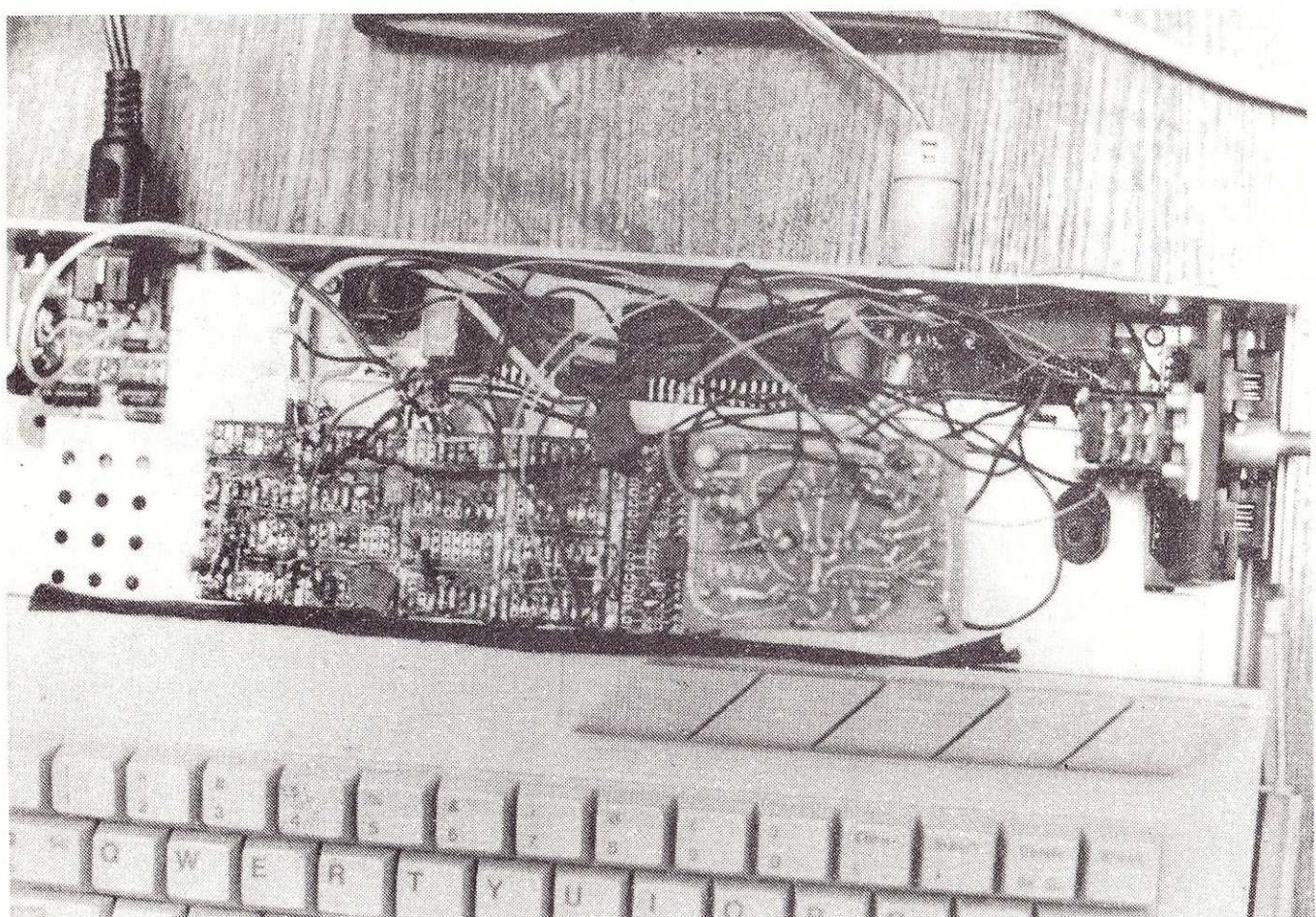
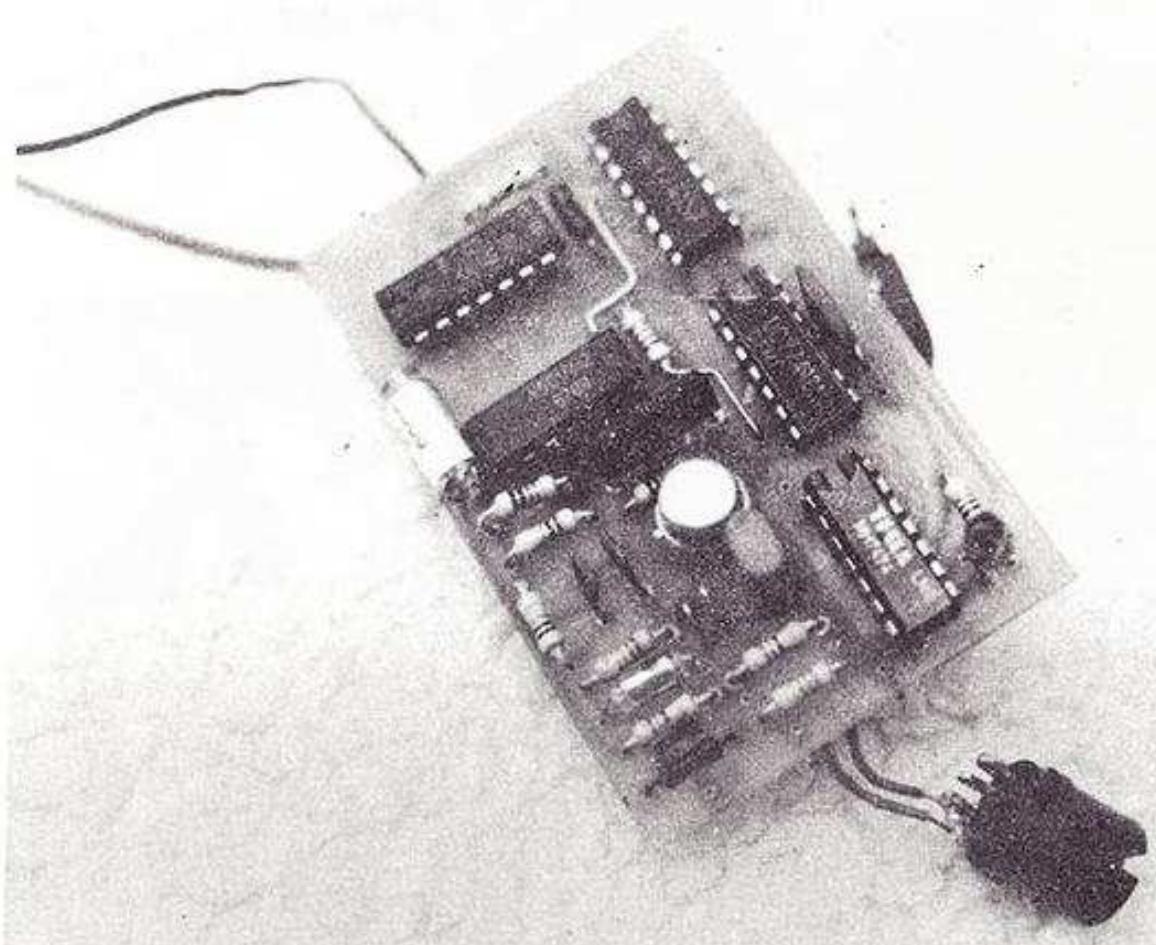
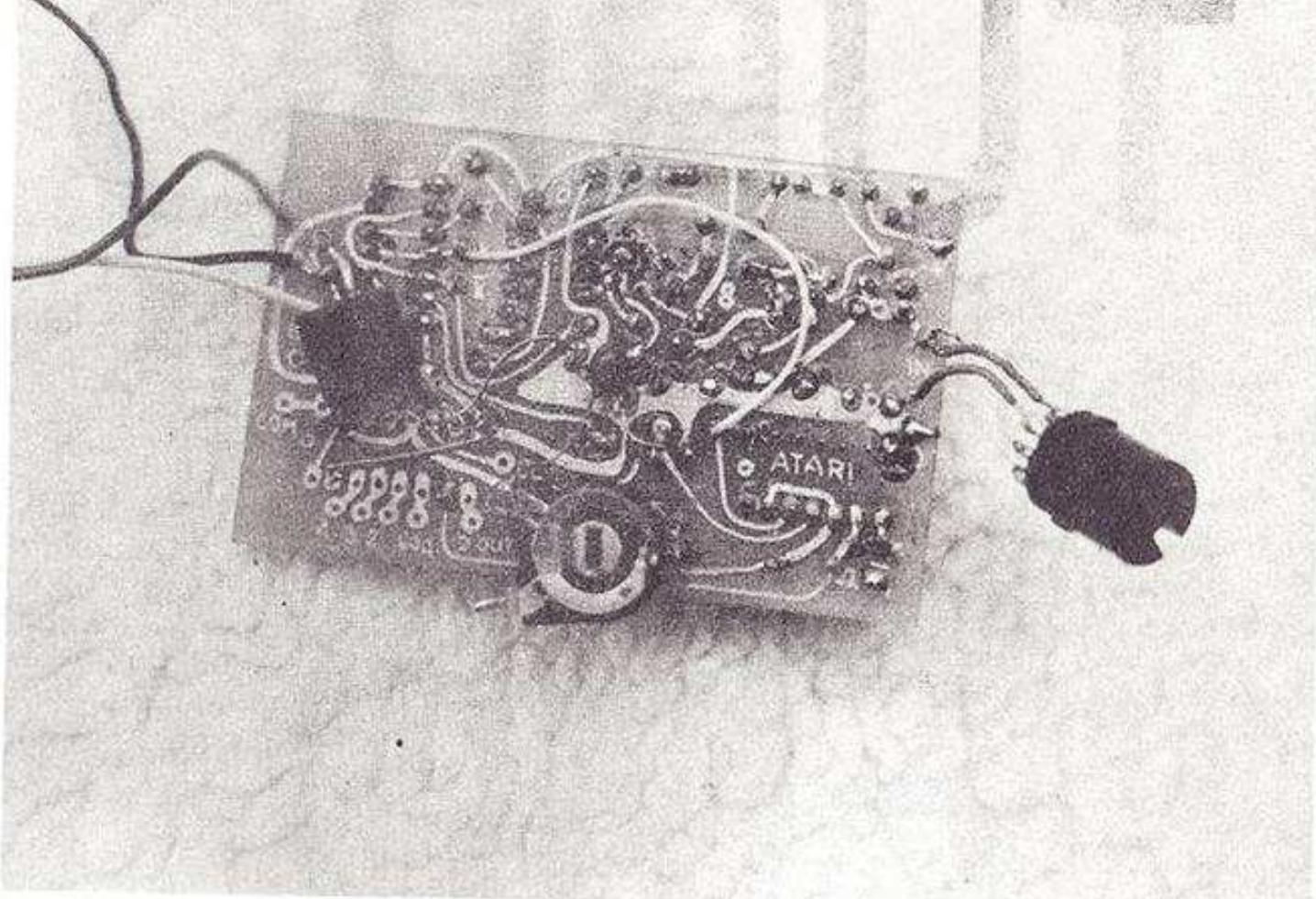


TURBO

EMO





POPIS ROS

Rýchly operačný systém pre prácu s kazetou vznikol v ATARI klube v Leviciach. ROS pozostáva z dvoch nedeliteľných samostatných častí - softwarovej a hardwarovej.

Popis softwarovej casti ROS:

ROS využíva oblasť pamäte RWM, ktorá je fyzicky umiestnená pod ROM-kou operačného systému. Je to oblasť \$C000 - \$FFFF. Po nahrati zavádzacího programu inicializácia t.j. prepis obsahu ROMky OS být po byte do oblasti RAM "pod OS" s následnou modifikáciou. ROM OS je ovládaný nultým bitom obvodu PIA. Normálne tam je log.1 a vtedy je zapnutý ROM OS. ROS vyžaduje, aby tento bit bol "zhodený". Prvým bitom obvodu PIA je ovládaná ROM BASIC. Preto v basicu musí byť na bráne A obvodu PIA hodnota \$FC, ak chceme zaradený ROS a \$FD ak chceme pôvodný OS. Z basicu to dosiahneme inštrukciami POKE 54017,252 resp. POKE 54017,253. Každé stlačenie systémového tlačítka RESET spôsobí preprogramovanie bitu 0 brány A PIA na log.1 a tým vyradenie ROSu z činnosti. Na prerobenie programov do rýchlej formy v basicu nie je teda potrebný ziazen transformačný program stačí program načítať so zapnutou ROMkou OS klasickým spôsobom a vyslať s vypnutou ROMkou OS.

ROS nemôže spolupracovať s programami, ktoré využívajú oblasť pamäte kde je umiestnený on sám, t.j. oblasť \$C000 až \$FFFF.

Zavedenie ROS do počítača.

Zavádzací ROSu nahráme známym spôsobom START + OPTION. Je vytvorený univerzálny zavádzac pre všetky typy programov, ktoré prichádzajú do úvahy. Zavádzací ROSu po nahrati a odštartovaní nie je už potrebný, čiže môže dôjsť aj k jeho znicieniu. Po odštartovaní dôjde k jeho prepísaniu pod OS, kde zostane až dovtedy kým počítač nevypneme zo siete. ROS vďaka tomuto umožňuje BOOTovanie strojových programov aj z BASICu pomocou príkazu USR. Takýmto spôsobom je možné "bezrestne" prechádzat z BASICu napr. do ATMASu a naopak bez toho, aby bolo nutné znova nahrať zavádzac.

Po nahrati zavádzací počítač ponúkne možnosti:

B - skok do BASICu, kedy prebehne úplná inicializácia ako po zap. počítača (dôjde k vymazaniu zavádzacího). Po inicializácii zostane zaradený pôvodný OS. Zapnutie ROS dosiahneme inštrukciou POKE 54017,252.

S - bootovanie s prítomným ROM BASIC, používa sa pri programoch, ktoré v originálnej forme treba načítať pomocou START.

L - bootovanie strojových programov, ktoré v originálnej forme treba načítať pomocou START+OPTION.

Obsluhu najlepšie vysvetlí na príklade nahávania súboru v BASICU. Napríklad chceme nahrať na kazetu súbor, ktorý sme do počítača dostali ľubovoľným spôsobom (kazeta, disk, klávesnica). Použijeme niektorý zo štandardných príkazov CSAVE, SAVE "C:", PUT, LIST "C:" (Predpokladám vypnutý ROM BASIC). Po odoslaní príkazu dôjde k vymazaniu obrazovky a v ľavom hornom rohu sa zobrazí "MENO:". Počítač očakáva názov súboru so stručným popisom súboru. Na popisanie súboru je k dispozícii 122 bytov a prostriedkom na popisanie je editor OS. Ak sa pomýlime v písani hlavičky fungujú riadiace znaky editora, ale pozor do buffera sa ukladajú spolu s významovými bytami, čím zbytočne zabera ju miesto. Klávesa DELETE funguje na obrazovke sice správne (pri SAVE), ale pri výpise informačného bloku (pri LOAD) je výpis bloku na obrazovku nesprávny (poposúvaný). Preto doporučujem pri omyle vždy stlačiť CONTROL+DELETE, čím dôjde k vymazaniu obrazovky a buffera a popis možeme urobiť od začiatku. Klávesa RETURN služí ako znak EOL - koniec riadku a návrat kurzora na ďalší riadok. Na ukončenie tvorby hlavičky súboru slúži klávesa ESC, kedy po jej stlačení počítač trikrát "zavŕší", čo je znamenie pre stlačenie PLAY+RECORD na magnetofón. Ďalším stlačením ľubovoľnej klávesy dôjde k vyslaniu najprv nastavovacieho a informačného bloku (hlavička+popis) a potom samotného súboru.

Ak by sme z nejakého dôvodu nechceli vysielaný súbor popísat, tak hned po zobrazení "MENO:" treba stlačiť klávesu ESC, vtedy počítač "zavŕší" len dvakrát a nedôjde k vyslaniu nastavovacieho a informačného bloku, ale len samotného súboru.

Čítanie súboru v BASICu.

Na príjem tiež používame štandardné príkazy CLOAD, LOAD"C:", RUN "C:", GET, ENTER"C:". Po odoslaní príkazu sa v ľavom hornom rohu objaví "MENO:". Počítač od nas žiada názov súboru, ktorý chceme načítať. Na tento účel ROS pracuje s tzv. hviezdickovou konverzáciou. Napr. po napísani názvu súboru "AT*" by počítač nahral súbor s názvom "ATOM,ATMAS,ATARI...". Podobne po napísaní iba hviezdickej počítač nahrá prvý súbor, ktorý nájde na kazete. K samotnému čítaniu dat dôjde po stlačení klávesy RETURN. Od tohto momentu počítač číta bloky. Ak narazí na informačný blok, zobrazí ho a skontroluje prvých 6 bytov, či sú totožné s našou voľbou súboru, ktorý chceme do počítača nahrať. Ak nesúhlasí volba, čítanie sa nepreruší a v ľavom hornom rohu obrazovky sa budú postupne zobrazovať hexadecimálne hodnoty poradových čísel blokov súboru. Počítač precíta každý blok, zistí či nedošlo ku chybe v kontrolejnej sume a načítaný blok zostane len v bufferi. V tomto režime zostane dovtedy kým nenájde informačný blok "s naším súborom" alebo nestlačíme

BREAK. Vtedy v ľavom hornom rohu zostane číslo 02, čo je poradové číslo informačného bloku (01=nastavovací blok) a v tom dobrém prípade zostane až do konca prenosu dat. Ak dojde ku chybe čítania v niektorom bloku, magnetofon zastane, zvuk sa vypne a v pravom hornom rohu sa zobrazí číslo bloku, v ktorom došlo ku chybe prenosu dat. Vtedy je potrebné vrátiť kúsok kazetu a stlačiť niektorú zo systémových kláves (okrem RESET) a počítač bude hľadať posledný dobre prečítaný blok a zobrazuje postupne čísla blokov. Ak ho nájde, opäť sa zapne zvuk a čísla blokov v ľavom rohu sa prestanú zobrazovať. Túto operáciu možme opakovať neobmedzene, ak však je na páske "tvrdá chyba", prečítať blok sa nám nepodari. Návrat do BASICu je možný kedykoľvek cez klávesu BREAK. Tu je vhodné upozorniť na možnosť, ktorú ROS umožňuje. Ak sa nám niektorý blok súboru na kazete poškodí, máme možnosť ho načítať trebárs z úplne inej kazety, kde máme ten istý súbor. Pri dôležitých programoch doporučujem archivovať programy duplicitne.

Treba ešte upozorniť na možnosť využitia čítania dat od určitého bloku. Do okamžitého čítania sa dá prejsť po zobrazení "MENO:" okamžitým stlačením klávesy RETURN - využitie u súborov, ktoré majú z nejakého dôvodu poskodený informačný alebo nastavovací blok prípadne datový blok - ak ide o texty.

Operačný systém vysiela pred samotným súborom 20 sekundovú log.1=klud na zbernicu. Pauza medzi blokmi je nainicializovaná pri vysielaní pre mňa z neznámych dôvodov až na 15x20 ms, hoci počítaču staci na uloženie bloku dat 20 ms. ROS tieto časy skracuje na 1 sek. resp. 2x20 ms. Tým je dosiahnuté ďalšie zefektívnenie využívania pásky.

ROS pracuje pod systémom to znamená, že spolupracuje s programami, ktoré využívajú štandardné rutiny OS. Základná verzia nemodifikuje tabuľku HATABS, spolupracuje so všetkymi perifériami nezmeneným spôsobom. Vsetky užívateske programy budú pracovať pod ním dovtedy pokiaľ z nejakého dôvodu nedojde k zapnutiu pôvodného OS. Môže k tomu dojsť preprogramovaním portu A bitu 0 na log.1, alebo stlačením RESET. Toto môže byť niekedy potrebné (načítať klasickú nahrávku), inokedy nežiaduce. Východiská sú 2. Bud softwarová úprava programu, alebo inštalovanie prepínača bitu 0, ktorý riadi cez MMU prístup k jednotlivým oblastiam pamäte, priamo do počítača. Táto úprava nie je nijak náročná a rieši všetky problémy u tohto systému. V jednej polohe prepínača funguje pôvodný OS a v druhej ROS. V súčasnosti je k dispozícii ROS, ktorý štandardne pracuje s tlačiarou s rozhraním typu CENTRONIC. To zas prináša výhodu, že netreba nijakým spôsobom upravovať profesionálne programy a doplniť ich o rutinu pre tlač a ihned po nahratí ROSu sa dajú využívať tlačiarne typu CONSUL, D100 a pod., ktoré sú relativne dostupnejšie ako ATARI 1029.

Podobne bol vyvinutý RAMDISK pre ATARI 130XE a zpracovaný do OS na miesto "španielskych znakov" ako zariadenie typu "Q:". Tento ROS dostal nazov "QRROS". Práca

pod týmto systémom je pohodlná a značne zrýchluje a zefektívňuje prácu pri vývoji programov najmä v strojovom kóde, kedy každé odštartovanie neodladeného programu môže viest k jeho zničeniu resp. k poškodeniu "nadradeneho" programu, pod ktorým program pišeme. Vždy pred týmto odštartovaním je vhodné si poslednú verziu programu uložiť do RAMDISKu, od kiaľ ho v prípade potreby nepoškodený vytiahneme. Do RAMDISKu je možné uložiť súbory o sumárnej dĺžke \$FE00 bytov.

OBSLUHA RAMDISKU

Archivácia dat prebieha opäť pomocou štandardných príkazov, určených na vstupno/výstupné manipulácie. Čiže v hocijakom neupravenom profesionalnom programe, ktorý umožnuje vyslať data na dataset resp. disk stlačíme pri opýtani sa na druh zariadenia "Q:". Príklady:

BASIC-> SAVE"Q:", LIST"Q:", PUT#CISLO KANALA, hodnota (musí byť otvorený kanál)

ATMAS->WQ z editoru, CTRL+S z monitoru

SPEEDSCRIPT->CTRL+S,CTRL+P

Vždy pred prvým použitím RAMDISKu je nutné ho zinicializovať. Inicializácia prebehne vždy po odoslani zavináča ako názvu súboru. Dôjde k nastaveniu ukazovateľov, ktoré RAMDISK vnútorne využíva.

Formát uloženia dat záleží na použitom príkaze, ktorým data do RAMDISKu zapíšeme, čiže pri spätnom čítaní treba zvoliť "zrkadlový" príkaz na načítanie dat (LIST"Q:"->ENTER"Q:"). Na popísanie súboru pri zápisе je k dispozícii 8 bytov. Pri čítaní sú kontrolované prvé štyri znaky názvu súboru, ktorý chceme z RAMDISKu natiahnuť. Preto je vhodné si jednotlivé verzie značiť číslom, ktorým je jednoznačne určený súbor v RAMDISKU. Pri naťahovaní súboru z RAMDISKU funguje tiež hviezdicková konverzácia obdobne ako pri datasete, preto ju nebudem dalej popisovať.

Mazanie posledne uloženého súboru dat v RAMDISKu:

Pri zadávaní mena súboja treba stlačiť BREAK+START a dôjde k zníženiu ukazovateľa počtu uložených súborov v RAMDISKu (pri omyle sa dá ešte zachrániť).

Pri "oslovení" RAMDISKU sú vždy zobrazia názvy súborov v ňom uložené súčasne s ich dĺžkou, ktoré fyzicky zaberajú v pamäti. Posledným údajom, ktorý sa zobrazí je sumárna dĺžka všetkých súborov v hexadecimálnej forme. Takto máme neustály prehľad o volnej kapacite pamäte RAMDISKu.

BLOKOVÝ PRENOS DAT

Operačný systém pri vstupno-výstupných operáciách s kazetou resp. disketou pracuje s datami vždy po blokoch. Znamená to, že súbor o ľubovoľnej dĺžke sa "rozseká" na jednotlivé bloky o významovej dĺžke 128bytov. Je to dané jeho univerzálnou CIO rutinou, ktorá v princípe pracuje so všetkými zariadeniami, ktoré sú zadefinované v tabuľke HATABS, rovnakým spôsobom. Vo všeobecnosti sa dá povedať, že blokový prenos údajov je z hľadiska vylúčenia chybného prenosu dat spoločne súčasťou ako prenos monoblokom. Takyto prenos koresponduje so spôsobom záznamu dat na disketu, kde jednotlivé bloky predstavujú sektory na diskete. Ziaľ, originálny spôsob záznamu dat na kazetu nevyužíva žiadne z možných výhod blokového prenosu dat. Pri návrhu ROS bolo snahou zmeniť v OS minimum a odstrániť najväčšie nevýhody originálneho spôsobu záznamu dat na kazetu.

- "anonymita" jednotlivých blokov
- nemožnosť opakovaného čítania bloku
- označovanie súborov pri archivácii
- selektívny výber súborov pri čítaní
- nízku prenosovú rýchlosť a jeho malú spoločnosť

Blokový prenos dat prebieha vždy cez tzv. buffer, čo je vyrovnavacia pamäť, do ktorej resp. z ktorej sa data dočasne prenesú na skutočnú (fyzickú) adresu resp. zaznamenajú na kazetu.

Kazetový buffer začína na adrese \$3FD a končí na \$47F čiže má dĺžku 131 bytov.

Prvé dva byty majú význam synchronizačných značiek, podľa ktorých sa "dotiahujú" vnútorné vzorkovacie hodiny na danú prenosovú rýchlosť v rozsahu cca \$480-\$960 Baudov. Synchronizačné byty majú hodnotu \$55hexa, čo sú pravidelne prestriedané log. a log.0.

Tretí byt na adrese \$3FF je tzv. riadiaci a podľa jeho hodnoty OS rozpoznáva bloky:

- \$FC-datový blok úplný
- \$FA-datový blok neúplný
- \$FE-koncový blok

Blok nahratý na kazete má vždy 132 bytov. Posledný byt nesie informáciu o kontrolnej sume bloku. Aj neúplný blok obsahuje $3+128+1$ bytov, neúplným sa nazýva preto, že platných bytov nie je 128 ale len 127, akú hodnotu obsahuje posledný významový byt bloku (\$47F). Koncový blok obsahuje $3+128+1$ bytov, kde všetky významové byty majú hodnotu 0. K vyslaniu koncového bloku dojde vždy na záver ľubovoľného vysielania. Pri systémovom programovaní nie je možné vyrobiť súbor bez koncového bloku.

ROS má dodefinované bloky s riadiacim bytom

\$FF - nastavovací blok
\$FD - informačný blok

ROS zmenil štruktúru bloku nasledovným spôsobom.

Pri mode s externými hodinami synchronizačné byty stratili význam a preto prvé dva byty nesú informáciu o poradovom čísle bloku v súbore. Všetko ostatné ostane v pôvodnej forme.

. Význam nastavovacieho bloku:

- navrhovaný hardware vyžaduje vždy pred čítaním bloku definovaný stav "klúč na zbernicu=log.1", ktorý zabezpečí správe tento blok.

. Význam informačného bloku:

- slúži na popisanie súboru (hlavička) pri zázname a selektívny výber súborov pri čítaní.

Význam ostatných blokov ostal nezmenený.

POPIIS HARDWARE ROS

Na úvod trocha k historii vzniku systému.

Po predvedení programového čítania dat Košickým klubom zhruba pred rokom nám nedalo spávať neskúsiť íst vlastnou cestou vo vývoji rýchleho záznamu dat. V záujme kompatibility s existujúcimi programami sme hneď na začiatku odmietli cestu programového čítania vstupu a pokúsili sa o hardwarové riešenie problému. To sa podarilo a ďalej zlepšovanie systému bolo čiste softwarovou záležitosťou. Mylnou sa ukázala cesta stereo záznamu.

Výhodou tohto zapojenia je v jeho univerzálnosti, pracuje rovnako dobre s Pražským aj Novozámockým TUREOM samozrejme len s použitím prepinača, ktorým treba jednotlivé vstupy do počítača prepínat.

Dalšiu perspektívou systému vidím v možnosti napálenia ROS do ROMKY, kedy by nebolo nutné nahrávať zavádzac a aj pamäť RAM pod OS by sa dala využiť na iné účely.

PRINCIP ZÁZNAMU DAT NA PÁSKU

Pre úspešnú realizáciu ďalej popisovaného interfacu je nutné pochopiť princíp záznamu.

Obvod POKEY má v počítači ATARI okrem iných funkcií na starosti aj prenos dat. K tomuto účelu mu slúžia 4 vývody:

CLKIN - vstup externých hodín
 CLKOUT - výstup vnútorných hodín
 DATAIN - vstup datového signálu
 DATAOUT - výstup dat

Všetky signály majú úroveň TTL.

VÝZNAM SIGNÁLOV:

CLKIN - používa sa v režimoch, kedy je prenosová rýchlosť diktovaná zvonku práve týmito "hodinami" (disketa, tlaciaren). Obvod POKEY vždy na padajúcu hranu vzorkne datový signál a postupne si skladá bit po bite az vznikne celý byt. Tento vstup teda úzko súvisí so vstupom DATAIN.

CLKOUT - výstup, ktorý definuje vysielaciu rýchlosť prenosu dat. Vždy na nábežnu hranu týchto hodín výsle POKEY ďalší bit a postupne celý byt.

DATAIN - datový signál vzorkovaný buď vnútornými hodinami alebo signálom CLKIN.

DATAOUT - tento signál priamo reprezentuje vysielané data, či sú už zakódované alebo nie. POKEY totiž v jednom zo svojich módov umožnuje frekvenčne modulovať tento výstup datami.

Tento mód využíva OS pre prácu s datasetom.

Postup naprogramovania jednotlivých módov do už zverejnený takže je zbytočné to robiť znova.

Kazeta ako neinteligentná periféria využíva len signály DATAIN a DATADOUT. OS používa pri práci s kazetou programovo nastavovanie prijmovej rýchlosťi (frekvencie vnútorného vzorkovania) na základe doby trvania 20 zmien signálu z log.1 na log.0. OS má tabuľkovo zadane referenčné časy, ktoré potom skokovite mení podľa doby trvania spomínaných 20 zmien signálu na vstupe DATAIN. Toto nastavovanie prijmovej rýchlosťi robí vždy na začiatku každého bloku. Týmto spôsobom sa vie dotiahať na prenosové rýchlosťi v rozsahu 480 - 960 Baudov. Táto časť OS sa zdá byť práve naj slabším miestom celého OS ATARI. Naše praktické skúsenosti ukázali, že "pevne" daná prenosová rýchlosť nie je príčinou nespolahlivosti, protože kolísanie súčasných magnetofónov je pri týchto prenosových rýchlosťach skutočne zanedbateľné ale práve pri vypočítavaní prijmovej rýchlosťi sa OS "naladí" dosť často nesprávne. ATARI moduluje data frekvenčne a to log.1 5280 Hz a log.0 3950 Hz. Takýto spôsob modulácie ostro ohraňuje dosažiteľnú prenosovú rýchlosť asi na 1800 Baudov. Preto bolo nutné, ak sme chceli ľatšie rýchlejsie, upustiť od originálneho záznamu.

Pri prenose dat sa obyčajne používa štartstopový prenos. Znamená to pridanie ďalšieho nadbytočného bitu pred byt a ďalšieho za byt. Tieto sa volajú štartbit a stopbit. Start a stop bity si POKEY vyrába automaticky sám a pri príjme ich zas "odfiltruje".

Pokial' je klúč na zbernicu t.j. na vstupe DATAIN je log.1, bude POKEY stále čakať na padajúcu hranu - štartbit a od tohto momentu začne bráť do úvahy či už vnútorné alebo externe hodiny od ktorých zas bude vzorkovať samotný datový signál. Pokial' teda nepríde štartbit môže sa CLKIN meniť akol'vek a POKEY nebude robiť "nič". Po príchode štartbitu POKEY bude čakať na ďalšiu padajúcu hranu hodín, kedy vzorkne data a túto vzorku si zapamäta ako 1.významový bit bytu. Toto sa zopakuje 8. krát až si POKEY takto "poskladá" celý byt.

Počas trvania stopbytu POKEY generuje IRQ, žiadosť o obsluhu prerušenia od vstupných dat a on sám čaká na štartbit ďalšieho bytu. OS má na spracovanie bytu relatívne dosť času, lebo ďalšie prerušenie príde zas až keď POKEY spracuje ďalší byt. Z tohto hľadiska obmedzenie prenosovej rýchlosťi nehrozí (pri prenose priamo z počítača do počítača bez ďalšieho prídavného hardwaru prepojením DATADOUT - DATAIN a CLKOUT - CLKIN bola dosiahnutá prenosová rýchlosť 32000 Baudov).

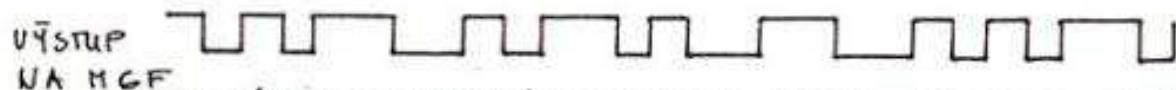
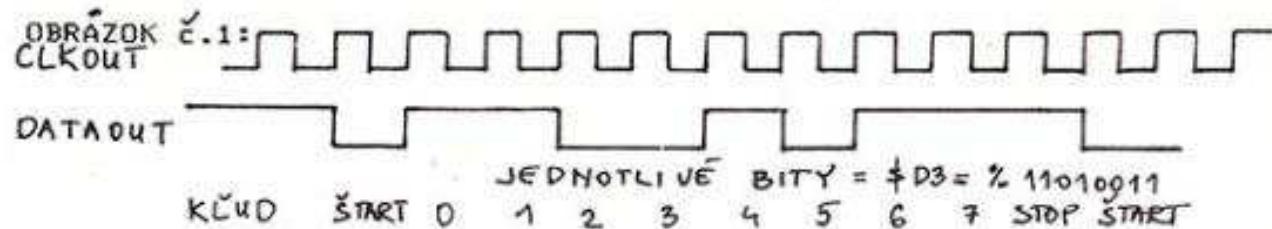
Po prvých pokusoch prenosu dat medzi dvoma počítačmi sme sa presvedčili o správnosti našej cesty a potom bolo treba vyriešiť formu záznamu, v ktorom by bola zakódovaná informácia aj o prenosovej rýchlosťi aj o dátach. Nie je totiž možné s ohľadom na prenos jednosmernej zložky zaznamenať data v pôvodnej forme priamo na pašku.

FÁZOVÝ SPÔSOB ZÁZNAMU

Záznam signálu, ktorého polarita sa mení v závislosti na datach sa nazýva fázovou moduláciou. Prakticky sa dá realizovať najjednoduchšie pomocou obvodu EOR. Tento obvod realizuje funkciu danú pravdivostnou tabuľkou:

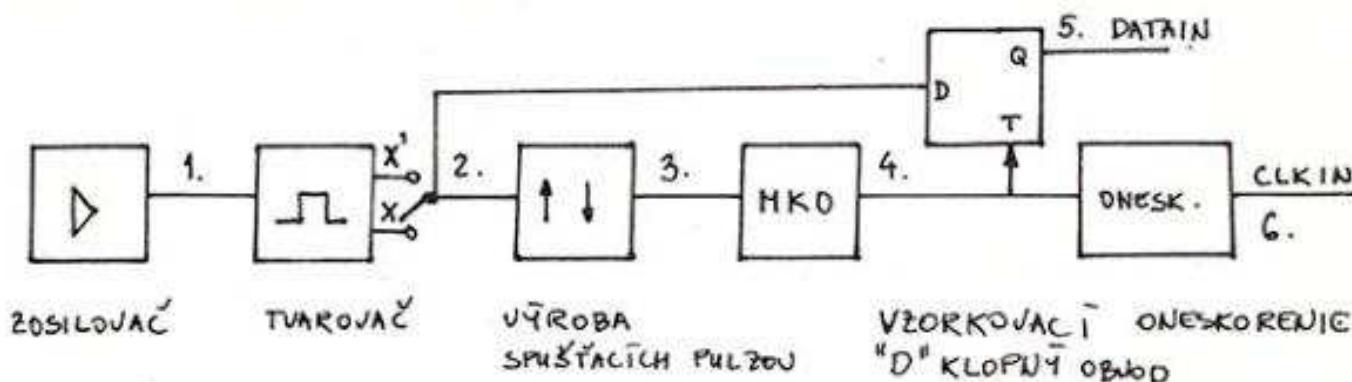
VSTUP 1	VSTUP 2	VÝSTUP
0	0	0
0	1	1
1	0	1
1	1	0

Ak na vstup 1 priviedieme DATAOUT a na vstup 2 CLKOUT, na výstupe sa bude meniť signál tak ako potrebujeme. Síce každá zmena dat sa prejaví ako zmena fáze výstupného signálu a ak sú data monotónne, výstup bude sledovať hodiny bud' v takej alebo opačnej fáze ako hodiny. Podstatné je to, že takýto signál obsahuje len minimalnu jednosmernú zložku, ktorú je možné zanedbať. Ako uvidíme ďalej obsahuje v sebe "zakódovanú" informáciu aj o prenosovej rýchlosťi aj dátach. Najlepšie to vysvetlí na konkrétnom príklade.
Je treba zaznamenať byt o hodnote napr.: \$D3, čiže binárne 11010011.



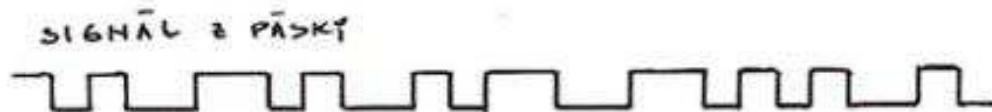
Z obrázku je vidieť ako vznikne signál, ktorý sa nahrá na pásku z datového a hodinového signálu.
Späťné dekódovanie takéhoto signálu do pôvodnej formy je zložitejší proces, ktorý si vysvetlíme z blokovej schémy.

BLOKOVA SCHÉMA DEKODERA:



Po dostatočnom zosilnení a odfiltrovani nežiadúcich zložiek sa tento prevedie do úrovne TTL. V ideálnom prípade dostaneme v bode 2 blokovej schémy taký istý priebeh signálu aký sme pôvodne zaznamenali na páske, avšak z dôvodov ktoré uvediem neskôr, negovaný.

OBR.č.2 - BOD 2 blokovej schémy.



Poznámka: Signál z tohto bodu sa dá využiť aj pri TURBOsystémoch z Prahy a Nových Zámkov, stačí ho len priviesť na odpovedajúci vstup počítača.

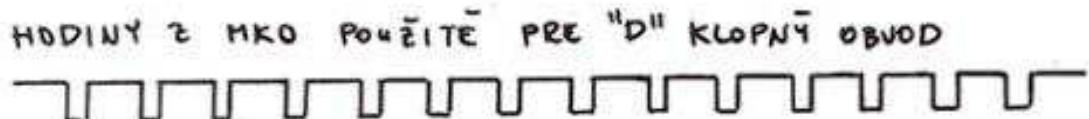
V ROS systéme je ho nutné ďalej spracovávať. Potom sa zo signálu vyrábajú spúšťacie impulzy pre monostabilný klopný obvod (MKO). Tieto impulzy sa vyrobia z každej nábežnej a padajúcej hrany signálu.

OBR.č.3 - BOD 3 blokovej schémy.



Sírka týchto pulzov musí byť čo najmenšia, ale zas dostatočná pre spoloahlive naštartovanie MKO. Doba kyvu MKO musí byť nastavená na $3/4$ períody prenosovej rýchlosťi.

OBR.č.4 - BOD 4 blokovej schémy.



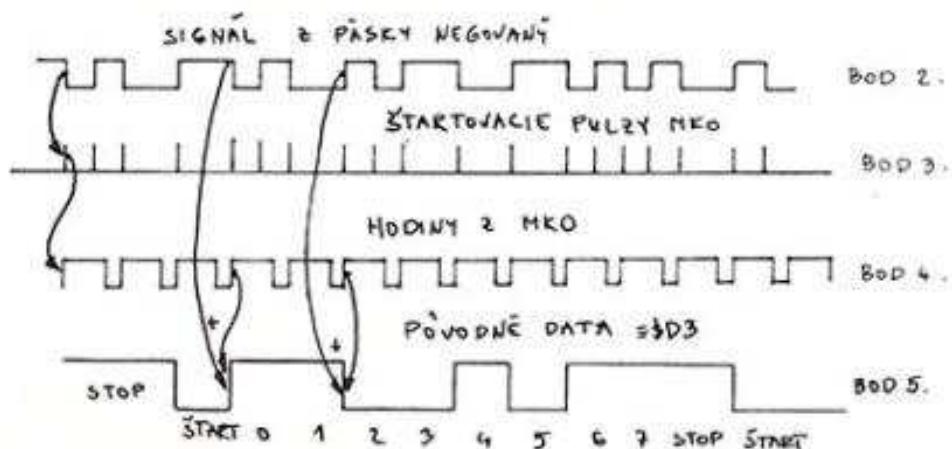
Takýmto spôsobom získame pôvodné hodiny, ktoré sú už použitelné ako signál CLKIN. Tvar týchto hodín nie je presne

symetrický, ako bol originálny priebeh hodín. Pri čítaní dat

nezáleží na tvare hodín, možu byť dokonca v tvare pulzov, podstatnou pre POKEY pri prijme je padajúca hraná hodín, kedy si POKEY vzorkne vstup dat (vysiela na nábežnú hranu). Koli lepšiemu odfiltrovaniu možných porúch v signále je zaradený ešte D klopný obvod, cez ktorý sa vzorkuju vstupné data. D klopný obvod vyžaduje určitý predstih platných dat pred hodinami na svojom vstupe, aby došlo k správnemu prenosu z jeho vstupu "D" na výstup "Q". D klopný obvod vzorkuje svoj datový vstup "D" na nábežnu hranu signálu privedeného na jeho hodinový vstup "C".

V tomto zapojení sa využíva to, že celkové oneskorenie hodinového signálu zapríčinene výrobou spuštacích pulzov a štartom MKO je väčšie ako potrebný presah dat pred hodinami, ktorý vyžaduje D klopný obvod!!!. Tým sa dosahuje, že signál je navzorkovaný určite len jedenkrát a na správnom mieste, v podstate ihned po zmene, ktorá odstartovala MKO. Tu si treba uvedomiť dôležitú vec, ktorá sa na prvý pohľad zdá nezmyslom. Pôvodný signál sa nevzorkuje v strede datového signálu, ale ihned po odozve hraná (či už padajúca alebo nábežná) -> spuštaci pulz -> štart MKO !!! . Tento spôsob dekódovania sa ukazal hlavným prínosom pre spoločnosť záZNAMU aj pri extrémnych prenosových rýchlosťach akou rýchlosť 9600 Baudov nepochybne je.

OBR.c.5 - Graf. znázornenie vzájomných súvislostí



Takýmto spôsobom vyrábime pôvodne hodiny a data zo zakódovaného signálu. Pre POKEY však ešte nie sú tieto signály vhodné z hľadiska vzájomného časového posunu. Ako bolo už uvedené POKEY vzorkuje svoj vstup DATAIN padajúcou hranou signálu CLKIN. Preto sa ešte hodinový signál najprv otocí o 180 stupňov (zneguje) a trochu spozdi.

Poslednou časťou popísaného interface je multiplexor, čo je elektronický prepinač klasického systému (600B.) alebo "ROS" systému. Multiplexor je ovládaný softwarovo signáлом COMMAND obvodu PIA podobne ako pri Prazskej TURBO2000.

Princíp bol objasnený na blokovej schéme, ďalej popíšem

význam jednotlivych súčiastok a ich vplyv na vlastnosti interface priamo zo schémy zapojenia.

R1,R2,C1-prispôsobujú vstupné úrovne z magnetofonu. S hodnotami v rozpiske pracuje interface pri úrovniach 300 až 1500 mV. Pri iných výstupných napätiach z mgf je nutné odpory pozmeniť.

R3,R4,C2-nastavujú jednosmerný pracovný bod operačného zosilňovača (OZ) na hodnotu 2.1 V. Táto hodnota sa tiež môže lísiť s ohľadom na typ použitého OZ. Môže sa pohybovať v rozmedzí 2.0-2.7 Voltu. Nastavuje sa tak, aby OZ prenášal rovnako kladnú aj zápornu polvinu signálu. Kondenzátor odstraňuje prípadnu kladnú spätnú väzbu, ktorá sa môže uzavierať cez napajací zdroj.

C4,R5,D1,D2-svoju výslednou hodnotou impedancie dynamicky menia stupeň zápornej spätej väzby a tým aj zosilnenie. Diody majú za úlohu stabilizovať amplitúdu pri rôznych vstupných úrovniach.

C5,R6,C6-derivačný člen, ktorý "vylepšuje" hrany signálu z mgf a zároveň odstraňuje prípadné nízkofrekvenčné zložky (brum).

R8/R7-určuje stupeň zápornej väzby t.j. zosilnenie hradieľ I02a+I02b.

R9/R7-určuje stupeň kladnej spätej väzby atým hysteréziu obvodu.

I07-schmidtov klopny obvod s veľmi malou hysteréziou slúži na prevedenie signálu do úrovne TTL s veľmi strmhými hranami.

R10- začažovací odpor I07, ktorý pracuje s otvoreným emitorom.

I02c-slúži na výkonové zosilnenie signálu pre ďalšie použitie.

I03a,I03b,I03c-na výrobu spúšťacích impulzov MK0.

I05-MKO

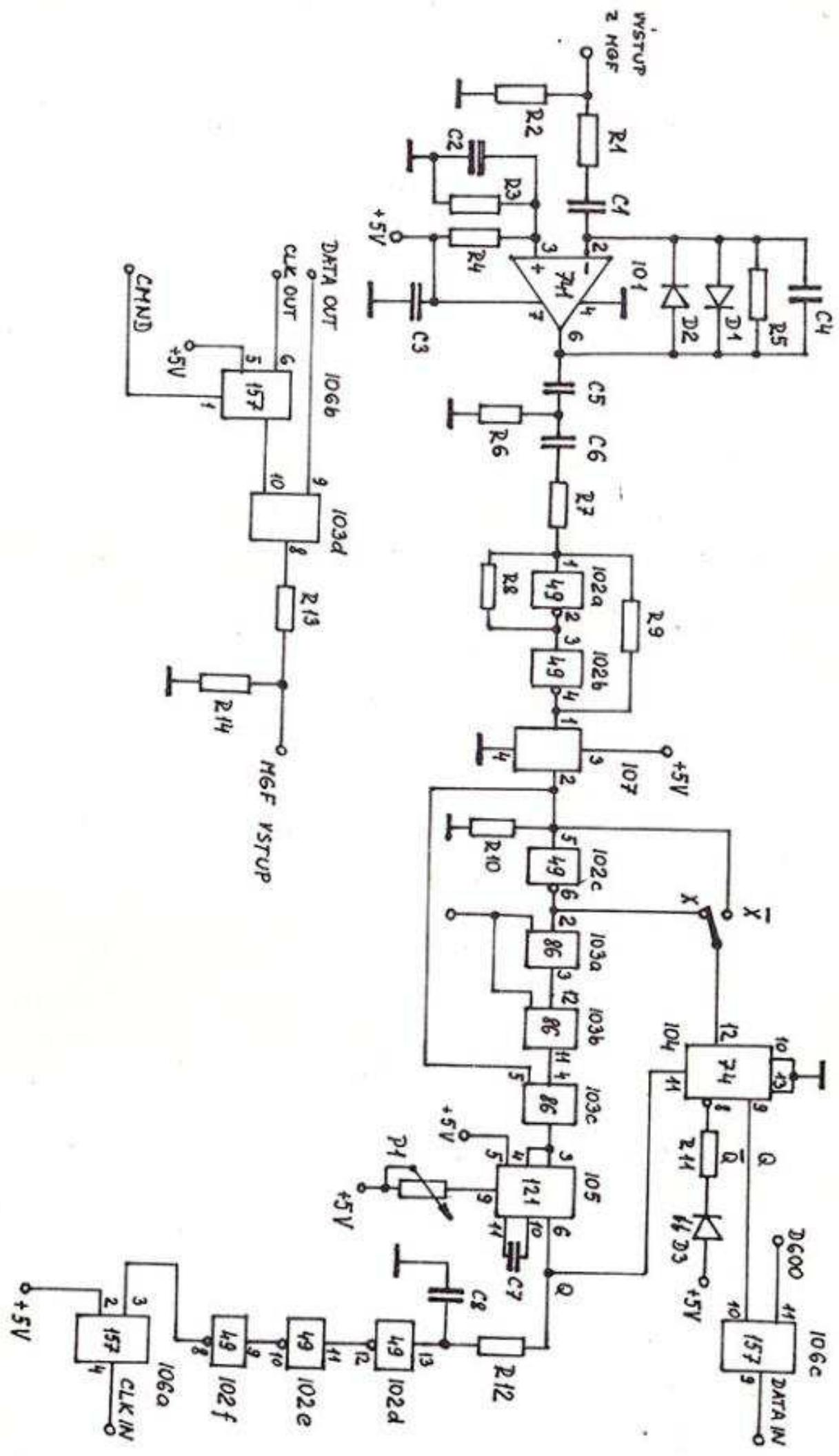
C7,P1-tvoria časovú konštantu MK0, ktorú treba nastaviť na 3/4 periody prenosovej rýchlosťi.

I04- "D" klopny obvod, v ktorom sa vzorkuje signál z mgf a vyrába sa datový signál.

R11,D3-indikujú stav na datovej zbernicí (svieti=log1=DATAIN).

R12,C8-posúvajú časovo hodinový signál-spozdújú ho.

I02d,I02e,I02f-spoločne otáčajú a tvorujú signál DATAIN.



ZÁVER:

Popisovaný systém je určený pre prácu s obyčajným magnetofónom určeným pre záznam zvuku. Prvá verzia vznikla v septembri minulého roku. Najprv pre 3600 Baudov. Ďalej sme sa "neopovážili" ľať, pretože sme si mysleli, že je to prenosová rýchlosť na hranici fyzikálnych možností kazety. Ďalším impulzom k zvýšeniu rýchlosťi bolo stretnutie v Dudincích. Až po tomto stretnutí som doma skúsil ľať ešte rýchlejšie a nechcel som veriť vlastným očiam, pretože stropom sa ukázala rýchlosť až 11000 Baudov (s nezmeneným interface). Chcem zdôrazniť, že vobec nie s HIFI mgf, ale s obyčajným BRG MK27, na ktorom bola vymenená hlava z mono za stereo. Kupodivu však z "kvalitnejšími" magnetofónmi ako TESLA M710, resp. AIWA prenos ťiel spoločivo "len" 8000 Baudami. Pravdepodobne kôli fázovým skresleniam pretože, tieto mgf majú "strmsie" korekcie. Atari mgf bol skúšaný len s pôvodným systémom na 3600 Baudov s dobrými výsledkami. Pri snahе zvýšiť rýchlosť s ATARI datasetom pokus zlyhal pravdepodobne na nahrávacej strane, pretože dataset nemá žiadne frekvenčné korekcie, čo sa už pri zvýšených rýchlosťach prejavilo. Na ďalšie objektívnejšie merania neboli čas ani chut'.

Musím upozorniť na dôležitú vec z hľadiska použitého magnetofónu. Ide o fázu výstupného signálu. Žiaľ v tomto smere si každý výrobca robi čo chce a preto sa môže stať, že práve Váš mgf bude dávať signál v opačnej fáze. V hardware je na túto skutočnosť pamätané a vtedy treba signál na vstup D klopného obvodu zobrať z bodu X'. Toto opačnú fázu Vášho mgf najlahšie poznáte tak, že po tzv. nastavovacom bloku LED dioda zhasne hoci má práve svietiť (musí byť klúč na zbernicu=log1). Vtedy pomôže už spomínaná úprava.

Prvé pokusy doporučujem robiť pri skromných rýchlosťach (3600), kedy by uvedenie do chodu pri použití dobrých súčiastok nemalo byť problémom. Potom už zvýšenie rýchlosťi spočíva len v nastavení inej časovej konštanty MKO. Pri ozivovaní bez meracích prístrojov treba postupovať asi takto:

Nahráme si nejaký čo najdlhší súbor rýchlosťou 3600 Baudov, aby sme nemuseli pásku stále pretáčať. Nastavíme kazetu na začiatok súboru a nejakým príkazom pre čítanie odštartujeme čítanie (názov programu dáme neexistujúci). Trimrom P1 otáčame dovtedy kým LED dioda pravidelne poblikáva a v okamihu medzery medzi blokmi ostáva svietiť. Malo by sa podarit precítať jednotlivé bloky, čo zistíme indikáciou čísel v ľavom hornom rohu. Orientujeme sa podľa svitu LED diody, ktorá musí !!! po prvom tzv. nastavovacom bloku zostať svietiť. Ak pravidelne zhasne máte mgf s opačnou fázou výstupného signálu, vtedy treba prehodiť signál a bráť ho za inventorm - z bodu X'. A znova pretočíme kazetu pred súbor a už by LED dioda mala ostat'

svietiť. Potom by čítanie jednotlivých blokov malo fungovať. Češte jemne otáčame trimrom P1 na jednu i druhú stranu až kým nezačne čítanie vypadávať takto zistíme minimálnu a maximálnu časovú konštantu MKO. Trimr otocíme do stredu týchto hraníc a označíme si polohu pre danú prenosovú rýchlosť. Potom môžeme skúsať zvýšovať prenosovú rýchlosť, čiže si nahráme súbory s rôznymi postupne sa zvyšujúcimi rýchlosťami a skúšame ich čítať. Hodnotu trimra musíme pre každú vyššiu prenosovú rýchlosť znížiť. Horná prenosová rýchlosť bude dana vlastnosťami Vášho mgf, nemala by však nikdy byť nižšia ako 6500 Baudov, inak máte asi opotrebovanú hlavu a treba ju vymeniť.

Poznámka: Pre záznam dat je výhodnejšia stereo hlava (využívame len jednu stopu) z dôvodu väčšieho pomera šírka pásy ku šírke stopy, čo dovoluje menej presné nastavenie kolmosti hlavy.

Najvhodnejší mgf je nejaký prenosný s akustickým výstupom a možnosťou ovládania motorečka signálom MOTORCTRL z počítača. Akustický výstup je doležitý najmä kďaž hľadaniu začiatkov programov, lebo na pár otáčkach mgf je umiestnených niekoľko blokov. Ovládanie je doležité zas pri výpadku bloku, aby mgf ihned zastal a my sme ľahko našli posledný dobrý blok, pretože u niektorých programov, ktoré majú vlastný zavádzací a modifikujú obrazovku nebudem vidieť číslo vadného bloku. Ak ovládanie motoru nie je možné realizovať, treba sa orientovať len podľa charakteristickeho lúpotania, ktoré vyrába pri správnom prenose počítač.

Na demonštráciu uvediem tabuľku s meraniami, ktoré sme realizovali už dôvnejsie. Vždy bol merany čas za aký počítač vyslal celú oblasť pamäte t.j. 0000-\$FFFF.

PRENOSOVÁ RÝCHLOSŤ	MEDZERA MEDZI BLOKMI	VYSL. ČAS
19200	05 (100msec)	1min35s
19200	0F (300msec)	3min05s
9000	05	2min10s
9000	01 (20msec)	1min05s
10000	05	2min15s

Z tabuľky jasne vyplýva, že na výsledný čas môžu značný vplyv aj medzera medzi blokmi, ktorú ako som už uvádzal ROS skracuje na 40 msec. V tom čase, keď boli merania robené ešte nebolo jasné akú medzera medzi blokmi ponecháme a ani akú prenosovú rýchlosť zvážím. (9600Baudov som volil preto, že ju doporučuje komunikačný protokol a aby bola ešte nejaká rezerva). Aj ďalší údaj nie je nezaujímavý: pri rýchlosťi 9000Baudov sa na kazetu C90 zmesti 2.5 Mbyt dat!!

U všetkých Turbosystémov vzniká problém evidencie programov na kazete. Pomocou ROS sa to však dôľa elegantne riešiť tak, že na začiatok kazety si nahráme informačné bloky, ktoré nám vypisú aké programy sú na kazete aj s prípadnými popismi, ktoré sa pri normálnom čítaní ROS budú zobrazovať, čiže netreba nato žiaden zvláštny program (simulácia DIR).

Na ūplný záver chcem uviesť informácie, na ktoré som zabudol pri jednotlivých kapitolách.

K prenositeľnosti programov:

Pri rýchlosťi 9600Baudov aj najmenšia odlišnosť v kolmosti nahrávok znemožní spoľahlivé čítanie. Z vlastnej skúsenosti viem, že u menej kvalitných mgf nie je ich pásková dráha koštruovaná perfektne a preto, ak by sme chceli čítať programy nahráte na inom mgf budeme musieť pravdepodobne prestavovať hlavu. Podobne prehrávanie na DOUBLE mgf nebude asi možné.

Na nahrávanie nie sú vhodné mgf, ktoré majú automatické nahrávanie nastavené na dlhú časovú konštantu (tzv. platičkovú), pretože zareggujú na prvý impulz stiahnutím úrovne a dlho im trvá, kým si zase "pridajú". Naopak mgf, ktoré majú automatiku s krátkou časovou konštantou (tzv. mikrofónne) sú vhodné. Samozrejme najlepšie sú tie, u ktorých môžeme úroveň nastaviť ručne presne na hranicu kedy už zacínajú mierne skreslovať.

K použitým kazetám.

K opakovanej prepisovaniu nie sú vhodné naše EMGETONY, ale inacé prvé nahrávky držia dobre. Väčšinu kaziet mám typu AUDIOSTAR k plnej spokojnosti. Neosvedčili sa mi jedine CR kazety, ale to je asi dané inými hodnotami potrebných korekcií. Vždy však kazeta musí byť OK po mechanickej stránke a nesmie mať samozrejme pokŕčený okraj príp. otlacené miesta. Tu by som uviedol, že odkedy používam tento spôsob záznamu, sa mi ešte nestalo, aby som raz nahrávky program nenacítal nazad pokial ho mechanicky nepoškodil neopatrnou manipuláciou. Aj tak doporučujem pri doležitých súboroch tieto si zdvojit a po vytvorení nahrávky si túto verifikovať príkazom, ktorý nevymaže pôvodny program (ENTER "C:" a pod.). Možme použiť ENTER aj na strojové programy, len musíme zadat neexistujúci názov súboru, aby sa počítač nepokusil program nahráť, pretože by nám vyhlásil chybu 21. Vtedy počítač neustále hľadá neex. súbor a v ľavom rohu sa zobrazuju čísla jednotlivých blokov programu. Pri prípadnej chybe v bloku sa v pravom rohu zobrází číslo bloku, ktoré sa počítaču na prvykrát nepodarilo precítať. Neznamená to však, že musí byť naozaj vadný o tom sa presvedčíme vrátením kazety atď.

K výrobe interface:

U prvom rade treba dobre prezrieť plošný spoj a odstrániť prípadne mikroskraty resp. mikrotrhliny ušetrí Vám to kopce cca pri hľadaní závady.

K použitým polovodičom. Interface bol realizovaný vo viacerých vzorkach a zo súčiastok, ktoré sa pozbaňali z roznych pramenov. Stalo sa nám, že pri použití EDR typu MEV7486PC (z MLR) nechcel chodiť MKO, pretože bol veľmi rýchly a pulzy, ktoré takto vznikli boli nedostatočne na spustenie MKO. Vtedy pomohol kondenzátor 100pF pripojený na vývod c.3 tohto obvodu.

Bašou súčiastkou s dosť širokými rozptylmi parametrov je CMOS. Tu treba skúsať meniť odpór 2M7 v kladnej spätnej väzbe.

S ďalšími obvodmi by nemali byť problémy.

ROZPISKA SÚČIASKOV

R1=1.2 kOhm	C1=22nF/ keram.
R2=33 kOhm	C2=50M/6V eliyt.
R3=680 Ohm	C3=100nF/ ker.
R4=680 Ohm	C4=680pF/ lubov.
R5=100 kOhm	C5=22nF/ ker.
R6=10 kOhm	C6=22nF/ ker.
R7=10 kOhm	C7=22nF/ svitkový
R8=100 kOhm	C8=4.7nF/ svitkový
R9=2.7 MOhm	

Ďalšie blokovacie kondenzátory nie sú v schéme zakreslené a majú hodnotu 68nF alebo 100 nF (súprávajajú sa priamo na napájanie puzdier IO zo strany plošných spojov)

R10=680 Ohm

R11=270 Ohm

Všetky odporu sú miniatúrne lub. typu.

P1=22 kOhm trimer podľa možnosti cermetový, príp. nahradite odporom.

POLOVODICE:

D1,D2=lubovolne krem. diody KA xxx.

D3=LQ xxx

IO1=MAA 741, príp. ekvivalent

IO2=K561LN2 - sovietska nahradza MHB 4049, pozor nesedia vývody (plošný spoj navrhnutý pre sovietsky obvod)

IO3=UCY 7486, CDB486 atd.

IO4=MH 7474, D174

IO5=UCY 74121

IO6=UCY 74157

IO7=MH3ST2

