovu tipk v naslovnem prostoru mikroprocesorja (slika 7). Prva vrstica v tabeli na primer ustreza tipki na naslovu 0x1f. Prvi stolpec tabele vsebuje ASCII kode znakov, ki ustrezajo pritisku na tipko skupaj s pritisnjeno tipko *SHIFT*. Drugi stolpec pa vsebuje ASCII kode, ki ustrezajo pritisku na tipko brez tipke *SHIFT*.

Za vnos jugoslovanskih znakov se uporablja druga tabela `KEY_SHIFT_YU_TABLE` na naslovu 0x0d94. Ta vsebuje ASCII znake, ki ustrezajo tipkam, ki ob pritisnjeni tipki *SHIFT* vnesejo šumnike.

Tabela `KEY_SHIFT_YU_TABLE` v večji tabeli `KEY_SHIFT_SYM_TABLE` zaseda mesta, ki ustrezajo tipkama na naslovih 0x31 in 0x32. Ti naslovi pripadajo tipkama *BREAK* in *REPEAT*, ki jih koda operacijskega sistema obravnava posebej in zaradi česar ASCII koda, ki jima pripada po tabeli `KEY_SHIFT_YU_TABLE`, ni pomembna.

5.2 Večličnost podatkovnih struktur

Adr.	Hex	SYM_TABLE			YU_TABLE	
0x0d70	0x20	0x1f	□	□		
0x0d71	0x20					
0x0d72	0x5f	0x20	_	0		
0x0d73	0x30					
0x0d74	0x21	0x21	!	1		
0x0d75	0x31					
0x0d76	0x22	0x22	"	2		
0x0d77	0x32					
0x0d78	0x23	0x23	#	3		
0x0d79	0x33					
0x0d7a	0x24	0x24	\$	4		
0x0d7b	0x34					
0x0d7c	0x25	0x25	%	5		
0x0d7d	0x35					
0x0d7e	0x26	0x26	&	6		
0x0d7f	0x36					
0x0d80	0xbf	0x27	□	7		
0x0d81	0x37					
0x0d82	0x28	0x28	(8		
0x0d83	0x38					
0x0d84	0x29	0x29)	9		
0x0d85	0x39					
0x0d86	0x2b	0x2a	+	;		
0x0d87	0x3b					
0x0d88	0x2a	0x2b	*	:		
0x0d89	0x3a					
0x0d8a	0x3c	0x2c	<	,		
0x0d8b	0x2c					
0x0d8c	0x2d	0x2d	-	=		
0x0d8d	0x3d					
0x0d8e	0x3e	0x2e	>	.		
0x0d8f	0x2e					
0x0d90	0x3f	0x2f	?	/		
0x0d91	0x2f					
0x0d92	0x0d	0x30	CR	CR		
0x0d93	0x0d					
0x0d94	0x58	0x31	X	C	Č	shift X
0x0d95	0x43				Ć	shift C
0x0d96	0x5a	0x32	Z	S	Ž	shift Z
0x0d97	0x53				Š	shift S
0x0d98	0x0c	0x33	FF	NUL		
0x0d99	0x00					

Tabela 9: Primer večličnosti podatkovnih struktur. Del pomnilnika se interpretira kot vsebina dveh različnih tabel

5.3 Organizacija programske kode

Operacijski sistem zaradi optimizacije ni strogo razdeljen na zaključene enote - programske funkcije. Namesto standardnih klicev funkcij z ukazi `call` in `ret` se uporablja kombinacija klicev z ukazom `call` in skokov `jp` oziroma `jr`. S tem se je pri razvoju prihranilo veliko število ukazov za zaključek funkcije `ret`, ki bi bili sicer potrebni, pridobilo pa se je tudi na hitrosti delovanja operacijskega sistema (tako je potrebnih manj operacij s skladom).

Na sliki 27 je prikazana tipična razporeditev funkcij:

- Pogosto ima funkcija več vstopnih točk. Odvisno od vstopne točke se v določeni podrobnosti spremeni učinek funkcije. Na primer: Prva vstopna točka opravi neko operacijo na nizu na naslovu `DE+1`, medtem ko druga vstopna točka opravi enako operacijo na naslovu `DE+0` (`f1a`, `f1b` in `f1c` na sliki 27).
- Funkcija `f2` tik pred zaključkom kliče funkcijo `f1a`. Namesto funkcijskega klica `call f1a` je funkcija `f2` v ROM pomnilniku nameščena tik pred funkcijo `f1`. Na ta način mikroprocesor izvajanje brez skoka nadaljuje v naslednji funkciji. Zadnji ukaz `ret` pa vrne izvajanje procesorja v del kode, ki je klical funkcijo `f2`. Ker to prihrani celoten `call` ukaz (v najboljšem primeru 3 byte), je ta pristop nekajkrat uporabljen tudi takrat, ko tako klicana funkcija ni najbolj primerna.⁶
- Funkcije imajo pogosto enak zaključni del kode (tik pred ukazom `ret`). Namesto da bi bila ta koda ponovljena v vsaki funkciji, je zapisana samo enkrat (`fend` na sliki 27), ostale funkcije pa so ali nameščene v ROM pomnilniku tako, da mikroprocesor to kodo doseže brez skoka (`f1` in `f2`) ali pa nanjo skočijo z ukazom `jp` ali `jr` (`f3`).

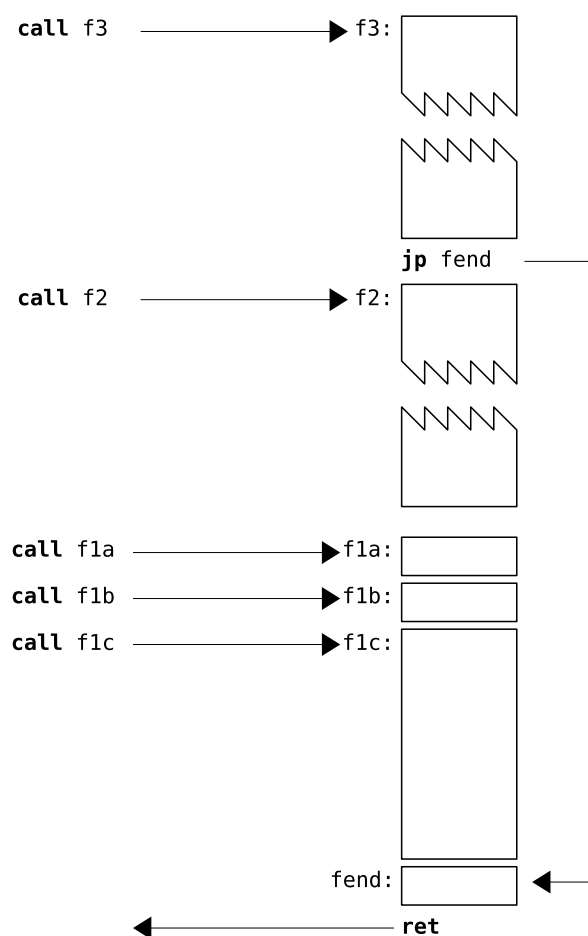
Velikokrat se v kodi uporabljajo tudi stranski učinki funkcij. Na primer funkcija `CLEAR_LINE`, katere glavna naloga je izbris ene zaslonske vrstice, se v nekaterih primerih uporablja tudi za premik kazalca v registru `HL`.

5.4 Uporaba procesorskih registrov

Posledica relativno nestrukturirane programske kode je konsistentna uporaba registrov. Vzorcev v tabeli 10 se drži velika večina kode operacijskega sistema.

⁶Na primer klic `SAVE_WORD` za shranjevanje kontrolne vsote na koncu rutine `SAVE`, ko bi bil bolj primeren klic `SAVE_BYTE`, ki ne bi na trak shranil neuporabnega byta.

5.4 Uporaba procesorskih registrov

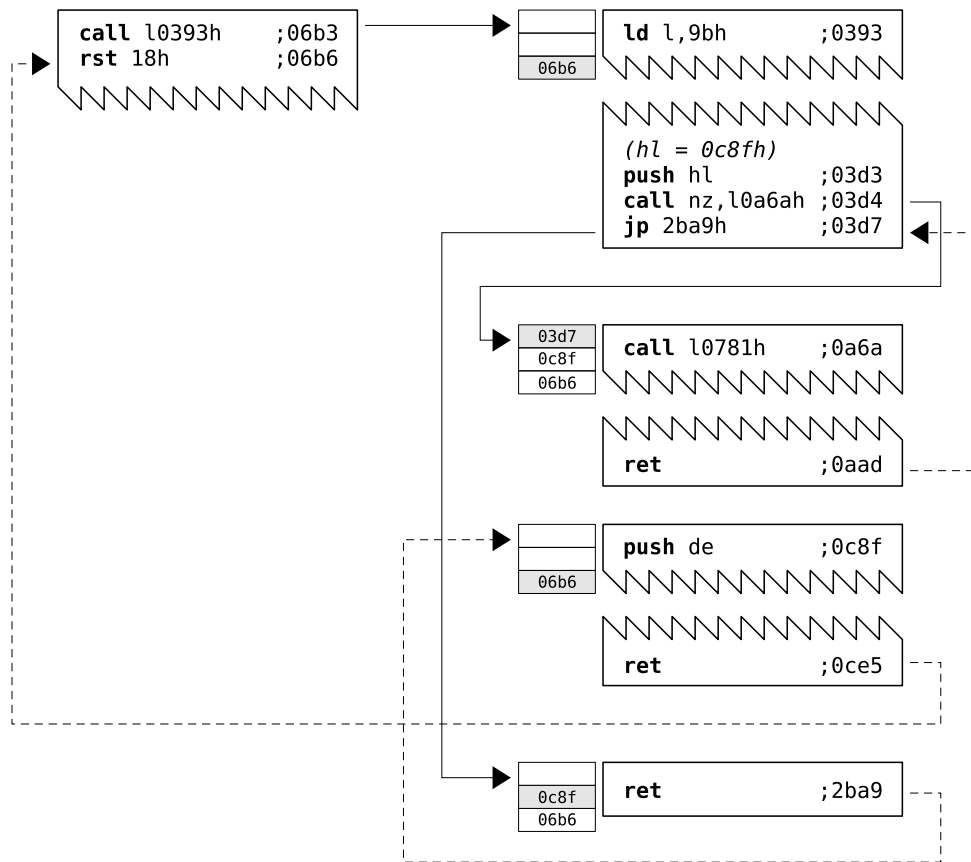


Slika 27: Tipična razporeditev funkcij v operacijskem sistemu

Register	Vsebina
DE	Kazalec na trenutni znak BASIC interpreterja
HL	Kazalec na BASIC spremenljivko
IX	Kazalec na vrh aritmetičnega sklada
IY	Kazalec na funkcijo, ki se izvrši ob video prikinitvi
HL' C' HL	Prvi operand operacije s plavajočo vejico
DE' B' DE	Drugi operand operacije s plavajočo vejico

Tabela 10: Tipična uporaba procesorskih registrov v operacijskem sistemu

5 POSEBNOSTI OPERACIJSKEGA SISTEMA



Slika 28: Primer spreminjanja povratnega naslova funkcije na skladu

5.5 Procesorski sklad

Koda operacijskega sistema pogosto neposredno spreminja vrednosti na procesorskem skladu (angl. stack), kar drugod ni običajno. Eden od primerov manipulacije sklada je prikazan na sliki 28. Tukaj funkcija na naslovu 0x0393 kliče funkcijo 0c8f tako, da na sklad potisne njen naslov, naslednjo funkcijo (0x2ba9) pa kliče preko ukaza `jp` namesto `ret`. Na ta način `ret` ukaz v funkciji 0x2ba9 skoči na klicano funkcijo 0x0c8f, namesto da bi vrnil izvajanje v 0393.

Sklad se uporablja tudi za prenašanje kazalcev na argumente določenih funkcij. Primer je prikazan na sliki 29.

Tukaj funkcija `READ_PAR` uporabi naslov, ki ga na vrh sklada shrani ukaz `rst`, kot naslov dveh operandov (na naslovih 0x0134 in 0x0135).

Ob koncu izvajanja funkcija prišteje 2 kazalcu, ki ga je na začetku prebrala

5.6 Shranjevanje podatkov na magnetofon

rst 18h	;0133	Call READ_PAR
db '('	;0134	
db 1015dh-\$-1	;0135	
rst 08	;0136	
inc h1	;0137	
add h1,h1	;0138	

Slika 29: Primer uporabe sklada za prenos kazalca na argumente funkcije READ_PAR

z vrha sklada, in ga vrne na vrh sklada. Tako **ret** ukaz na koncu vrne izvajanje procesorja na naslov 0x0136.

Podobno je implementirana tudi funkcija **BASIC_ERROR**, kjer vrednost na vrhu sklada kaže na sporočilo o napaki.

Na ta način se je prihranila dodatna strojna koda, ki bi bila sicer potrebna ali za nalaganje argumentov funkcije v registre ali za nalaganje kazalca na podatkovno strukturo. Ta poenostavitev v primeru velikega števila klicev določene funkcije več kot odtehta potrebno dodatno kodo v sami funkciji.

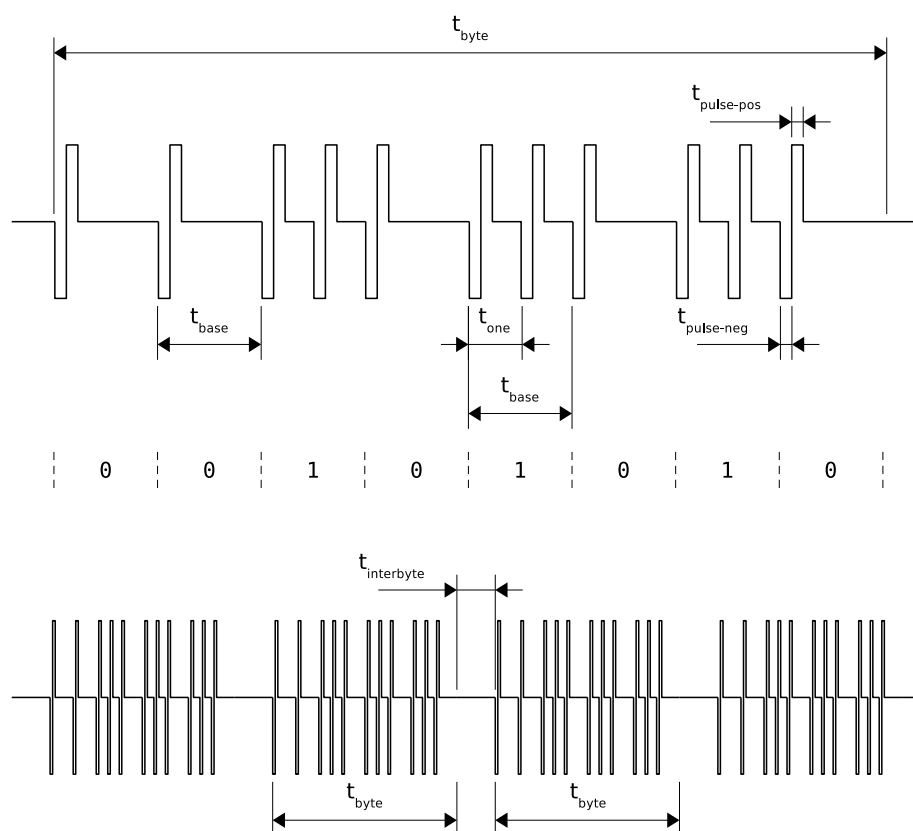
Tako prepletanje strojne kode in argumentov funkcij je eden od glavnih razlogov za omenjene težave z avtomatskimi disasemblerskimi programi. Ne-navadni načini uporabe procesorskega sklada pa zelo otežujejo tudi uporabo programskih simulatorjev za raziskovanje funkcij operacijskega sistema, saj zabrišejo podatke o klicni poti funkcij (angl. backtrace).

5.6 Shranjevanje podatkov na magnetofon

Za shranjevanje podatkov na magnetofon je uporabljena preprosta impulzna modulacija. Časovni diagram signala je prikazan na sliki 30, podatki o časovnih intervalih v tabeli 11, logični pomen posameznih shranjenih bytov pa v tabeli 12.

Tipično je dosežena hitrost prenosa podatkov približno 330 bit/s. Izbira preproste modulacije in nizka hitrost prenosa je najverjetneje posledica omejenega prostora za modulacijske in demodulacijske rutine v EPROM pomnilniku, saj mikroračunalniki z enako strojno opremo in obsežnejšim ROM pomnilnikom dosegajo bistveno višje hitrosti prenosa (na primer Sinclair Spectrum tipično 1500 bit/s).

5 POSEBNOSTI OPERACIJSKEGA SISTEMA



Slika 30: Časovni diagram modulacije, ki se uporablja za shranjevanje podatkov na magnetofonski trak

Symbol	Komentar	Min	Tipično	Max	Enota
$t_{pulse-pos}$	Trajanje pozitivnega impulza	140 45	650 210		T μs
$t_{pulse-neg}$	Trajanje negativnega impulza		650 210		T μs
t_{base}	Trajanje zapisa enega bita	7800 2.5	9200 3.0	16000 5.2	T ms
t_{one}	Zakasnitev med prvim in drugim impulzom	4400 1.4	4600 1.5	7400 2.4	T ms
t_{byte}	Trajanje zapisa enega byta		74000 24.0		T ms
$t_{interbyte}$	Zakasnitev med zapisoma dveh bytov	8200 2.7	13000 4.3		T ms

Tabela 11: Vrednosti simbolov (k sliki 30)

5.6 Shranjevanje podatkov na magnetofon

Odmik	Dolžina	Komentar
0x00	0x01	Start byte (0xa5)
0x01	0x02	Naslov prvega podatkovnega byta (najnižji byte naslova najprej)
0x03	0x02	Naslov zadnjega podatkovnega byta + 1 (najnižji byte naslova najprej)
0x05	N	Podatkovni byti
0x05 + N	0x01	Kontrolna vsota (število, ki ga je potrebno dodati vsoti predhodnih bytov po modulu 256, da dobimo število 0xff)

Tabela 12: Pomen posameznih bytov v zapisu podatkov za shranjevanje na magnetofon

Naslov	Dolžina	Vsebina	Komentar
0x2c36	0x02	0x2c3a	Naslov prvega byta BASIC programa (najnižji byte naslova najprej)
0x2c38	0x02	0x2c3a+N	Naslov zadnjega byta BASIC programa + 1 (najnižji byte naslova najprej)
0x2c3a	N		BASIC program

Tabela 13: Podatkovni del zapisa pri shranjevanju BASIC programa na magnetofon

Operacijski sistem ne razlikuje med shranjevanjem podatkov, strojne kode in BASIC programa. Podatkovna struktura v tabeli 12 omogoča le shranjevanje vsebine poljubnega dela naslovnega prostora mikroprocesorja.

V primeru shranjevanja BASIC programa operacijski sistem shrani na magnetofon podatke od naslova 0x2c36 do konca BASIC programa. Pri kasnejšem nalaganju shranjenih podatkov se tako prepišeta sistemski spremenljivki BASIC_START in BASIC_END na naslovih 0x2c36 in 0x2c38, kar operacijskemu sistemu omogoča lociranje naloženega programa.

6 Zaključek

Izdelava replike Galaksije je pokazala, da izdelava nadomestila tudi za tako majhen elektronski sistem, kot je Galaksija, ni enostavna naloga, ki zahteva znanje tako analogne in digitalne elektronike kot tudi programiranja v zbirniku. Po drugi strani pa to delo dokazuje, da je izdelava delujoče replike mogoča le na podlagi ohranjene dokumentacije in programske opreme, tudi ko za analizo ni na voljo delujočega originala.

Pred izdelavo in načrtovanjem same replike je bilo potrebno rešiti vrsto stranskih problemov: reverzni inženiring originalnega elektronskega vezja je bil seveda načrtovan, ni pa bila načrtovana neuporabnost trenutnih orodij za preučevanje strojne kode za procesor Z80, ki je zahtevala ovinek z glavne poti in izdelavo uporabnega Z80 disasemblerja. Zaradi težav pri razumevanju delovanja operacijskega sistema je bilo potrebno tudi preučevanje delujočega sistema pri poganjanju testnih programov, ne le statična analiza vsebine ROM pomnilnikov. Razvoj testnih programov pa je zahteval najprej izdelavo razvojnih orodij, s katerimi je bilo te testne programe mogoče razviti.

Težave je povzročala tudi izdelava tipkovnice replike. Zaradi velike prevlade standardnih tipkovnic za osebne računalnike v okviru sredstev, ki so bila na razpolago, ni bilo mogoče izdelati ustreznega nadomestila za originalno Galaksijino tipkovnico. Tako ima izdelana sodobna replika Galaksije slabšo, s folijo prekrito tipkovnico (profesionalna tipkovnica je bila sicer ena od redkih prednosti, ki jih je iz tehničnega vidika Galaksija imela pred tekmeči).

Te stranpoti so prispevale k temu, da rezultati pokrivajo še nekoliko širše področje, kot je bilo prvotno načrtovano v ciljnih naloge. Posledica pa je tudi nekoliko obsežnejši spisek prilog.

Danes se poleg velikega pomena dokumentacije predvsem poudarja pomen pisanja razumljivih in preprostih programov pred visoko optimiziranimi in težko razumljivimi. Rezultati dela to potrjujejo, saj se je prekomerno optimizirana strojna koda izkazala za zelo težko razumljivo. Hkrati pa se je z njo povezana ohranjena dokumentacija izkazala za praktično neuporabno. Nepričakovana posledica optimizacije je bila tudi že zgoraj omenjena zmanjšana uporabnost sodobnih disasemblerskih orodij, tako da so določeni deli kode zahtevali ročno prevajanje strojne kode v zbirnik na podlagi tabel strojnih ukazov.

Kljub boljši dokumentaciji so bile pri strojni opremi ugotovitve podobne kot pri programski: tudi tukaj se je izkazalo, da je poleg dokumentacije

pomembno tudi načrtovanje preprostih in razumljivih vezij. Prekomerna optimizacija z vidika števila elektronskih komponent, kot je na primer Galaksijina uporaba sekvenčnih vezij namesto kombinacijskih, podobno kot pri programski opremi veliko prispeva k zahtevnosti dela. Kar nekajkrat je ohranjena dokumentacija zaradi tiskarskih napak prej otežila kot olajšala delo in pogosto je bilo potrebno navzkrižno primerjanje podatkov iz različnih virov (na primer iz vezalnega načrta in načrta tiskanega vezja).

Po drugi strani rezultati kažejo na pozitiven vpliv izključne uporabe standardnih elektronskih komponent (predvsem integriranih vezij). Izdelava podobne replike mikroračunalnika, ki za svoje delovanje uporablja namensko integrirano vezje (angl. ASIC), bi bila bistveno težja naloga.

Pri ohranjanju dokumentacije se je pokazal tudi velik pomen javne objave dokumentacije. Veliko mikroračunalnikov iz tega obdobja se ni ohranilo predvsem, ker je dokumentacija izginila skupaj s podjetjem, ki jo je skrbno varovalo. Pri ohranjanju programske opreme pa se je podobno pokazal pozitiven vpliv odsotnosti zaščit proti nepooblaščenemu kopiranju, ki so sicer pri podobnih mikroračunalnikih botrovale izgubi velikega dela programske opreme.

7 Literatura

Literatura

- [1] *Hišni računalnik*. Ljubljana, Mladinska knjiga, 1984, str. 42-46.
- [2] AntoniĆ, V.: Napravi i ti računar Galaksija. *Računari u vašoj kući*, januar 1984, str. 50-56.
- [3] *ZX81 Assembly Instructions*. Sinclair Research Ltd, 1981.
- [4] *Servicing Manual for ZX Spectrum*. Thorn Datatech Ltd, Sinclair Research Ltd, 1984.
- [5] *Commodore 64 Service Manual*. Commodore Business Machines Inc, 1985.
- [6] *Z80 Family CPU User Manual*. San Jose, ZiLOG Inc., 2005.
- [7] Benson, K. B.: *Television Engineering Handbook*. McGraw-Hill Inc., 1985.
- [8] Wedam, A.: *Radiotehnika: ojačevalniki in sprejemniki*. Ljubljana, Državna založba Slovenije, 1955.
- [9] *Bandwidth Versus Video Resolution*. Maxim Integrated Products, Dallas Semiconductor, 2005. Dosegljivo na naslovu http://www.maxim-ic.com/appnotes.cfm/appnote_number/750 (citirano 12. 6. 2007).
- [10] *Z8400/Z84C00 NMOS/CMOS Z80 CPU Central Processing Unit: Product specification*. ZiLOG Inc.
- [11] Rifter, W.: *ZX81 Video Display System*. 1996. Dosegljivo na naslovu <http://home.germany.net/nils.eilers/zx81/wilfvid.htm> (citirano 24. 6. 2007).
- [12] Ristanović, D.: *Računar Galaksija*. Dosegljivo na naslovu <http://user.sezampro.yu/dejanr/galaks.htm> (citirano 13. 7. 2007).
- [13] Harris, R.: *Altair BASIC 3.2 (4K) - Annotated Disassembly*. Dosegljivo na naslovu <http://www.interact-sw.co.uk/altair/index2.html> (citirano 13. 7. 2007).

LITERATURA

- [14] Dinu, C.: *Decoding Z80 opcodes*. Dosegljivo na naslovu <http://z80.info/decoding.htm> (citirano 25. 6. 2007).
- [15] Ristanović, D.: *Računar Galaksija Uputstvo za upotrebu*.
- [16] Ristanović, D.: Galaksija bez tajni. *Računari*, julij 1984, str. 53-63.

8 Priloge

1. Vezalni načrt matične plošče,
2. vezalni načrt tipkovnice,
3. pregledna shema tiskanega vezja matične plošče,
4. pregledna shema tiskanega vezja tipkovnice in
5. CD z naslednjo vsebino (natančnejši opis vsebine se nahaja v datoteki v korenskem direktoriju):

```
/
├── /besedilo.....Besedilo (Postscript, PDF)
├── /programi
│   ├── /programi/orodja.....Orodja za okolje UNIX (izvorna koda)
│   ├── /programi/orodja/galaksija-tools.....Razvojna orodja za Galaksijo
│   ├── /programi/orodja/z80dasm.....Pametni Z80 disassembler
│   └── /programi/gtp.....Zbirka programov za Galaksijo (GTP)
├── /clanki.....Zbirka dokumentacije o Galaksiji
│   ├── /clanki/Brez tajni
│   ├── /clanki/Disassembly
│   ├── /clanki/Gradnja
│   ├── /clanki/Moj mikro
│   ├── /clanki/Navodila
│   ├── /clanki/Racunari1
│   ├── /clanki/Racunari2
│   └── /clanki/Rom B
├── /nacrti
│   ├── /nacrti/tiskanina.....Načrt tiskanine (PCB, Gerber, Postscript, PDF)
│   ├── /nacrti/ohisje.....Načrt ohišja (ODG, PDF)
│   ├── /nacrti/nalepke.....Nalepke za tipkovnico in konektorje (ODG, PDF)
│   └── /nacrti/vezje.....Shema električnega vezja (Gschem, Postscript, PDF)
├── /spice.....SPICE datoteke, uporabljene pri načrtovanju
├── /rom
│   ├── /rom/disassembly.....Delno komentirana izvorna koda ROM-a A (HTML)
│   └── /rom/bin.....Binarne slike ROM pomnilnikov
│       ├── /rom/bin/original.....Originalna Galaksija
│       └── /rom/bin/nova.....Replika
└── /video.....Demonstracijski video posnetek (AVI)
```

A Napetostni inverter

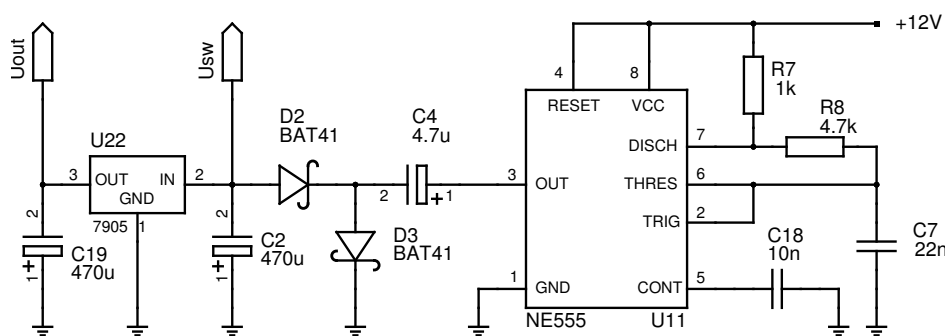
Napetostni inverter mora proizvajati stabilizirano napajalno napetost -5V v vseh pogojih delovanja video ojačevalnika.

SPICE simulacija video ojačevalnika v obeh skrajnih primerih (popolnoma bel zaslon in popolnoma črn zaslon) je pokazala, da video ojačevalnik za napajanje potrebuje od 25 do 34mA toka.

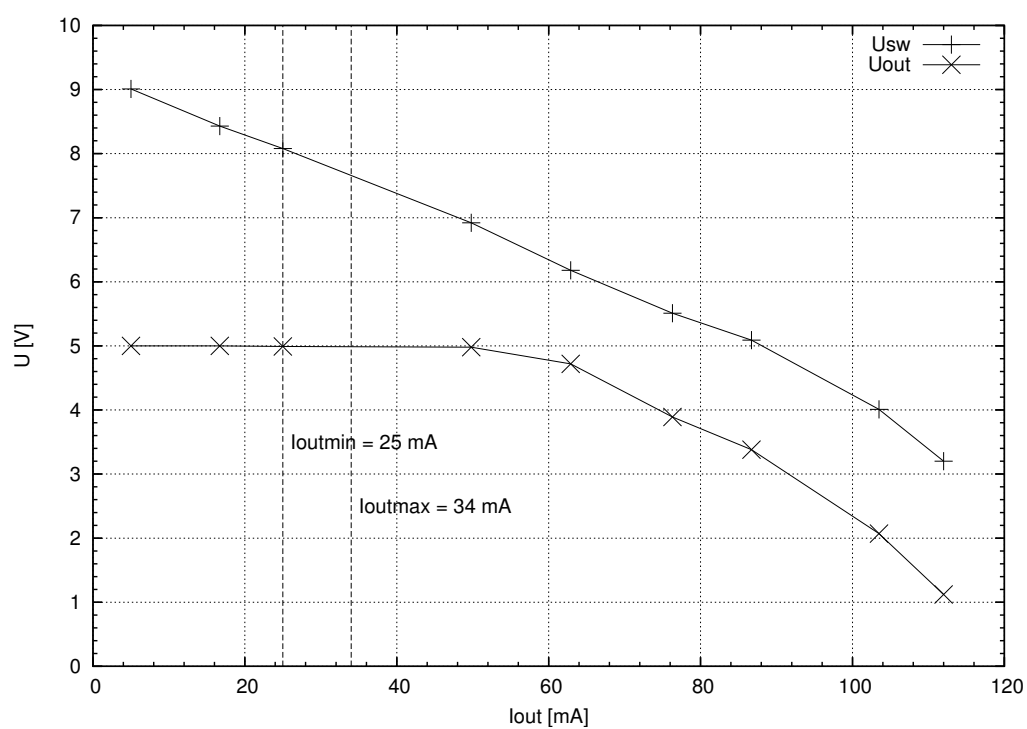
Zaradi enostavnega vezja in zanesljivega delovanja je bila izbrana izvedba vezja s preklopnim kondenzatorjem (angl. switched capacitor) in linearnim stabilizatorjem napetosti. Zaradi nizkega izhodnega toka slabosti te izvedbe (velike izgube v primerjavi z izvedbo s tuljavo) ne pridejo do izraza.

Preklopni del (integrirano vezje U11, kondenzator C4 ter diodi D2 in D3) deluje s frekvenco približno 10kHz in proizvaja nestabilizirano napetost približno -9V . Ta napetost se nato stabilizira s stabilizatorjem v integriranem vezju U22.

Rezultati meritev izhodne napetosti v odvisnosti od izhodnega toka so prikazani na sliki 31.



A NAPETOSTNI INVERTER



Slika 31: Rezultat meritev napetosti pred (U_{sw}) in po (U_{out}) stabilizatorju napetosti v odvisnosti od izhodnega toka I_{out}

B Izračun impulznega ojačevalnika

Predpostavimo, da je vhodni impulz pravokotne oblike in da ima priključeni magnetofon majhno izhodno upornost.

Najprej izračunamo največji kolektorski tok, za katerega hkrati privzamemo, da je minimalni tok skozi R1, pri katerem je na izhodu ojačevalnika logična 0.

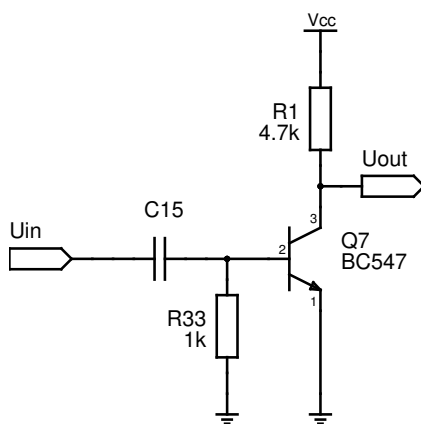
$$I_{Cmin} = \frac{U_{CC} - U_{CEsat}}{R_1} \quad (1)$$

Minimalni bazni tok, ki omogoča I_{Cmin} :

$$I_{Bmin} = \frac{I_{Cmin}}{\beta} \quad (2)$$

V kolikor držijo zgornje predpostavke, se kondenzator C15 po prvi fronti impulza najprej polni predvsem skozi bazo tranzistorja Q7, ko pa napetost na bazi pade do U_{BE} tranzistorja Q7, se polni predvsem čez upor R33.

Impulz na izhodu ojačevalnika traja toliko časa, dokler teče v bazo tok, ki je večji od I_{Bmin} .



B IZRAČUN IMPULZNEGA OJAČEVALNIKA

Zaradi eksponentne $I_B(U_{BE})$ karakteristike tranzistorja se kondenzator čez bazo napolni v zelo kratkem času, širina impulza na izhodu pa je predvsem odvisna od polnjenja čez upor R_{33} in tako časovne konstante RC člena $R_{33} C_{15}$.

Predpostavimo, da tok v bazo postane zanemarljiv takrat, ko postane manjši od približno ene desetine toka čez upor. Napetost na bazi je takrat:

$$\frac{1}{10} \frac{U_{BE}}{R_{33}} = \frac{I_S}{\beta} \left(e^{\frac{U_{BEstart}}{U_T}} - 1 \right) \quad (3)$$

$$U_{BEstart} = U_T \ln \left(\frac{U_{BE} \beta}{I_S R_{33} 10} + 1 \right) \quad (4)$$

Podobno je napetost na bazi, ko teče minimalni bazni tok, enaka:

$$U_{BEstop} = U_T \ln \left(\frac{I_{Bmin} \beta}{I_S} + 1 \right) \quad (5)$$

Trajanje impulza je tako:

$$t = R_{33} C_{15} \ln \frac{U_{BEstart}}{U_{BEstop}} \quad (6)$$

Potrebna kapacitivnost C_{15} pri znanem uporu in trajanju impulza pa:

$$C_{15} = \frac{t}{R_{33} \ln \frac{U_{BEstart}}{U_{BEstop}}} \quad (7)$$

Pri uporu $R_{33} = 1\text{k}\Omega$ in trajanju impulza $t = 100\mu\text{s}$ ⁷ tako dobimo vrednost $C_{15} = 0.72\mu\text{F} \approx 1\mu\text{F}$.

⁷Minimalno trajanje impulza $45\mu\text{s}$ sledi iz zanke `LOAD_READ_PULSE` na naslovih od `0x0ef9` do `0x0f05`. Zaradi večje zanesljivosti je bila uporabljena približno dvakrat večja vrednost.

C Števila s plavajočo vejico

Galaksijin operacijski sistem za izračun matematičnih izrazov interno uporablja kalkulator s skladom in RPN (angl. reverse Polish notation) notacijo. Tako se na primer vsi BASIC izrazi pred izračunom interno pretvorijo v tako obliko, kompleksnost izrazov pa je omejena le s pomnilnikom, ki je na voljo za aritmetični sklad.

Sistem uporablja za zapis števil s plavajočo vejico dve različni obliki. Krajša 4-bytna se uporablja za dolgoročno shranjevanje vrednosti (na primer za zapis vrednosti BASIC spremenljivk in polj). Daljša 5-bytna, ki je zaradi razporeditve podatkov primernejša za izvajanje aritmetičnih operacij, pa se uporablja za hranjenje vmesnih rezultatov izračunov (na primer za vrednosti na aritmetičnem skladu). Na ta način je bil dosežen kompromis med hitrostjo izvajanja aritmetičnih operacij in potrebami po pomnilniku.

Oba zapisa vedno vsebujeta normalizirane vrednosti in uporabljata 24 bitov za zapis mantise, 8 bitov za zapis eksponenta in en bit za zapis predznaka. Vrednost števila je določena po naslednji enačbi:

$$N = \begin{cases} +1 \cdot M \cdot 2^{E-24} & \text{pri } S = 0 \\ 0 & \text{pri } E = -128 \\ -1 \cdot M \cdot 2^{E-24} & \text{pri } S \neq 0 \end{cases} \quad (8)$$

Kjer je M nepredznačena vrednost mantise, E predznačena vrednost eksponenta, S vrednost predznačnega bita in N zapisano število. Na ta način je mogoče zapisati števila v naslednjem obsegu:

$$N \in [-1 \cdot (2^{24} - 1) \cdot 2^{127}, +1 \cdot (2^{24} - 1) \cdot 2^{127}] \quad (9)$$

Zapis ne omogoča zapisa neštevilskih vrednosti (neskončnost, nedefinirana vrednost in podobno).

V tabelah 14 in 15 sta prikazani interpretaciji posameznih bitov v obeh oblikah zapisa števil. Najvišji bit mantise (M_{23}) v 4-bytnem zapisu manjka, ker je zaradi obvezne normalizacije števil vedno enak 1.

C ŠTEVILA S PLAVAJOČO VEJICO

bit			7	6	5	4	3	2	1	0	
IX - 5	L'	E'	M_7	M_6	M_5	M_4	M_3	M_2	M_1	M_0	mantisa
IX - 4	H'	D'	M_{15}	M_{14}	M_{13}	M_{12}	M_{11}	M_{10}	M_9	M_8	
IX - 3	C'	B'	M_{23}	M_{22}	M_{21}	M_{20}	M_{19}	M_{18}	M_{17}	M_{16}	
IX - 2	L	E	E_7	E_6	E_5	E_4	E_3	E_2	E_1	E_0	eksponent
IX - 1	H	D	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0	predznak

Tabela 14: 5-bytni zapis s plavajočo vejico

bit	7	6	5	4	3	2	1	0
HL + 0	M_7	M_6	M_5	M_4	M_3	M_2	M_1	M_0
HL + 1	M_{15}	M_{14}	M_{13}	M_{12}	M_{11}	M_{10}	M_9	M_8
HL + 2	E_0	M_{22}	M_{21}	M_{20}	M_{19}	M_{18}	M_{17}	M_{16}
HL + 3	S_7	E_7	E_6	E_5	E_4	E_3	E_2	E_1

Tabela 15: 4-bytni zapis s plavajočo vejico

Izjava

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja prof. dr. Tadeja Tume, univ. dipl. in. el. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.