TESLA ELTOS

MIKROPOČÍTAČOVÁ STAVEBNICE

PETR

TESLA ELTOS, státní podnik,
Institut mikroelektronických aplikací

TESLA ELTOS

MIKROPOČÍTAČOVÁ STAVEBNICE

PETR

TESLA ELTOS, státní podnik,
Institut mikroelektronických aplikací

Příručka tvoří ucelenou uživatelskou dokumentaci mikropočítačové stavebnice PETR. Obsahuje jednak návod k sestavení a oživení stavebnice, jednak další podrobné informace, jak již hotovou mikropočítačovou stavebnici PETR využívat.

Protože stavebnice je, jako polytechnická pomůcka, určena k širokému použití, je i tato příručka psána populární formou spíše než jako strohý, odborný text. Pro názornost je doplněna řadou příkladů. Místy je text proložen orientačními otázkami k nimž lze nalézt odpovědi v jedné z příloh. V příručce uvedené podrobné informace o vnitřní struktuře stavebnice umožňují i profesionální využití mikropočítačové stavebnice PETR, včetně modifikací obvodového zapojení a řídicího programu.

Přehled revizí dokumentace		
verze	datum	poznámka
01	květen 87	IMA informace 6/87
02	duben 88	původní text uživatelské příručky
03	červen 88	korektura

Příručku připravil kolektiv autorů:
Ing. Tomáš Trpišovský, CSc. [1],
Ing. Jaroslav Pecina [1] a
RNDr. Petr Couf [2].

Na vypracování příkladů a ověřování stavebnice se podíleli Petr Beneš a Pavel Janoušek [3].

Za pečlivou přípravu tiskové předlohy, obrázků a korektury textu děkují autoři svým spolupracovnicím Evě Hanzalové [1] a Jitce Živnůstkové [1].

---

[1] TESLA ELTOS - Institut mikroelektronických aplikací, Praha
[2] MFF UK, KKI, Praha
[3] studenti, SPŠE Ječná, Praha

Obsah

PŘÍLOHA 1. Výpis programu PETR

PŘÍLOHA 2. Schema:

- zapojení mikropočítače PETR
- obvod klávesnice a displeje
- obvod připojení magnetofonu
- stabilizátoru napětí
- zapojení logické sondy
- zapojení vnější paměti s expanderem
- rozložení součástek na základní desce
- blokové schema mikropočítače PETR

Seznam součástek:

- stavebnice PETR
- logické sondy ke stavebnici PETR
- pro rozšíření stavebnice PETR

PŘÍLOHA 3. Odpovědi na otázky v textu

PŘÍLOHA 4. Tabulky 8048

- seznam instrukcí mikroprocesoru 8048/8049 seřazený vzestupně podle vnitřního kódu
- abecedně seřazený seznam instrukcí mikroprocesoru 8048/8049
- funkční přehled instrukcí mikroprocesoru 8048/8049

Tabulka instrukcí PETR

Každý návod k použití, i sebestručnější, zdržuje. Nejinak se povede i Vám, začnete-li číst tuto příručku od začátku do konce. Proto si namísto čtení příručku pouze prolistujte, přitom lze vynechat 4. a 7. kapitolu a všechny přílohy.

Poté si podrobně přečtěte několik slov ÚVODEM... (kap. 1) a můžete rovnou přistoupit k SESTAVENÍ A OŽIVENÍ STAVEBNICE (kap. 6).

Všechny další kapitoly a přílohy obsahují pouze doplňkové informace, přesto ale věříme, že je příležitostně využijete.

Místa označená v textu [?x] si doplníte sami, přitom seznam odpovědí respektive návodů k řešení je uveden v jedné z příloh.

Příjemnou zábavu i dobré výsledky Vám přeje

kolektiv autorů.

1. ÚVODEM ...

Tato příručka souhrnně popisuje mikropočítačovou stavebnici PETR a je podrobným návodem k jejímu využití. Text je psán s důrazem na názornost a je doplněn řadou praktických příkladů. Vzhledem k možnosti profesionálního využití stavebnice jako polotovaru universálního zapojení mikropočítače řady 8048 je popsána i implementace mikropočítače PETR a to jak technické řešení, tak řídicí program.

Text příručky je rozdělen do sedmi částí, které lze stručně charakterizovat takto:

1. kapitola [Úvodem ...] obsahuje základní informace o uspo- uspořádání příručky a o významu jednotlivých částí textu.

2. kapitola [Popis mikropočítače PETR] popisuje virtuální mikropočítač PETR, zejména jeho architekturu. Obsahuje také přehled všech základních instrukcí mikropočítače.

3. kapitola [Ovládání mikropočítače PETR obsluhou] je věnována popisu ovládání mikropočítače. Jsou názorně popsány všechny ovládací příkazy a manipulace s kazetovým magnetofonem.

4. kapitola [Popis implementace mikropočítače PETR] je určena hlavně těm zájemcům, kteří chtějí stavebnici využít jako universální systém s jednočipovým mikropočítačem řady 8048. Takové využití ovšem předpokládá hlubší znalost jednočipového mikropočítače řady 8048, jeho podpůrných obvodů a způsobu programování, proto i text této kapitoly je psán stručněji a možná i méně názorněji. Je však doplněn seznamem odkazů na další odbornou literaturu.

5. kapitola [Doplňky mikropočítače PETR] obsahuje popis doplňků vlastního mikropočítače, které jsou součástí stavebnice.

6. kapitola [Sestavení a oživení stavebnice] je podrobným návodem k tomu, jak sestavit a oživit jednotlivé části mikropočítače.

7. kapitola [Příklady programů a aplikací] je návodem k využití mikropočítače PETR. Obsahuje příklady programů a doplňkových zapojení některých prakticky ověřených aplikací.

Příručka je dále doplněna řadou příloh, které obsahují kromě schemat zapojení i stručný přehled některých často potřebných informací vybraných z textu. Dále jsou přiloženy přehledové tabulky pro jednočipový mikropočítač 8048 a výpis řídicího programu mikropočítače PETR. Tento listing doplňuje popis řídicího programu, uvedený v kap. 4 a je určen především zájemcům o univerzální využití stavebnice PETR.

Zbývá několik slov k použité terminologii. Přestože mikropočítač PETR je určen jako polytechnická pomůcka, není zřejmě nutné uvádět zvláštní výklad základních pojmů. Někdy je v textu v závorce uveden pro informaci anglický termín, jak se s ním lze běžně setkat v odborné literatuře. Konkrétně o mikropočítačové stavebnici PETR lze vysvětlit, že mikropočítač PETR se označuje jako stavebnice proto, že je ho třeba ze součástek sestavit a nikoliv proto, že by jej bylo možno stavebnicově rozšiřovat (jako lze rozšiřovat např. mikropočítač SAPI-1 připojením dalších stavebnicových prvků). Pokud jste četli pozorně až sem víte, že na jednom místě byl uveden "virtuální mikropočítač". Jako virtuální (zde ve smyslu zdánlivý) mikropočítač lze stavebnici PETR označit, pokud se na něj díváme z hlediska aplikátora jako na "černou skříňku" s určitou architekturou a množinou základních instrukcí (viz kap. 2). Přitom není důležité jeho vnitřní technické a programové řešení.

Na jiných místech je zase zmínka o jednočipovém mikropočítači řady 8048. Jednočipový mikropočítač je jeden poměrně složitý integrovaný obvod (obsahuje zhruba 20 tisíc tranzistorů na křemíkové destičce – čipu o velikosti 22 mm^2) a tvoří jádro mikropočítače PETR.

První verze stavebnice PETR vznikla v polovině roku 1986 a je příkladem dobré spolupráce autorského kolektivu pracovníků TESLA ELTOS – Institutu mikroelektronických aplikací a MFF UK v Praze. Jistou inspirací byl autorům malý řídicí mikropočítač KOSMOS, vyráběný v NSR [1]. PETR je sice funkčním rozšířením uvedeného mikropočítače, liší se však jak konstrukčně, tak i po stránce technické realizace a řídicího programu. Nicméně jako upomínka na inspirační vzor je záměrně zobrazováno písmeno C jako základní nápovědná zpráva mikropočítače PETR.

2. POPIS MIKROPOČÍTAČE PETR

Mikropočítač PETR je víceúčelová mikroprocesorová stavebnice malého řídicího počítače. Je určen především pro polytechnickou výchovu. Proto se také dodává ve formě stavebnice, kterou lze po sestavení programovat jako běžný mikropočítač.

Původním záměrem autorů bylo připravit mikropočítač s velmi jednoduchou architekturou, přitom však umožňující přímé řízení reálných aplikací - modelů, měřicích a vyhodnocovacích přípravků, signalizačních zapojení, kolejišť ap. Tomuto záměru bylo podřízeno technické a konstrukční řešení stavebnice. Jistým omezením byl požadavek na co nejnižší cenu, proto je řada funkcí mikropočítače PETR realizována jeho řídicím programem, nikoliv obvodovým zapojením.

Jádro mikropočítačové stavebnice PETR tvoří jednočipový mikropočítač řady 8048. Po prvních funkčních zkouškách stavebnice se obvodové řešení ukazalo jako perspektivní. Proto byl řídicí program stavebnice přepracován tak, aby umožnil zcela universální využití stavebnice. To spočívá v možnosti doplnit obvodové řešení stavebnice PETR vlastním řídicím programem určeným pro složitější cílovou aplikaci, např. pro automatizaci topení nebo zabezpečovací signalizaci v rodinných domcích aj. Stavebnice tedy může sloužit i jako určitý polotovar pro takovéto aplikace.

V dalších odstavcích této kapitoly je popsán virtuální mikropočítač PETR (viz Úvodem ...) tak, jak se jeví programátorovi, který s ním má pracovat.

2.1 Architektura mikropočítače PETR

Mluvíme-li o architektuře mikropočítače, máme na mysli charakteristiku jeho základních částí - tu tvoří obvykle procesor, paměť a blok styku s vnějším prostředím (periferiemi) a způsob jejich vzájemného propojení.

Procesor

Mikropočítač PETR má jednostřadačový procesor a umí zpracovávat instrukce (1 až 27) a číselné operandy v intervalu 0 až 255. Jednostřadačový procesor má tedy jeden osmibitový střadač ACC (Accumulator), v němž lze uchovávat hodnotu

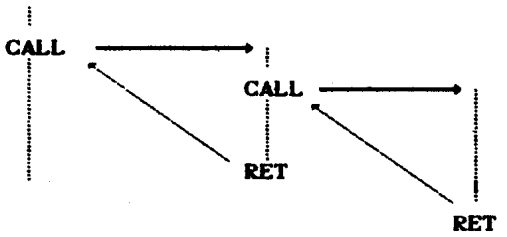
jednoho operandu. Střadač vždy slouží jako místo pro úschovu buď prvního (jsou-li dva) nebo jediného operandu právě zpracovávané instrukce. Obvykle se do střadače uloží i výsledek právě provedené instrukce.

Kromě střadače je součástí procesoru zvláštní jednobitový **příznak F** (Flag), do něhož se ukládají výsledky typu "ano/ne" některých instrukcí. Například výsledek instrukce pro porovnání obsahu střadače s obsahem určitého místa v paměti. Stav příznaku F je možno testovat a větvit pak program podle předem vypočtené podmínky.

Důležitou částí procesoru je **čítač instrukcí PC** (Program Counter). Tento čítač ukazuje vždy na tu instrukci programu, která je určena k provedení v dalším kroku. Procesor si po provedení instrukce čítač PC automaticky upravuje. Přitom záleží na typu provedené instrukce a na výsledku.

?1 Bude čítač adres PC po provedení jedné instrukce ukazovat vždy na v pořadí další následující instrukci zapsanou v programu ?

Jinou důležitou částí procesoru je zásobník. Je to vlastně vnitřní paměť procesoru určená pro čtení i zápis (RWM - Read Write Memory). Zásobník mikropočítače PETR má 10 položek a slouží pro ukládání návratových adres při volání podprogramů. Se zásobníkem pracují výhradně instrukce CALL (volání podprogramu) a RET (návrat z podprogramu). Pro potřeby ladění programu lze zobrazovat ukazovátko do zásobníku SP (Stack Pointer). Pro prázdný zásobník je SP=0. Závěrem připomeňme ještě princip využívání zásobníku, který je obdobný u většiny mikropočítačů. Zásobník pracuje systémem poslední (zapsaná návratová adresa) dovnitř - první ven (LIFO Last-In-First-Out). Tento princip umožňuje postupné vnořování jednoho podprogramu do druhého a je znázorněn na obr. 1.



Obr. 1 Vnoření dvou podprogramů

Poznámka:

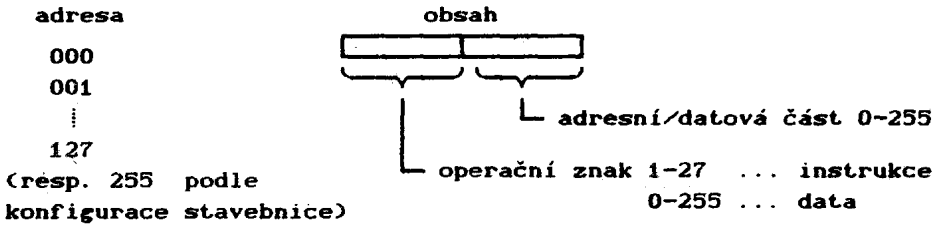
Při ladění programu lze na displeji kdykoliv zobrazit stav ukazovátka do zásobníku SP. Tak je možno zjistit, kolik podprogramů je v té chvíli do sebe vnořených. Nelze však prohlížet obsah zásobníku adres a tedy ani zjišťovat, které podprogramy to jsou. To je však zřejmé z výpisu laděného programu.

?2 Kolik podprogramů lze do sebe nejvýše vnořit tak, aby nedošlo k přeplnění zásobníku ?

Paměť

Paměť mikropočítače PETR se skládá z buněk. Každá buňka paměti může obsahovat buď jednu instrukci, nebo data z rozsahu 0 až 255. Buňka paměti se skládá ze dvou částí: z operačního znaku, má hodnotu 1 až 27 pro instrukce, nebo 0 až 255 pro data, a z adresní/datové části. V té je uloženo číslo z intervalu 0 až 255, které má význam buď adresové části instrukce, nebo hodnoty uložených dat.

Základní kapacita paměti je 128 buněk, pomocí přídavného modulu ji lze rozšířit na 256 buněk. Každá buňka paměti je označena adresou - celým číslem z intervalu 0 až 255.



Obr. 2 Uspořádání paměťové buňky

Je zřejmé, že podle hodnoty operačního znaku nelze rozlišit, zda na určité adrese je uložena instrukce programu či data. Toto rozlišení záleží na programátorovi resp. obsluze, která program spouští.

Programátor plní paměť nejčastěji příkazem (MEM), kdy přímo vkládá obsah jak operačního znaku, tak adresní/datové části. Je-li buňka paměti využívána pro uložení dat (nikoliv jako instrukce), na operačním znaku nezáleží.

Poznámka:

Počítače s takto uspořádanou pamětí (jedna paměť společná pro program i data) se označují jako počítače "von Neumannova typu" na počest amerického matematika maďarského původu Johna von Neumanna (*1903 †1957), který významně přispěl k rozvoji teorie her a teorie počítačů.

?3 Kolik kroků může nejvýše obsahovat program pro mikropočítač PETR ?

Periferie

Procesor, tedy i běžící program, může jednak komunikovat s obsluhou, jednak s řízeným objektem.

Ke styku s obsluhou slouží membránová klávesnice a sedmisegmentový displej. Instrukční soubor umožňuje snímat stav klávesnice a ovládat všechny segmenty displeje. To znamená, že lze vytvářet programy, jejichž činnost může řídit obsluha tlačítky membránové klávesnice. Program přitom může své výsledky nebo zprávy zobrazovat na segmentový displej stavebnice. Tak lze například mikropočítačem PETR snadno realizovat číslicový voltmetr, který má tlačítky přepinatelné rozsahy a další podobné aplikace.

Pro připojení do vnějšího prostředí - obvykle do řízeného objektu - má mikropočítač PETR vyčleněno 18 vstupních/výstupních linek. Základní instrukce mikropočítače přitom umožňují tyto linky přímo ovládat a vyhodnocovat, což velmi usnadňuje vytváření řídicích programů. 16 linek je sdruženo do dvou osmibitových bran P1,P2 (Ports), z nichž brána P1 je obousměrná a brána P2 pouze výstupní. Instrukce umožňují pracovat buď s celou bránou najednou, nebo s jednotlivými linkami. Zbylé dvě linky T0 (Test Pin) a INT (Interrupt) mají zvláštní význam. Linka T0 je pouze vstupní, její stav je testovatelný instrukcí podmíněného skoku, linka INT je rovněž vstupní, má význam žádosti o přerušení.

K mikropočítačové stavebnici PETR lze bez úprav přímo připojit běžný kazetový magnetofon, který je možno využívat pro úschovu již hotových programů. Přitom se používá záznam kompatibilní se záznamem počítačů IQ-151 tak, aby byla zajištěna vzájemná přenositelnost nahrávek. Podrobněji je práce s magnetofonem popsána v kap. 3.

Přerušovací systém

Přerušovací systém mikropočítače PETR je řešen obdobně jako u mikroprocesoru 8080. Počítač má jednoúrovňový přerušovací systém, jediným zdrojem přerušení je nízká úroveň na vstupní lince INT.

Přijme-li procesor požadavek na přerušení,
- zablokuje přijetí dalších požadavků (zamaskuje přerušení),
- dokončí rozpracovanou instrukci a
- provede "mimo pořadí" jednu předem definovanou instrukci.

Instrukce prováděná při přerušení může být libovolná přípustná základní instrukce, tedy např. HALT, JMP, CALL ap Tuto instrukci je nutno umístit do programu na adresu 0. Po provedení této vnucené instrukce pokračuje program dále z toho místa, ve kterém byl přerušen, pokud ovšem v důsledku vnucené *řídící instrukce* není určeno jinak.

Využitím přerušení lze řídit činnost programu vnější událostí nebo stavem řízeného objektu. Tak lze například snadno naprogramovat hlídací časovač WDT (Watch Dog Timer). Tento časovač je programem stále cyklicky inkrementován (zvětšován o 1) a nuluje se jen při přerušení, které má význam odezvy řízeného objektu. Pokud se čítač přeplní, znamená to, že řízený objekt neodpověděl v předepsané době. Program pak může vypsat zprávu o chybě nebo jinak ošetřit tuto výjimečnou situaci.

Po zapnutí mikropočítače PETR je přerušení zakázáno, to znamená, že nenastane, ať je linka INT v jakémkoli stavu. Jednou ze základních instrukcí (INT *i*) lze přerušení povolit nebo zakázat - to je dáno hodnotou operandu i.

Pokud je přerušení povoleno, testuje se před provedením každé instrukce, zda v průběhu zpracování minulé instrukce nepřišla žádost o přerušení (linka INT=0). Pokud ano, zpracuje se mimo pořadí instrukce umístěná v paměti na adrese 0, jinak program pokračuje normálně dále.

Po každém zpracovaném přerušení je procesor ve stavu "přerušení zakázáno" a před dalším očekávaným výskytem je proto nutné jej opět povolit instrukcí INT i .

Přerušení se vyvolá *logickou úrovní 0* na vstupní lince INT. Pak je nutné počítat s tím, že pokud tato úroveň na lince INT trvá, toto přerušení nastane pokaždé, když ho program povolí. Jednu a tutéž žádost o přerušení může tedy počítač obsloužit i několikrát. Tomu lze předejít, pokud žádosti o přerušení na lince INT mají tvar pulsů s minimální přípustnou délkou 7,5 μs. Maximální délka není omezena technickým řešením počítače, ale konstrukcí řídicího programu (puls musí skončit dříve, než program znovu povolí přerušení instrukcí INT *i*). Návod, jak zajistit generování vhodných pulsů, je uveden jako příklad v kapitole 7.

2.2 Instrukční soubor mikropočítače PETR

Procesor mikropočítače zpracovává celkem 27 instrukcí. Lze je rozdělit do několika skupin:

- instrukce pro přesun dat v počítači,
- aritmetické a logické operace,
- instrukce pro řízení výpočtu (především skoky),
- periferní operace,
- ostatní instrukce (prodleva, přerušení).

Dále je uveden podrobný popis všech instrukcí. U každé instrukce je kromě jejího názvu v jazyce symbolických adres uvedeno mnemotechnické vyjádření její funkce, její vnitřní reprezentace (tedy strojový kód) a úplný popis.

Jako názvy instrukcí v jazyce symbolických adres jsou vesměs použity zkratky standardně rozšířených anglických termínů.

V mnemotechnickém popisu funkce značí symbol CONST, MEM a @MEM hodnotu druhého operandu instrukce, a to:

CONST	...	je přímá konstanta, uvedená jako operand instrukce,
MEM	...	je obsah buňky paměti na adrese dané operandem instrukce a
@MEM	...	označuje obsah buňky paměti na adrese dané obsahem buňky na adrese dané operandem instrukce.

Tedy CONST má význam dat, MEM má význam adresy a @MEM má význam nepřímé adresy.

Kromě běžných matematických značek se v textu rozlišuje značka pro rovnost (=) a značka pro přiřazení (:=). Zápis ACC=0 vyjadřuje, že obsah střadače je 0, zatímco ACC:=0 značí, že střadači se přiřadí hodnota 0 (střadač se touto hodnotou naplní).

Strojová reprezentace instrukce představuje obsah jedné buňky paměti. V popisu je uvedena tak, jak se zobrazuje obsluze počítače při prohlížení paměti - jako pěticiferné číslo, jehož první dvě číslice udávají operační znak a zbylé tři číslice operand (zpravidla adresu). Pro data i adresy platí, že musí být z intervalu 0 až 255, není-li v popisu instrukce výslovně uvedeno jinak.

V dalším textu jsou souhrnně popsány jednotlivé skupiny instrukcí a dále je uveden jejich abecedně seřazený podrobný popis.

Instrukce pro přenos dat
(LDA, LDAI, LDC, STA, STAI)

Všechny instrukce pro přenos dat kopírují data z paměti do střadače nebo opačně. Přitom se původní data nemění. Žádná z instrukcí pro přenos dat neovlivňuje příznak F. Data musí být umístěna výhradně v adresní/datové části paměťové buňky. Do místa operačního znaku nelze ani zapisovat, ani z něj nelze číst.

Aritmetické a logické instrukce
(ADD, SUB, AEQ, ALT, AGT, AND, OR, CPL)

Všechny instrukce tohoto typu buď ovlivňují stav příznaku F nebo hodnotu střadače ACC nebo obojí tak, jak je uvedeno v tabulce:

instrukce	ovlivňuje ACC	F
ADD, SUB	■	■
AEQ, ALT, AGT		■
AND, OR, CPL	■	

Instrukce pro řízení výpočtu
(JMP, JMPI, JF, JT, CALL, RET)

Všechny instrukce pro řízení výpočtu ovlivňují obsah čítače adres PC. Lze je proto využít pro programování cyklů, podmíněných větvení programů, volání podprogramů apod. Přitom se obsah střadače ACC ani stav příznaku F nemění.

Instrukce pro periferní operace
(P1OUT, P2OUT, P1IN, DISP, KEY)

Do této skupiny patří jednak instrukce pro ovládání vstupních a výstupních linek, jednak instrukce pro čtení klávesnice a zápis na displej stavebnice PETR.

Tyto instrukce nemění příznak F a s výjimkou vstupních instrukcí P1IN a KEY ani obsah střadače.

Ostatní instrukce
(HALT, NOP, INT)

Tyto instrukce ovlivňují stav procesoru mikropočítače PETR. Umožňují ukončit program (HALT), vkládat prázdné instrukce (NOP) a ovládat přerušovací systém (INT).

Upozornění: Příklady, uváděné u popisu jednotlivých instrukcí nejsou samostatné programy. Při jejich praktickém ověřování je proto třeba je buď krokovat, nebo vhodně ukončit např. instrukcí HALT.

ADD *adresa*

Přičtení obsahu paměti ke střadači
(Add Memory to Accumulator)

kódování: 07.*aaa*

funkce: ACC := ACC + MEM

Přičtení dat uložených v paměti na uvedené adrese ke střadači. Je-li výsledek větší než 255, uloží se do střadače výsledek zmenšený o 256 a příznak F se nastaví na hodnotu 1. Jinak je hodnota příznaku F nula.

Příklad:

```
adr   kód       instrukce     komentář
000   04.100    LDC  100      ;naplní ACC:=100
001   07.010    ADD  010      ;ACC:=100+50 , F:=0
002   07.011    ADD  011      ;ACC:=150+200  tedy
 |                            ;výsledek je ACC:=94 (tj.350-256)
010   00.050                 ; a příznak F:=1
011   00.200
```

Všimněte si, že příznak F zde má význam přenosu při sčítání a lze jej tedy využít při sčítání větších čísel než 255.

AEQ *adresa*

Test na rovnost obsahu střadače s obsahem paměti

(Accumulator Equal to Memory ?)

kódování: **10.*aaa***

funkce: **F := (ACC = MEM)**

Porovná obsah střadače s hodnotou dat uložených na dané adrese. Jsou-li si obsahy rovny, nastaví se příznak F na hodnotu 1, jinak je hodnota příznaku F nula.

Příklad:

```
adr   kód       instrukce     komentář
000   05.010    LDA  010      ;naplní ACC:=111
001   10.011    AEQ  011      ;ACC=222, tedy F:=0
002   10.010    AEQ  010      ;ACC=111, tedy F:=1
 |
010   00.111
011   00.222
```

Poznámka:

Symbolický zápis F:=(ACC=MEM) značí, že se vyhodnotí podmínka ACC=MEM a výsledek (tj. ano/ne) se zapíše jako hodnota příznaku F.

AGT *adresa*

Test zda je obsah střadače větší než obsah paměti

(Accumulator Greater then Memory ?)

kódování: 12.*aaa*

funkce: F := (ACC > MEM)

Porovná obsah střadače s hodnotou dat uložených na dané adrese paměti. Je-li obsah střadače větší, nastaví se příznak F na hodnotu 1. Jinak se nastaví na hodnotu 0. Výsledkem instrukce je pouze nastavení příznaku F, obsah střadače ani paměti se při provedení instrukce nezmění.

Příklad:

```
adr   kód       instrukce     komentář
000   04.001    LDC  001      ;naplní ACC:=1
001   06.005    STA  005      ;uloží na adresu 005  MEM:=1
002   12.005    AGT  005      ;ACC=1, tedy F:=0
003   12.006    AGT  006      ;ACC>0, tedy F:=1
 |
005   **.***
006   00.000
```

ALT *adresa*

Test zda je obsah střadače menší než obsah paměti

(Accumulator Less then Memory ?)

kódování: 13.*aaa*

funkce: F := (ACC < MEM)

Porovná obsah střadače s hodnotou dat uložených na dané adrese paměti. Je-li obsah střadače menší, nastaví se příznak F na hodnotu 1. Jinak se nastaví na hodnotu 0. Výsledkem instrukce je pouze nastavení příznaku F, obsah střadače ani paměti se při provedení instrukce nezmění.

Příklad:

```
adr    kód        instrukce     komentář
000    19.010     LDAI 010      ;naplní ACC:=33
001    13.010     ALT  010      ;ACC<40, tedy F:=1
002    13.040     ALT  040      ;ACC=33, tedy F:=0
 |
010    00.040
 |
040    00.033
```

#4 Jsou tyto dva úseky programů funkčně stejné ?

```
(1)          :
             AEQ  100
          ┌─ JF   ANO
NE:       │   ...
  :       │
  :       │
ANO:      └─> ...
```

```
(2)          :
             ALT  90
          ┌─ JF   NE
          │  AGT  90
          │  JF   NE
          │  JMP  ANO
          └─> ...
NE:
  :
  :
ANO:      ┌─> ...
```

AND *adresa*

Logický součin obsahu střadače s obsahem paměti

(Logical And Accumulator with Memory)

kódování: 15.*aaa*

funkce: ACC := ACC and MEM

Provede logickou operaci and s obsahem střadače a obsahem buňky paměti na dané adrese. Výsledek se uloží do střadače. (Oba parametry se rozvinou v binární soustavě na osmi bitech a provede se logická operace and na odpovídajících pozicích. Výsledek se interpretuje opět jako zápis čísla v binární soustavě.) Příznak F není ovlivněn.

Příklad:

```
adr   kód       instrukce     komentář
000   04.015    LDC  015      ;naplní ACC:=15 (binárně 00001111)
001   15.010    AND  010      ;log. součin bude ACC = 00001111
 ¦                            ;                 MEM = 00010001
 ¦                            ;      výsledek v ACC := 00000001
010   00.017     ;tj. binárně 00010001
```

Po provedení instrukce AND bude tedy ve střadači výsledek ACC = 001 .

Pravdivostní tabulka pro logický součin je:

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

CALL *adresa*

Skok do podprogramu
(Call Subroutine)

kódování: 23.*aaa*

Vyvolání podprogramu na dané adrese. Do zásobníku návratových adres se automaticky uloží adresa instrukce následující za instrukcí CALL a ukazatel zásobníku se zvětší o jedničku. Při přeplnění zásobníku dojde k chybě E 005 a výpočet se zastaví. Jako podprogram lze s výhodou psát úsek programu, který by se jinak opakoval na několika místech programu. Podprogram obvykle končí instrukcí RET, která zajistí návrat zpět za instrukci CALL. Podprogramy lze do sebe vnořovat jak je znázorněno na obr. 2. Zásobník mikropočítače PETR umožňuje vnoření až 10 podprogramů.

Příklad:

```
adr   kód       instrukce     komentář
000   23.100  ┌─>CALL 100     ;testuje stav vstupní linky P14
001   11.000  └─ JF   000     ;cyklus dokud je P14=1
002   01.000    HALT          ;jinak STOP
 |
099   00.001    ; srovnávací konstanta
100   16.004    P1IN 004      ;přečte P14 do ACC
101   10.099    AEQ  99       ;porovná s konstantou
102   24.000    RET           ;a vrátí se do hlavního programu
```

Kdy se v tomto programu nastavuje příznak F, který testuje instrukce JF ?

CPL

Invertuje obsah střadače

(Complement Accumulator)

kódování: 14.000

funkce: ACC := 255 - ACC

Obsah střadače je (bitově) invertován, do střadače se dosadí hodnota 255 - původní obsah střadače. Instrukce neovlivňuje příznak F.

Příklad:

```
adr   kód      instrukce     komentář
000   04.003   LDC  003      ;naplní ACC:=003 (binárně 00000011)
001   14.000   CPL           ; výsledek je  ACC:=11111100
```

Tedy po provedení instrukce CPL bude ve střadači výsledek ACC=252.

DISP *parametr*

Zobrazení na displej
(Display)

kódování: 02.rrr

funkce: zobrazení

Instrukce DISP umožňuje zobrazení informací na sedmisegmentovém displeji, který je součástí stavebnice. Výpisy na displej lze tedy vytvářet přímo z programu.

Podle adresní části instrukce se výpis provádí takto:

1) Je-li parametr rrr=0, zobrazí se obsah střadače dekadicky na třech pravých pozicích displeje, obsah tří levých pozic se nemění:
Např.:

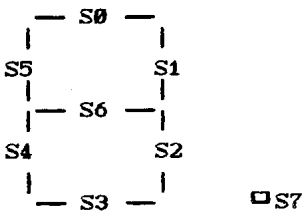
```
adr  kód      instrukce   komentář
000  04.123   LDC  123    ;ACC := 123
001  02.000   DISP 0      ;zapíše do 3 pravých pozic displeje
```

Byl-li obsah displeje před zápisem [x][x][x][x][x][x]

bude po zápisu [x][x][x][1][2][3]

2) Je-li parametr rrr=255, vymaže se obsah všech pozic displeje (zhasnou se všechny segmenty ve všech pozicích).

1) Je-li parametr rrr = 1 až 6, přepíše se obsah střadače do zvolené pozice displeje 1 až 6. Přitom jednotlivé bity binárního rozvoje střadače přímo ovládají jednotlivé segmenty displeje (1=svítí, 0=zhasnuto). Přiřazení jednotlivých segmentů je pak:



segment	dekadicky	binárně
S0	4	00000100
S1	16	00010000
S2	128	10000000
S3	32	00100000
S4	64	01000000
S5	8	00001000
S6	2	00000010
S7	1	00000001

Obr. 3 Přiřazení segmentů

Tímto způsobem lze zapsat na libovolnou pozici displeje libovolný znak vytvořený kombinací svítících a zhasnutých segmentů.

	dekadicky
0	252
1	144
2	118
3	182
4	154
5	174
6	238
7	148
8	254
9	190

Příklad:

Pro rozsvícení písmene S na pravé krajové pozici displeje potřebujeme rozsvítit segmenty 0, 5, 6, 2 a 3. Tedy do střadače je třeba zapsat

S0+S5+S6+S2+S3 = 4+8+2+128+32 = 174

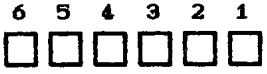
a pak zobrazit znak na 1.pozici.

Úlohu řeší např. následující posloupnost instrukcí

```
LDC   174
DISP  1
```

Obr. 4 Zobrazení dekadických cifer

Přiřazení displejů je



pravý krajní

#6 Tabulku pro zobrazení písmen a symbolů si laskavě doplňte sami podle výše uvedeného návodu. Přitom tvary písmen lze volit různě, např. c lze zobrazit jako nebo apod. Některá písmena se zobrazují těžko, např. k jako , některá zobrazit nelze - např. x .

písmeno	a	b	c	d	e	f	g	h	i	j	k	l	m
tvar													
hodnota													

písmeno	n	o	p	r	s	t	u	v	x	y	z		
tvar													
hodnota													

symbol	+	-	?	!	°								
tvar													
hodnota													

číslice	0	1	2	3	4	5	6	7	8	9			
tvar													
hodnota													

HALT

Zastavení výpočtu

(Halt)

kódování: 01.000

Zastaví výpočet. V levé krajní pozici displeje se zobrazí písmeno H a adresa instrukce HALT, na níž byl program zastaven. Počítač přejde do režimu komunikace s obsluhou.

Příklad:

```
adr  kód       instrukce    komentář
000  03.000    NOP   000    ;dvě prázdné instrukce
001  03.000    NOP   000
002  01.000    HALT         ;zastavení programu
```

Po zastavení tohoto programu se na displeji zobrazí zpráva:

Mikropočítač PETR čeká na zásah obsluhy. Instrukce HALT je jedinou instrukcí, kterou může rozpracovaný výpočet sám sebe ukončit. Jinak může být ukončen zásahem obsluhy nebo při chybě.

INT i

Povolení / zákaz přerušení

(Interrupt Enable / Disable)

kódování: 27.00i

Instrukce umožňuje povolit nebo zakázat přerušení. Adresní část instrukce i může nabývat pouze hodnot 0 nebo 1. Přitom hodnota 0 znamená přerušení zakázat, hodnota 1 přerušení povolit.

Příklad: Program cyklicky testuje klávesnici a je-li stisknuto nějaké tlačítko, zobrazí jeho kód do pravé části displeje (3 cifry). Při výskytu vnějšího přerušení se program zastaví. Program začíná na adrese 001. Hned první instrukcí je povoleno přijmout žádost o přerušení. Při vnějším přerušení, které může nastat kdykoliv v průběhu programu, se skočí na adresu 000. Tam je připravena instrukce HALT, na které se program zastaví.

```
adr   kód       instrukce      komentář
000   01.000    HALT           ;stop při vnějším přerušení
001   27.001    INT  1         ;začátek programu je od adresy 001
002   26.000  ┌→KEY            ;dále v cyklu testovat
003   10.007  │ AEQ  007       ;a zobrazovat kódy tlačítek
004   11.002  ├ JF   002
005   02.000  │ DISP 000
006   09.002  └ JMP  002
007   00.128    ;data pro testování klávesnice
                ;128 je příznak, který vrací klávesnice,
                ;není-li stisknuto žádné tlačítko
```

Podrobně je princip přerušovacího systému a způsob jeho využití popsán v odstavci 2.1.

JF *adresa*

Skok podle příznaku F

(Jump if Flag)

kódování: 11.*aaa*

Podmíněný skok na danou adresu. Je-li hodnota příznaku F rovna 1, potom se provede skok na adresu uvedenou v instrukci. Jinak se pokračuje v sekvenčním vykonávání instrukcí, tj. provede se další v programu zapsaná instrukce. Hodnota příznaku F ani střadače se při tom nemění.

Příklad: Vyhledání položky v tabulce podle obsahu střadače.

```
adr   kód        instrukce     komentář
000   06.050       STA  50      ;uloží hledanou položku na adr. 50
001   04.100       LDC  100     ;ACC:=100 (směrník do tabulky)
002   06.099   ┌─→STA  99      ;uloží směrník do tabulky
003   19.099   │   LDAI 99      ;ACC:=první položka z tabulky
004   10.050   │   AEQ  50      ;porovnat s hledanou hodnotou
005   11.011   │ ┌ JF   012     ;položka nalezena ?
006   05.099   │ │ LDA  99      ;ne
007   07.051   │ │ ADD  51      ;posunout směrník na další položku
008   10.098   │ │ AEQ  98      ;test na konec paměti
009   11.011   │ ┌ JF   010     ;chyba - konec paměti (128 buněk)
010   09.002   │ └ JMP  002     ;jinak zkusit další položku
011   01.000   └─→HALT          ;STOP při chybě
012     ¦        └─→       ;adresa nalezené položky je
                         ; v paměti na adrese 99

050   00.xxx             ;hledaný vzor
051   00.001             ;délka položky v tabulce
       ¦
098   00.127             ;omezovač prohledávání
099   00.xxx             ;směrník do tabulky
100   00.aaa             ;1.položka tabulky
101   00.bbb             ;2.položka tabulky
          ¦                    ¦
                           atd.
```

JMP *adresa*

Skok
(Jump)

kódování: **09.***aaa*

Instrukce způsobí nepodmíněný skok na danou adresu.

Příklad:

```
adr    kód        instrukce     komentář
-------------------------------------------------------------
000    04.000      LDC  000     ;naplní ACC:=0
001    07.100   ┌► ADD  100     ;upraví obsah ACC:=ACC+001
002    11.004   │  JF   004
003    09.001   └─ JMP  001     ;cyklus výpočtu
004    01.000   └─►HALT         ;stop při přeplnění střadače
 |
100    00.001
```

V příkladu je naprogramován cyklus, který neustále zvětšuje obsah střadače o jedničku. Přeplnění střadače lze zjistit podle příznaku F=1. Po přeplnění střadače se program zastaví na instrukci HALT.

?7 **Jaký bude obsah střadače v okamžiku, kdy se program zastaví na instrukci HALT, a proč ?**

JMPI *adresa*

Nepřímý skok
(Jump Indirect)

kódování: **21.*aaa***

Nepodmíněný nepřímý skok. Cílová adresa skoku je uložena v operační paměti na dané adrese.

Příklad:

```
adr    kód        instrukce     komentář
000    06.010      STA  010     ;uloží obsah ACC=xxx na adresu 010
001    21.010     JMPI 010     ;nepřímý skok na adresu xxx
 ¦                  ¦
010    00.xxx      ¦
 ¦                  ¦
xxx                 └→          ;cíl skoku
```

Nepřímý skok lze výhodně použít například pro větvení programu podle vypočtené hodnoty střadače nebo podle stavu vstupních linek mikropočítače a podobně.

28 **Popište funkci takto zapsané instrukce:**

```
adr    kód        instrukce     komentář
003    21.003     JMPI 003     ;  ?
```

JT *adresa*

Skok podle linky T0

(Jump if Test Pin)

kódování: 25.*aaa*

Podmíněný skok na danou adresu. Je-li hodnota vstupní linky T0 rovna 1, potom se provede skok. Jinak se pokračuje v sekvenčním vykonávání instrukcí, tj. provede se další instrukce zapsaná v programu. Hodnota příznaku F ani obsah střadače se při tom nemění.

Příklad: Měření doby trvání pulzu na vstupní lince T0.

```
adr  kód       instrukce    komentář
000  04.000      LDC  000    ;počáteční hodnota střadače ACC=0
001  25.001    ┌ JT   001    ;čekat dokud není T0=0
002  03.002    └→NOP  002    ;prodleva  2*1,28 milisec.
003  07.006      ADD  006    ;zvětšit obsah střadače
004  25.007    ┌ JT   007    ;konec pulzu ?
005            └ JMP  002    ;ne, měřit dále
006  00.001
007    ¦         └→ ;pulz trval dobu τ = ACC * 2,56 [ms]
```

Nepřesnost měření je dána dobou trvání instrukcí ADD, JT a JMP. Provedení každé z těchto instrukcí trvá zhruba 200 až 400 μs.

Jedna a táž instrukce může mít v různých případech proměnnou délku trvání. To je způsobeno vnitřním zapojením mikropočítače PETR. Konkrétně cyklickou obsluhou klávesnice a displeje, která probíhá asynchronně s činností procesoru a může již rozpracovanou instrukci přerušit a pozdržet.

KEY

Čtení klávesnice
(Keyboard Read)

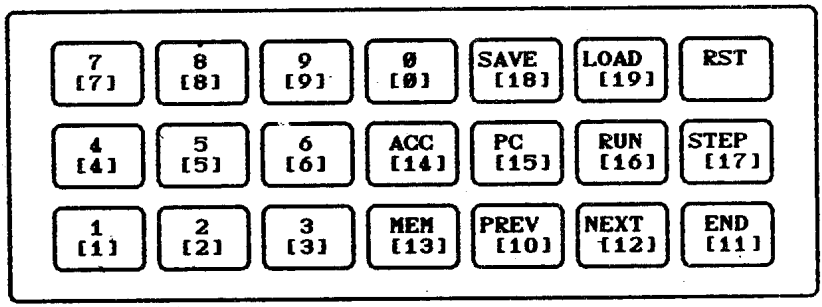
kódování: 26.000

funkce: ACC := stav klávesnice

Instrukce přečte stav klávesnice a uloží ho do střadače. Je-li stisknuto některé tlačítko, uloží se do střadače jeho kód. Jinak se do střadače zapíše hodnota 128. Stisk tlačítka se detekuje právě jednou při každém zmáčknutí, klávesnice tedy nemá funkci "autorepeat" (dlouhodobé stisknutí tlačítka není tedy ekvivalentní řadě opakovaných stisků).

Při čtení klávesnice se provádí automaticky filtrace možného chvění při stisku a puštění tlačítka. Výše uvedeným způsobem nelze přečíst klávesu [END], protože ta způsobí zastavení běžícího programu.

Jednotlivá tlačítka jsou kódována takto:



Obr. 5 Klávesnice stavebnice PETR

Tlačítko (RESET) je vyvedeno samostatně a přímo ovládá stejnojmenný signál na desce mikropočítače PETR. Protože není zapojeno do matice spolu s ostatními tlačítky, nelze jej tedy číst programem.

Příklad: Program pro zobrazování kódu jednotlivých tlačítek na displeji může mít např. tento tvar:

```
adr   kód          instrukce     komentář
000   26.000   ┌──→KEY           ;přečíst stav klávesnice do ACC
001   18.018   │   AEQ   018     ;je stisknuté tlačítko?
002   11.000   ├── JF    000     ;ne, číst znovu
003   02.000   │   DISP  000     ;ano, zobrazit až do
004   09.000   └── JMP   000     ; stisknutí dalšího tlačítka
 |
018   08.128       ;konstanta pro test stavu klávesnice
```

Úkol: Upravte program tak, aby se po stisknutí tlačítek (0), (1), (2), nebo (3) zastavil !

LDA *adresa*

Naplnění střadače přímo

(Load Accumulator Direct)

kódování: 05.*aaa*

funkce: ACC := MEM

Naplní střadač ACC daty, umístěnými na dané adrese paměti. Data musí být v paměťové buňce uložena v její adresní/datové části.

Je-li adresa z intervalu 128 až 255, čtou se data z přídavné paměti (obvodu 8155). Není-li vnější paměť osazena, hlásí se chyba E 002 .

Příklad:

```
adr   kód       instrukce     komentář
000   05.100    LDA  100      ;naplní střadač obsahem adresy 100
                               ;po provedení bude ACC:=123
001   05.130    LDA  130      ;není-li osazena vnější paměť 8155,
                               ; ohlásí se při zpracování této
 ¦                             ; instrukce chyba E 002 - pokus
                               ; o čtení z paměti mimo rozsah
                               ; (0-127), jinak bude ACC:=321
100   00.123    ; data
 ¦
130   00.321    ; data
```

LDAI *adresa*

Naplnění střadače nepřímo

(Load Accumulator Indirect)

kódování: 19.*aaa*

funkce: ACC := @MEM

Naplní střadač daty, jež jsou uložena v paměťové buňce jejíž adresa se přečte z adresy dané operandem *aaa* instrukce LDAI.

Je-li adresa z intervalu 128 až 255, čtou se data z přídavné paměti (obvodu 8155). Není-li vnější paměť osazena, hlásí se chyba E 002 .

Příklad:

```
adr    kód        instrukce     komentář
000    19.100     LDAI  100     ;naplní střadač z adresy 110
 ¦                               ;po provedení bude ACC:=234
100    00.110
 ¦
110    00.234
```

Jaký výsledek bude mít instrukce LDAI ležící na adrese dané operandem, tedy například:

```
adr        kód        instrukce
001        19.001     LDAI  001     ; ???
```

LDC konstanta

Naplnění střadače konstantou
(Load Accumulator Immediate)

kódování: 04.*ccc*

funkce: ACC := CONST

Naplní střadač konstantou, která je uvedena jako operand *ccc* instrukce. Tato konstanta musí být v rozsahu 0 až 255.

Příklad:

adr	kód	instrukce		komentář
000	04.000	LDC	000	;vynulování střadače

Příklad: Výpis zprávy "Stop" doprostřed displeje:

adr	kód	instrukce		komentář
000	04.174	LDC	174	; "S" .. 4+8+2+128+32=174
001	02.005	DISP	5	
002	04.106	LDC	106	; "t" .. 8+64+2+32=106
003	02.004	DISP	4	
004	04.226	LDC	226	; "o" .. 2+64+128+32=226
005	02.003	DISP	3	
006	04.094	LDC	094	; "p" .. 4+16+8+2+64=94
007	02.002	DISP	2	
008	09.008	JMP	008	;program nemůže končit instrukcí ;HALT, protože by se nápis "Stop" ;vzápětí přepsal zprávou ;o zastavení programu

Pozn.: Původní obsah levého a pravého místa displeje se nemění.

NOP *parametr*

Prázdná instrukce / prodleva

(No Operation / Delay)

kódování: **03.***ddd*

funkce:

Pozastavení výpočtu na dobu danou parametrem. Prodleva se měří v jednotkách 1.28 milisekundy, hodnota parametru musí být v rozmezí 0 až 255. První "tik" hodin při čekání může mít kratší délku. Proto doba prodlevy τ leží v intervalu

$$1{,}28 * (p-1) < \tau < 1{,}28 * p \quad [ms]$$

pro hodnotu parametru p=1 až 255. Pro p=0 je prodleva zhruba 0,2 až 0,4 [ms].

Příklad: Podprogram pro prodlevu cca 1 sekundy lze vytvořit např. takto:

```
adr   kód       instrukce     komentář
100   03.255  T1: NOP  255
101   03.255      NOP  255
102   03.255      NOP  255
103   03.016      NOP  16    ;čekat 781 * 1,28 = 999,68 [ms]
104   24.000      RET        ;návrat z podprogramu
```

Podprogramy bývá zvykem (obdobně jako cílová místa skoků) označovat návěštím. V našem případě je návěští, nebo též jméno podprogramu, T1. Takový podprogram se pak volá instrukcí CALL T1 (= CALL 100), jejíž kód by byl 23.100 .

OR *adresa*

Logický součet obsahu střadače s obsahem paměti

(Logical Or Accumulator with Memory)

kódování: 22.*aaa*

funkce: ACC := ACC or MEM

Provede logickou operaci or s obsahem střadače a obsahem buňky paměti na dané adrese. Výsledek se uloží do střadače. (Oba operandy se rozvinou v binární soustavě a provede se logická operace or na odpovídajících pozicích. Výsledek se interpretuje opět jako zápis čísla v binární soustavě). Příznak F není ovlivněn.

Příklad:

```
adr   kód       instrukce    komentář
000   04.252    LDC  252   ;naplní ACC:=252, (binárně 11111100)
001   22.100    OR   100   ;log. součet bude  ACC = 11111100
 |                         ;                  MEM = 00000001
100   00.001               ;      výsledek v ACC := 11111101
```

Po provedení instrukce OR bude tedy ve střadači výsledek ACC=253.

Pravdivostní tabulka pro logický součet je:

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

P1IN *parametr*

Čtení z portu P1
(Input from Port P1)

kódování: 16.*ppp*

funkce: ACC := P1

Linky portu P1 lze s jistým omezením používat současně jak jako vstupní, tak jako výstupní (kvazibidirectional port). Omezení spočívá v tom, že před čtením libovolné linky je do ní třeba zapsat 1, jinak by 0 ve výstupní paměti linky stahovala vstupní úroveň vždy na 0. Jednou zapsaná 1 do výstupní paměti se čtením linky nemění a platí do nejbližšího dalšího zápisu nezávisle na čtení téže linky.

Instrukce P1IN čte stav portu P1 do střadače. Je-li parametr instrukce *ppp*=8, přečte se stav všech osmi linek a jako binární číslo se přepíše do střadače. Je-li parametr z intervalu 0 až 7, přečte se pouze stav vybrané linky P10 až P17 a přepíše se do střadače. Ten pak obsahuje pouze hodnotu 0 nebo 1. Pro jinou hodnotu parametru *ppp* se výpočet přeruší na chybě E 004 (operand mimo povolený rozsah).

Obdobná instrukce pro čtení portu P2 neexistuje, protože port P2 je pouze výstupní. Stav linek P20 až P27 nelze tedy programem číst.

Příklad:

```
adr   kód      instrukce    komentář
000   04.253   LDC   253    ;ACC := 1111 1101
001   17.008   P1OUT 008    ;zapíše 11111101 na linky P17 až P10
002   16.008   P1IN  008    ;ACC := xxxx xx0x
                            ;stav linek P17 .. P12 a P10
                            ;  záleží na vnějším zapojení
                            ;na lince P11 se čte 0 nezávisle
                            ;  na vnějším zapojení
003   16.001   P1IN  001    ;čtení linky P11 samostatně↓
                            ;              ACC := 0000 0000
```

P1OUT *parametr*

Zápis na port P1

(Output to Port P1)

kódování: 17.*ppp*

funkce: P1 := ACC

Instrukcí se zapisuje obsah střadače na port P1 takto: je-li parametr instrukce *ppp*=8, zapíše se obsah střadače současně na všech 8 linek portu P1. Přitom na každou linku se zapíše vždy hodnota příslušného bitu binárního rozvoje střadače. Je-li adresní část z intervalu 0 až 7, zapíše se obsah střadače vždy na jedinou vybranou linku P10 až P17. Obsah střadače musí být v tomto případě roven 0 nebo 1, jinak dojde k chybě E 004 a výpočet se zastaví.

Příklad:

```
adr   kód       instrukce    komentář
000   04.016    LDC   016    ;naplní ACC := 00010000
001   17.008    P1OUT 008    ;zapíše vzorek 00010000
                             ;   na linky  P17 až P10
002   04.001    LDC   001    ;ACC := 1
003   17.000    P1OUT 000    ;zápis na linku P10
                             ; nyní bude P1=00010001
```

P2OUT *parametr*

Zápis na port P2

(Output to Port P2)

kódování: 18.*ppp*

funkce: P2 := ACC

Instrukce je obdobná jako P1OUT s tím, že se obsah střadače zapisuje na příslušné linky portu P2.

Příklad: Program pro "putující jedničku na portu P2. Všechny linky jsou nulové s výjimkou jediné, která se v sekundových intervalech cyklicky posunuje počínaje od P20.

```
adr   kód        instrukce   komentář
000   04.001  ┌─→ LDC   001   ;ACC := 0000 0001
001   18.008  │ ┌→ P2OUT   8   ;přepsat vzorek na port P2
002   23.100  │ │  CALL  100   ;čekat 1 sec !
              │ │  ; podprogram T1 viz popis instrukce NOP
003   06.010  │ │  STA   010   ;uložit obsah ACC
004   07.010  │ │  ADD   010   ;ACC:=2*ACC  tím se jednička
              │ │              ;posune o jedno místo doleva
005   11.000  │ └─ JF    000   ;posun z P27 na P20
006   09.001  └─── JMP   001   ;jinak rovnou výpis
 |
010   00.xxx      ; pomocná datová buňka
```

Napište úsek programu pro kopírování obsahu portu P1 na port P2.

RET

Návrat z podprogramu

(Return from Subroutine)

kódování: 24.000

funkce:

Návrat z podprogramu na adresu, která je uložena na vrcholu zásobníku. Po vyzvednutí návratové adresy se hodnota ukazatele zásobníku sníží o 1. Je-li v okamžiku provádění této instrukce zásobník prázdný, dojde k chybě E 006 a výpočet se zastaví.

Příklad:

```
adr   kód       instrukce     komentář
000   03.000    NOP  000      ;prázdná instrukce hlavního progr.
001   23.005    CALL 005      ;vyvolání podprogramu na adr. 005
002   24.000    RET           ;nyní nastane chyba, zásobník je
                              ; prázdný a program se zastaví
 ⋮                 ⋮
005   24.000    RET           ;tento podprogram neudělá nic
                              ; jiného než návrat zpět
                              ; za instrukci CALL
```

?12 Proč se u instrukce RET neudává adresa, ze které má program dále pokračovat ?

STA *adresa*

Uložení obsahu střadače přímo

(Store Accumulator Direct)

kódování: 06.*aaa*

funkce: MEM := ACC

Uloží obsah střadače do paměti na danou adresu. Je-li adresa z intervalu 128 až 255, čtou se data z přídavné paměti (obvodu 8155). Není-li vnější paměť osazena, hlásí se chyba E 002 .

Příklad:

```
adr   kód       instrukce     komentář
000   05.100    LDA  100      ;zkopíruje obsah paměti z adresy
001   06.050    STA  050      ;100 na adresu 050
```

Příklad: Podprogram pro násobení obsahu střadače deseti, výsledek je opět ve střadači. Pokud byl původní obsah střadače větší než 25, dojde při násobení k přetečení. To je signalizováno příznakem F=1 .

```
adr   kód       instrukce     komentář
000   06.020      STA  020     ;uložit ACC=X na adresu 20
001   07.020      ADD  020     ;ACC := 2*X
002   11.012    ┌ JF   012     ;chyba ?
003   06.021    │ STA  021     ;ne, uložit ACC
004   07.021    │ ADD  021     ;ACC := 4*X
005   11.012    ├ JF   012     ;chyba ?
006   07 020    │ ADD  020     ;ACC := 4*X+X=5*X
007   11.012    ├ JF   012     ;chyba ?
008   06.021    │ STA  021     ;uložit ACC
009   07.021    │ ADD  021     ;ACC := 10*X
010   11.012    ├ JF   012     ;chyba ?
011   01.000    │ HALT         ;ne, v pořádku
012   01.000    └→HALT         ;chyba přetečení
  ⋮
020   00.xxx      ; pomocná datová buňka
021   00.xxx      ; pomocná datová buňka
```

STAI *adresa*

Uložení obsahu střadače nepřímo

(Store Accumulator Indirect)

kódování: 20.*aaa*

funkce: @MEM := ACC

Uloží obsah střadače do paměťové buňky, jejíž adresa je uložena v buňce adresované instrukcí STAI.

Je-li adresa z intervalu 128 až 255, čtou se data z přídavné paměti (obvodu 8155). Není-li vnější paměť osazena, hlásí se chyba E 002

Příklad:

```
adr   kód       instrukce     komentář
000   05.033    LDA  033      ;naplní ACC:=111
001   20.033    STAI 033      ;na adr. 111 uloží obsah střadače
 ⋮
033   00.111
 ⋮
111   00.111                  ;tedy  @033:=111
```

Poznámka:

Instrukce STAI a LDAI umožňují nepřímé adresování paměti a hodí se proto zvláště pro práci s tabulkami.

SUB *adresa*

Odečtení obsahu paměti od střadače

(Subtract Memory from Accumulator)

kódování: **08.***aaa*

funkce: ACC := ACC - MEM

Odečtení dat uložených v paměti na uvedené adrese od střadače. Je-li výsledek záporný, uloží se do střadače zvětšený o 256 a nastaví se příznak F=1. Jinak je F=0 .

Příklad:

```
adr   kód       instrukce     komentář
000   04.011    LDC  011      ;ACC:=11
001   08.010    SUB  010      ;ACC:=11-6   ,   F:=0
002   08.010    SUB  010      ;ACC:=5-6
                               ;výsledek bude ACC:=255 (tj.-1+256)
010   00.006                  ;a příznak F:=1
```

Všimněte si, že příznak F zde má význam výpujčky při odčítání a lze jej tedy využít při odčítání takových čísel, jejichž rozdíl by byl záporný.

3. OVLÁDÁNÍ MIKROPOČÍTAČE PETR OBSLUHOU

3.1 Příkazy zadávané obsluhou

Po zapnutí sestavené a oživené stavebnice (podle návodu v kap. 6) je mikropočítač PETR připraven k činnosti. Obsluha zadává jednotlivé příkazy stisknutím příslušných tlačítek na fóliové klávesnici. Napovědné zprávy o stavu mikropočítače nebo o případných chybách se pak zobrazují na segmentovém displeji.

Stav, ve kterém lze zadat příkaz, je indikován nápovědným písmenem C v levém krajním místě displeje 

Obsluha může nyní prostřednictvím příkazů:

(MEM) prohlížet a měnit obsah paměti mikropočítače,

(ACC), (PC) prohlížet a měnit stav procesoru: střadač ACC, čítač adres PC, příznak F a ukazovátko do zásobníku SP; nelze měnit obsah zásobníku,

(RUN) spustit program od libovolné adresy,

(STEP) krokovat program od libovolné adresy, tj. nechat provést jen jednu zvolenou instrukci a

(LOAD), (SAVE) použít kazetový magnetofon pro zapsání programu na kazetu nebo naopak pro načtení programu do paměti.

Při provádění zvoleného příkazu je na displeji zobrazena odpovídající nápovědná zpráva. Výjimkou jsou pouze oba příkazy pro práci s kazetovým magnetofonem, při kterých je displej zhasnutý.

Modifikační příkazy (pro prohlížení a změnu obsahu paměti, ACC, PC, F a SP) zobrazí na displeji současný stav a očekávají, že obsluha zadá novou hodnotu. Nová hodnota se zadává jako posloupnost číslic, zadávání se ukončí jedním z omezovačů (NEXT), (PREV) nebo (END). Nechcete-li současnou hodnotu měnit, zadejte pouze omezovač. Během vkládání nové hodnoty ukazuje desetinná tečka, kolik číslic se očekává. Zadáváte-li více cifer, předchozí zadané číslice se posunují doleva a nejstarší cifry se ztrácejí. Zadáte-li méně číslic než se očekává, doplní se číslo zleva nulami. Činnost všech modifikačních příkazů lze v libovolném okamžiku ukončit tlačítkem (END).

Volba ostatních příkazů se provádí pouze stisknutím odpovídajících tlačítek. Dále je uveden abecedně seřazený podrobný popis jednotlivých příkazů. U každého příkazu je kromě popisu uvedeno i přehledné grafické vyjádření jeho funkce.

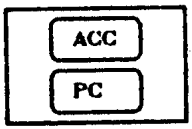
Důležité

Všechna čísla zadávaná i vypisovaná jsou čísla v dekadické soustavě ! Uvnitř počítače PETR se skládají vždy do jednoho osmibitového bytu jako celé číslo bez znaménka. Proto musí být čísla vždy v rozsahu (0 až 255). Mají-li vkládané hodnoty význam instrukcí, pamatujte, že některé instrukce požadují parametry v rozsahu (0 až 1) nebo (0 až 8).

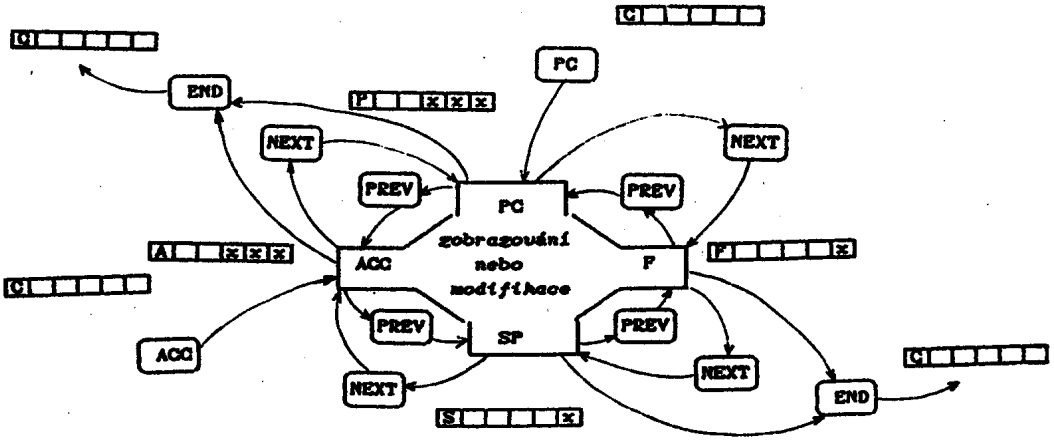
Počáteční nastavení mikropočítače

Po připojení napájení se mikropočítač PETR uvede do počátečního stavu. Na displeji svítí a počítač očekává zásah obsluhy.

Do počátečního stavu lze mikropočítač PETR uvést také kdykoli během činnosti stisknutého tlačítka (RESET) . **Přitom se ale vždy vymaže program uložený v mikropočítači PETR !**



Prohlížení a změna stavu procesoru

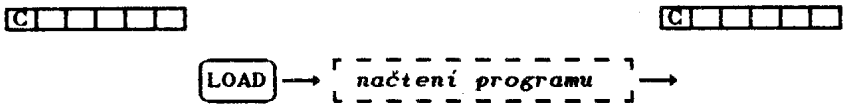


Obsluze jsou dostupné čtyři části procesoru: střadač ACC, čítač instrukcí PC, příznak F a ukazatel zásobníku SP. Tyto čtyři části jsou z hlediska ovládání uspořádány do kruhu, po němž se obsluha posunuje tlačítky (NEXT) a (PREV). V každém okamžiku se pracuje s částí procesoru, znak v levém krajním místě displeje pak napovídá, se kterou. Příkaz lze v libovolném okamžiku ukončit klávesou (END).

Protože se nejčastěji prohlíží obsah střadače a čítače instrukcí, jsou do "modifikačního kruhu" dva vchody: příkaz (ACC) začne zobrazením a modifikací střadače, příkaz (PC) začíná čítačem instrukcí. Příznak F a ukazatel zásobníku jsou dostupné oběma příkazy a opakovaným stisknutím (NEXT) nebo (PREV) podle uvedeného diagramu.

LOAD

Načtení programu z magnetofonové kazety

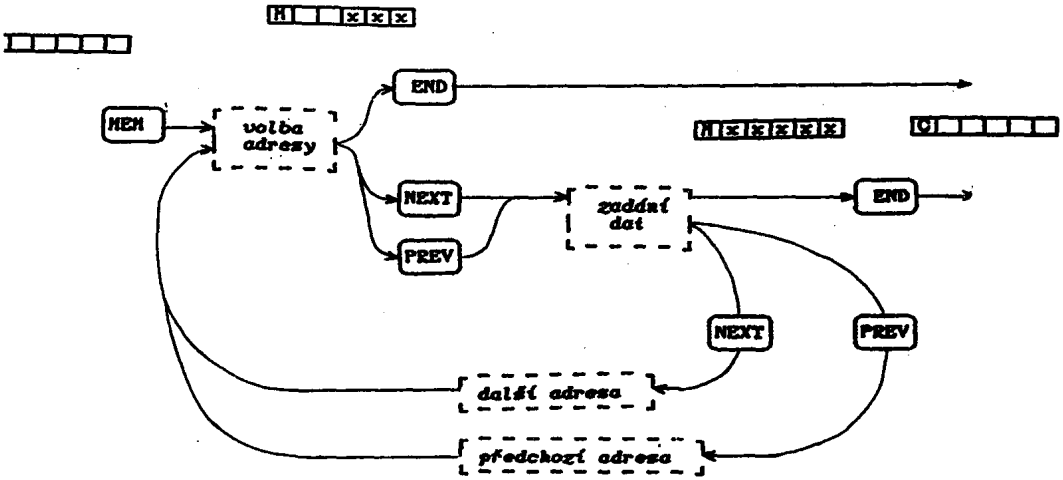


Příkaz LOAD je určen pro čtení programu z kazetového magnetofonu. Přečte jednu magnetofonovou nahrávku a její obsah uloží do paměti počítače. Čtecí program se sám synchronizuje na počátku nahrávky a automaticky se přizpůsobí i polaritě magnetofonu. Nesouhlasí-li kontrolní součet, jímž je každá nahrávka zajištěna, ohlásí se chyba E 007 . Čtení nahrávky trvá několik sekund a během čtení dat z magnetofonu je displej zhasnutý.

Záznam na magnetofoném pásku musí být pořízen buď ukládacím příkazem SAVE mikropočítače PETR, nebo speciálním záznamovým programem mikropočítače IQ 151 (viz kap. 5). Program je na magnetofonovém pásku uložen v jednom bloku, který má uvodní synchronizační úsek, datovou část a kontrolní součet. Bloky nemají hlavičky ani jiné identifikační značky a nemohou být tedy automaticky vyhledávány. Před čtením příkazu LOAD je třeba připojit magnetofon a přetočit kazetu kamkoli před čtený blok. Při ukládání více programů je užitečné namluvit před jednotlivé bloky jejich označení a tím usnadnit zpětné vyhledávání. Jedenblok obsahuje vždy obsah celé paměti mikropočítače PETR.

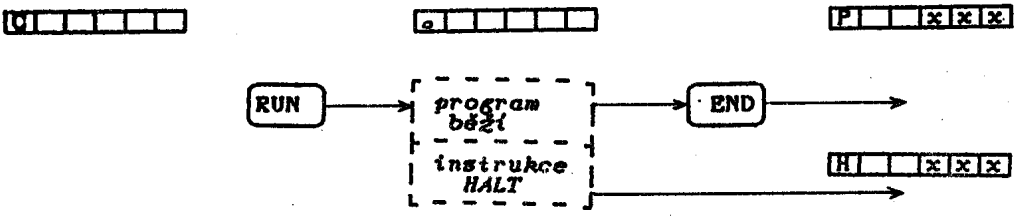
MEM

Prohlížení a změna obsahu paměti



Po vyvolání příkazu se nejprve musí zadat adresa místa v paměti, které chceme prohlížet nebo měnit. Zobrazí se implicitní adresa - poslední použitá adresa v předchozím příkazu MEM . Tuto adresu můžeme ponechat, nebo zadat jinou. Po stisku omezovače (NEXT) se zobrazí obsah buňky paměti na zvolené adrese. Ten můžeme ponechat anebo změnit.
Tlačítky (NEXT) a (PREV) pak lze v paměti krokovat směrem k vyšším nebo nižším adresám, omezovačem (END) se příkaz ukončí.

Spuštění programu

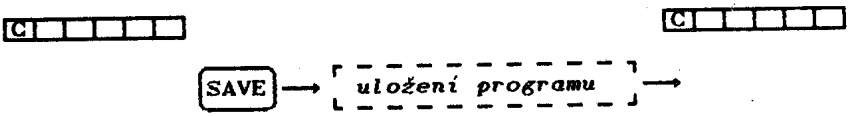


Příkazem (RUN) se spustí program uložený v paměti počítače, počínaje adresou uloženou v čítači instrukcí PC . Běžící program je možno kdykoli zastavit tlačítkem (END) . Tím program přejde do režimu krokování (viz příkaz (STEP)). Bezprostředně po spuštění programu příkazem (RUN) se v levém krajním místě displeje rozsvítí čtvereček jako příznak, že program běží. Tato nápovědná zpráva je na displeji zobrazena dokud ji program sám nepřepíše, nebo dokud ji nepřepíše zpráva o zastavení programu.

?13 Jak lze spustit program z konkrétní adresy, řekněme 123 ? Lze program spustit od adresy 300 ?

SAVE

Uložení programu na magnetofonovou kazetu

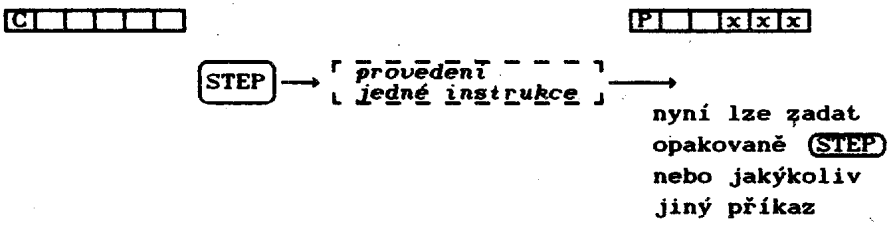


Příkazem SAVE se obsah celé paměti počítače (a tedy i program) zapíše na kazetový magnetofon. Záznam lze přečíst zpět do počítače příkazem LOAD. Během nahrávání na magnetofon je displej zhasnutý. Délka nahrávky jsou necelé 3 sekundy.

Způsob záznamu dat na magnetofonovou kazetu je u stavebnice PETR stejný jako u školního mikropočítače IQ-151. Na magnetofonové kazetě je tedy možno přenášet data mezi oběma počítači.

Program je na magnetofonovém pásku uložen v jednom bloku, který má úvodní synchronizační úsek, datovou část a kontrolní součet. Bloky nemají hlavičky ani jiné identifikační značky a nemohou být tedy automaticky vyhledávány. Před zadáním příkazu je třeba připojit magnetofon a přetočit kazetu před volné místo k uložení záznamového bloku. Při ukládání více programů je užitečné namluvit před jednotlivé bloky jejich označení a tím usnadnit jejich zpětné vyhledávání. Jeden blok obsahuje vždy obsah celé paměti mikropočítače PETR.

Krokování programu



Tlačítkem STEP programátor krokuje svůj program. Po každém kroku se na displeji zobrazí hodnota čítače instrukcí (tj. adresa následující instrukce). Dalším stiskem STEP se pokračuje v krokování, stisk jiného tlačítka krokování ukončí a způsobí přechod do základní příkazové smyčky, v níž si můžete podrobněji prohlédnout stav výpočtu.

Obdobně jako u příkazu RUN je i u příkazu STEP adresa prvního kroku dána čítačem adres PC. Ten je možno předem nastavit stejnojmenným příkazem PC na adresu, odkud zamýšlíme program krokovat.

3.2 Zpracování chyb

Při práci s mikropočítačem PETR mohou nastat chyby dvojího druhu. Jednak jsou to chyby způsobené nesprávnou obsluhou, jednak chypy vzniklé v průběhu zpracování programu. (Druhý typ chyb se běžně označuje jako "Run-Time Errors" - tedy chyby vzniklé v době běhu programu).

Pokud je to možné, detekuje řídicí program mikropočítače chyby v okamžiku jejich vzniku a nikoliv až v době, kdy by se projevily. Protože však bylo nutno řídicí program zkrátit, vyhodnocují se pouze nejdůležitější chyby. V přiloženém výpisu řídicího programu může pozorný čtenář dohledat některé potlačené testy chybových stavů.

V okamžiku, kdy je zjištěna chyba, systém ukončí dosavadní činnost (například přeruší provádění programu) a na displeji zobrazí zprávu [E][ ][ ][0][0][x], kde x je číslo chyby. Číslo chyby charakterizuje prohřešek, kterého se obsluha resp. programátor dopustil.

Číselník chyb:

1 Pokus zapsat do paměti hodnotu, kterou v ní nelze zobrazit (číslo větší než 255), nebo pokus zadat adresu větší než 255.

2 Adresa je mimo rozsah osazené paměti.

3 Neznámá instrukce - pokus interpretovat data jako instrukci (program nejspíše "zabloudil").

4 Operand mimo povolený rozsah (0..1, 0..8 apod.).

5 Přeplnění zásobníku návratových adres (při CALL).

6 Prázdný zásobník návratových adres (při RET).

7 Chyba čtení z magnetofonu - nesouhlasí kontrolní součet.

Stiskem tlačítka (END) odstraníte chybové hlášení a mikropočítač PETR přejde do základní příkazové smyčky. Došlo-li k chybě v běžícím programu, čítač instrukcí PC obsahuje adresu instrukce, která chybu způsobila.

Příklad: Odstranění zprávy o chybě a příčiny chyby [E | | 0 0 6]

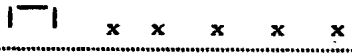
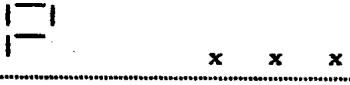
(prázdný zásobník při RET, čítač adres PC ukazuje na instrukci RET, při které došlo k chybě)

END → *nyní je nutno opravit program tak, aby každé instrukci RET předcházela instrukce volání podprogramu CALL*

214 Může dojít k chybě E 006 (prázdný zásobník) při zpracování tohoto programu ?

```
adr   kód       instrukce     komentář
000   23.010    CALL  010     ;volání podprogramu
001   09.000    JMP   000     ;stále dokola
 ⋮                  ⋮
010   04.001    LDC   001     ;podprogram pro výpis vzorku
011   17.008    P1OUT   8     ; 0000 0001 → port P1
012   24.000    RET           ;konec podprogramu
```

3.3 Zprávy zobrazované na displeji

tvar	význam
	základní zpráva mikropočítač PETR je v základní příkazové smyčce
	zadávání adresy paměti
	zadávání obsahu paměti
	obsah čítače adres PC
	obsah střadače ACC
	hodnota ukazovátka do zásobníku SP
	hodnota příznaku F
	adresa instrukce HALT na které se program zastavil
	zpráva o chybě
	příznak běžícího programu

4. POPIS IMPLEMENTACE MIKROPOČÍTAČE PETR

Tato kapitola je určena těm zkušenějším zájemcům, kteří chtějí využít stavebnici jako universální systém s jednočipovým mikropočítačem řady 8048. Takové využití, včetně úprav obvodového zapojení a modifikace řídicího programu, předpokládá hlubší znalosti. Proto i text této kapitoly je stručnější a možná i méně názornější. Je však v příloze doplněn seznamem odkazů na další odbornou literaturu. Vybrány jsou výhradně v češtině resp. slovenštině psané, texty.

4.1 Technické řešení stavebnice

Popis technického řešení stavebnice PETR lze rozdělit na popis mechanického uspořádání a popis obvodového zapojení.

Mechanické uspořádání stavebnice PETR

Stavebnice PETR je navržena pro umístění do skříňky od stolní kalkulačky (výrobek Tesla Bratislava). Skříňka se skládá ze dvou dílů, přičemž na horní díl je připevněna fóliová klávesnice se samolepicí vrstvou, konektor DIN pro připojení magnetofonu a přepínač.

Spodní díl má upravenou zadní stěnu výřezem, kterým vystupuje základní deska kontaktním uživatelským polem. Toto uspořádání umožňuje snadné připojení mikropočítače PETR do vnějšího prostředí.

Oba dva díly jsou spojeny šrouby přes úhelníky, vložené do výřezů horního dílu skříňky.

Obvodové zapojení mikropočítače PETR

Obvodové zapojení mikropočítače (viz schemata zapojení v příloze) lze rozdělit na napájecí část, vlastní zapojení mikropočítače, obvod klávesnice a displeje a obvod styku s magnetofonem.

Napájecí část

Napájecí obvod je tvořen můstkovým usměrňovačem s diodami D9 až D12, filtračními kondenzátory C1 až C4 a integrovaným stabilizátorem IO4 typu MA 7805 (blokovaný proti rozkmitání kondenzátory C15 a C16).

Pro správnou činnost mikropočítače je třeba na vstup zdroje připojit napětí 9 až 12V (stejnosměrné nebo střídavé) s min. výstupním proudem 350 mA.

Technické parametry zdroje:

vstupní napětí	9 až 12V ≃ / 350 mA
výstupní napětí	5V =

Vlastní zapojení mikropočítače

Obvod mikropočítače tvoří dva integrované obvody IO1 a IO2. Obvod IO2 je jednočipový mikropočítač řady 48 řízený krystalem 6 MHz s pamětí programu ROM/EPROM 1 KB na čipu. V paměti programu na čipu mikropočítače je uložen ovládací monitor a interpret příkazů stavebnice PETR.

Brána P1 mikropočítače je vyvedena na kontaktní uživatelské pole jako obousměrná bitově ovládaná sběrnice (označovaná jako port P1). Na kontaktní uživatelské pole jsou také vyvedeny vstupní signály T0 a INT. Vstupní signál T0 je v klidovém stavu na úrovni H (logická 1) a lze jej testovat z programu mikropočítače PETR. Vstupní linka INT je testovatelný vstup vnějšího přerušení. V klidu je na úrovni H. Instrukcí INT lze povolit vnější přerušení, které nastane při přechodu signálu INT na úroveň L (logická 0). S využitím přerušení může program efektivně obsluhovat vnější události.

Propojkou na vstupní lince T1 je možno modifikovat řídicí program stavebnice PETR - viz odst. 4.2. Drátěnou propojkou je vybavena také vstupní linka EA, kterou lze programové vybavení stavebnice úplně vyřadit. Pak je ale nezbytné doplnit vlastní řídicí program a umístit jej do vnější paměti EPROM v přídavné destičce. Vstupní signál Reset (RST) je aktivován automaticky při zapnutí napájecího napětí (vliv C7) nebo pomocí spínacího tranzistoru T15 samostatně vyvedeným tlačítkem foliové klávesnice (RST). Trojice signálů obvodu IO2 RST, SS, a ALE je vyvedena na pájecí body základní desky. Tyto signály slouží pro připojení krokovacího přípravku (ten není součástí stavebnice).

Zapojení klávesnice a displeje

Datovou sběrnicí a signály RD, WR, ALE, P2.2 a P2.3 je k mikropočítači IO2 připojen programovatelný obvod IO1 MHB8155.

Obvod IO1 obsahuje paměť 128 buněk mikropočítače PETR, výstupní 8 bitovou bránu A (označenou pro port P2) vyvedenou na uživatelské kontaktní pole. Dále obsahuje bránu B a C určenou pro obsluhu klávesnice a displeje a čítač, který není pro stavebnici PETR využit (vstup a výstup čítače je pouze vyveden na pájecí body TIN, TOUT).

Brána B ovládá přes spínací tranzistory T1 až T8 svítící segmenty číslicovek LED, brána C přes spínací tranzistory T9 až T14 připojuje napájecí napětí na jednotlivé číslicovky a současně přivádí úroveň L na jeden ze šesti sloupců maticově zapojené klávesnice.

Zbývající dvě tlačítka foliové klávesnice (STEP) a (END) jsou aktivována úrovní na lince P2.4 obvodu IO2. Tři řady klávesnice jsou potom programově testovány na linkách P2.5, P2.6 a P2.7 obvodu IO2.

Připojení kazetového magnetofonu

Základní deska mikropočítače obsahuje obvody pro připojení kazetového magnetofonu. Výstupní signál na lince P1.1 obvodu IO2 je tvarován obvodem R38, C10, R39, R40 a vyveden na dutinku č.1 konektoru DIN. Čtený signál z magnetofonu (dutinka č.3 konektoru DIN) je zesílen operačním zesilovačem IO3, tvarován tranzistorem T16 a snímán na lince P1.0 obvodu IO2. Oba signály slouží alternativně jako programovatelné porty uživatelského pole a pro funkci magnetofonu jsou proto přepínány přepínačem P.

Rozšíření obvodového zapojení stavebnice

Na základní desce mikropočítače je možno osadit konektor K2, na který je vyvedena sběrnice všech signálů vhodných pro připojení libovolného programovatelného obvodu (8155, 8243 ...) a vnější paměti EPROM 2 KB nebo 4 KB.

Stavebnici PETR lze tak modifikovat na specializovaný řídicí mikropočítač se zcela odlišným programem a použitím.

Technické parametry mikropočítače PETR

- procesor řady 48 s řídicím programem na čipu,
- řízení procesoru krystalem 6 MHz,
- paměť mikropočítače PETR 256B = 128 paměťových buněk,
- interface pro kazetový magnetofon,
- stabilizátor napětí,
- 18 signálů uživatelského pole,
- 6 číslicovek displeje,
- foliová klávesnice 21 tlačítek a
- spotřeba max. 5V/320 mA .

4.2 Popis řídicího programu

Dále uvedený popis řídicího programu vysvětluje především vazbu programu na obvodové zapojení stavebnice popsané v odst. 4.1. Je popsána celková koncepce řídicího programu tak, aby se čtenář mohl orientovat v přiloženém výpisu řídicího programu. Listing je podrobně komentován tak, aby bylo možno sdílet jeho již hotové podprogramy i z dodatečně doplněných aplikačních programů. Listing záměrně neobsahuje definice těl využívaných makroinstrukcí jejichž, znalost není nutná a dále výpis ovladače kazetového magnetofonu na fyzické úrovni, který není určen ke zveřejnění.

Úloha řídicího programu

Obvodové zapojení mikropočítače PETR, tak jak je popsané v odst. 4.1, tvoří základ počítače, který uživatel bude programovat. Bývá zvykem, že uživatel nepracuje "na holém hardware", ale že pracuje v nějakém již vytvořeném programovém prostředí, které mu poskytuje základní služby a pomoc při provozování a někdy i při tvorbě programů.

Základní programové vybavení počítače by mělo uživateli umožnit alespoň:

- zavést svůj program do počítače,
- nechat ho provést procesorem,
  a pokud chceme na cílovém počítači programy ladit i
- sledovat (a případně měnit) stav a průběh výpočtu.

Kromě toho by základní programové vybavení mělo poskytovat podprogramy pro ovládání periferií počítače.

Úlohu základního programového vybavení stavebice PETR musí plnit řídicí program vložený do jednočipového mikropočítače 8048. Je však otázkou, jak by měl uvedené funkce realizovat. K tomu lze přistoupit různými způsoby, z nichž každý má své výhody i nevýhody.

Nejjednodušší možností je nechat uživatele programovat přímo mikropočítač 8048. Uživatel by svůj program umístil do pevné paměti EPROM a tu by zasunul do patice ve stavebnici. Po zapnutí systému by mikropočítač začal jeho program vykonávat. V lepším případě je možno uživateli připravit několik základních podprogramů pro obsluhu periferních zařízení (klávesnice, displeje, magnetofonu), případně ještě některé další užitečné rutiny (kolik se jich vejde do vnitřní paměti mikropočítače). Uživatelův program ve vnější paměti by pak mohl tyto připravené podprogramy vyvolávat.

Obě uvedená řešení však umožňují pouze provozovat programy pro mikropočítač, nikoli je vyvíjet. Protože vývoj programů pro mikropočítač není možný bez použití speciálních vývojových pomůcek, které amatérům nejsou běžně dostupné, byla by takto řešená stavebice vhodná spíše pro profesionální využití než jako polytechnická pomůcka. Základní programové vybavení tedy musí programátorovi poskytovat bohatší prostředky, a to zejména pro vývoj a ladění jeho vlastních programů.

Koncepce řídicího programu

Proto byla zvolena tato koncepce programového vybavení mikropočítače PETR: uživatel nebude programovat přímo mikropočítač 8048, ale jistý virtuální jednostřadačový počítač, který bude řídicím programem simulován. Tento počítač (je nazván podle jména stavebnice "PETR") bude mít svůj (virtuální) procesor, paměť i instrukční soubor. Uživatel bude vytvářet programy pro tento virtualní počítač, řídicí program v mikropočítači pak bude uživatelovy programy vykonávat (tj. interpretovat). Vykonávání programu bude možno na pokyn obsluhy pozastavit. Uživatel si bude moci prohlédnout a případně změnit stav výpočtu (obsah střadače, paměti, stav procesoru) a potom třeba ve vykonávání programu pokračovat.

Řídicí program musí kromě interpretace programu zajišťovat i styk s obsluhou, musí přijímat a vykonávat její příkazy: prohlížení a modifikaci obsahu paměti a stavu procesoru počítače "PETR", spuštění nebo krokování programu s možností sledovat a měnit průběh výpočtu a uchovávání uživatelových programů na vnější paměti - kazetovém magnetofonu.

Z toho vyplývá, že řídicí program v mikropočítači musí zajišťovat tyto funkce:

- simulovat počítač "PETR", především interpretovat jeho instrukční soubor,
- komunikovat s obsluhou a vykonávat její příkazy (například spustit program),
- ovládat na fyzické úrovni všechny periferie systému, tj. klávesnici, displej a magnetofon.

Celý řídicí program se přitom musí vejít do paměti ROM v mikropočítači, tedy musí mít délku nejvýše 1024 bytů.

Jako paměť počítače "PETR" se používá vnější datová paměť (RWM v obvodu 8155). Její kapacitou je omezen rozsah paměti počítače "PETR". Ti, kterým základní rozsah paměti nedostačuje, si mohou dalším obvodem 8155 paměť zvětšit na dvojnásobek. Řídicí program (interpret počítače "PETR") při inicializaci testuje, zda je paměť rozšířena, a pokud ano, automaticky ji používá jako druhou polovinu paměti počítače "PETR".

Rozšiřující obvod 8155 se připojuje paralelně s obvodem 8155 na desce mikropočítače PETR s výjimkou vývodu $\overline{CE}$, který je nutno zapojit na vývod P21 mikropočítače 8048. Pro připojení lze využít konektor K2. Podrobněji viz schema zapojení mikropočítače PETR a výpis řídicího programu.

Zvolená koncepce řídicího programu přitom nevylučuje programovat ve stavebnici přímo mikropočítač 8048. Stačí umístit paměť (EPROM 2716) s vlastním řídicím programem do patice rozšiřujícího modulu stavebice. Přitom je nutno propojkou na lince T1 (T1=0) zvolit režim využití externí programové paměti. Tento režim je implementován řídicím programem stavebnice PETR. Po spuštění mikropočítače se inicializuje obvod 8155, část vnitřní datové paměti mikropočítače 8048 a jeho vnitřní čítač/časovač (přesně viz část výpisu řídicího programu uvedenou návěštím RESET:). Poté se provede skok na adresu 400H - tedy na začátek vnější paměti pro program (EPROM 2716). V této paměti musí pak být

umístěn řídicí program, který bude dále stavebnici PETR ovládat jako univerzální systém s mikropočítačem řady 8048. Pro úplnost jsou obdobně jako RESET vyvedeny v tomto režimu do vnější paměti EPROM i vektory pro vnější přerušení (na adresu 403H) a pro vnitřní čítač/časovač (na adresu 407H).

Tento způsob doplnění řídicího programu nelze zaměňovat s přemapováním adres paměti pro program propojkou na vývodu EA (External Access) mikropočítače. Ta způsobí odpojení interního řídicího programu umístěného na čipu mikropočítače 8048 a adresování vnější paměti EPROM 2716 od adresy 0.

Souhrnně je adresování paměťového prostoru znázorněno v následující tabulce:

PROPOJKY		ADRESOVÁNÍ PAMĚTI PRO PROGRAM		
EA	T1	1. KB	2. - 3. KB	4. KB
1	1	0 - 3FFH PETR	-	-
1	0	0 - 3FFH PETR	400H - 0BFFH řídicí program ve vnější paměti EPROM 2716 pozn.: 400H ... RESET 403H ... EXTERNAL INT 407H ... TIMER INT	0C00H-0FFFH lze event. doplnit
0	x	0 - 0FFFH pozn.:	vnější řídicí program 0 ... RESET 3 ... EXTERNAL INT 7 ... TIMER INT	

Poznámka:

V prvním případě lze využít pouze řídicí program PETR na čipu mikropočítače. Ve druhém případě je stavebnice řízena vnějším řídicím programem, který může využívat podprogramy řídicího programu PETR. Naproti tomu ve třetím případě je program PETR na čipu trvale odpojen a stavebnice je řízena výhradně vnějším řídicím programem.

5. DOPLŇKY MIKROPOČÍTAČE PETR

5.1 Popis obvodového zapojení logické sondy

Logická sonda je tvořena vstupním obvodem, obvody vyhodnocující úrovně L a H a obvody indikace. Vstupní obvod je totožný se zapojením logické sondy zveřejněné v [17], která byla konstrukční částí soutěže INTEGRA 1980.

Vstupní obvod je tvořen emitorovým sledovačem s ochranou proti zápornému vstupnímu napětí (dioda D4). Emitorový odpor tvořený trimry P1 a P2 umožňuje nastavit napěťové úrovně pro vyhodnocovací invertory. Tyto invertory spouští monostabilní klopné obvody, sloužící pro prodloužení měřených impulzů na viditelné bliknutí diod LED. Monostabilní obvody tvořené invertorem a klopným obvodem typu D je možno změnit na bistabilní obvody uzemněním vstupu invertoru (uzemění pájecích bodů). Sonda potom dokáže zachytit ojedinělý pulz úrovně L nebo H.

Technické parametry sondy

- indikace L a H pro obvody slučitelné s TTL,
- indikace oblasti X (0,8 ÷ 2,4V) zhasnutím obou diod LED,
- min. šířka vstupního impulzu 20 ns,
- zatížení měřeného místa 0,2 vstupu TTL,
- paměť impulzu L nebo H,
- ochrana proti zápornému vstupnímu napětí,
- napájecí napětí 4,75 ÷ 5,25 V a
- odběr z napájecího zdroje max. 60 mA.

Schema zapojení logické sondy je v příloze.

5.2 Popis obvodového zapojení vnější paměti a expanderu

Na výstup systémové sběrnice K2 lze připojit rozšiřující desku (označenou ***). Ta obsahuje adresní registr I08, vnější paměť programu (obvod I07) a expander I09. Výstupy expanderu jsou včetně napájecího napětí vyvedeny na konektor K3.

Drátěnou propojkou lze volit paměť programu EPROM 2KB nebo 4KB.

Tato deska je určena pro rozšíření stavebnice PETR při částečném využití jeho základního programového vybavení, nebo

pro zcela specifické aplikace při nahrazení řídicího programu stavebnice PETR vlastním aplikačním programem (viz kap. 4). Použití rozšiřující destičky a doplnění řídicího programu včetně ovládání expanderu vyžaduje dobrou znalost jednočipových mikropočítačů řady 48 a přístup k vývojovým prostředkům.

Podrobnější popis přesahuje rámec této příručky, lze však využít doporučenou literaturu [3], [4], [11], [12] nebo [15] a [16].

5.3 Vývojové prostředky pro programování stavebnice PETR

Nutnou (bohužel často zanedbávanou a opomíjenou) podmínkou použitelnosti každého programovatelného systému je dostupnost prostředků pro vytváření programů. Patří sem především editory, překladače, zavaděče a nejrůznější ladicí pomůcky (symbolické ladicí programy, simulátory, emulátory). Jednou ze základních charakteristik systému je, zda umožňuje vyvíjet programy přímo na něm samém, nebo zda je nutno programy pro tento systém připravit někde jinde (na jiném, hostitelském počítači) a do cílového systému je potom přenést.

Stavebnice PETR i přes svůj minimální rozsah umožňuje bez dalších prostředků vytvářet a ladit programy, které bude vykonávat. Nemůže však pochopitelně poskytnout komfort, na jaký jsme zvyklí u počítačů vyšších tříd.

Program se do počítače zavádí z klávesnice přímo ve strojovém kódu virtuálního procesoru. Jednou zavedené programy si však programátor může uchovávat na vnější paměti - kazetovém magnetofonu.

Ladicí možnosti jsou srovnatelné s běžnými ladicími prostředky jiných systémů - programátor může svůj program krokovat a přitom prohlížet a případně měnit stav výpočtu. Je škoda, že se při krokování nezobrazují všechny užitečné informace najednou (např. obsah střadače, příznaky), ale velikost zobrazovače (šestimístný displej) to prostě neumožňuje.

Nejslabším článkem ve vývoji programů je jejich zavádění do paměti ve strojovém kódu (i když nejde o žádné dlouhé programy, jejich velikost je totiž omezena kapacitou paměti počítače 128 buněk). Jednoduchost celé stavebnice neumožňuje jiný způsob programování, řešením však může být použití křížových vývojových prostředků.

Křížové vývojové prostředky

Křížové vývojové prostředky pro stavebnici PETR poskytují možnost programovat místo ve strojovém kódu v jazyku symbolických adres (kdo někdy zkusil obojí, ví, jaký kvalitativní skok to představuje). Vzhledem k omezené velikosti paměti a určení stavebnice není potřebné ani smysluplné vytvářet další vývojové pomůcky (např. překladače z jiných jazyků).

Úlohu hostitelského počítače zde plní běžně dostupný školní mikropočítač IQ-151. Programy se mezi obéma systémy přenášejí na magnetofonových kazetách díky kompatibilitě nahrávání.

Vývojové prostředky na počítači IQ-151 budou pracovat pod jeho operačním systémem AMOS - podrobněji viz [18]. Sestávají ze čtyř programů:

- překladač z jazyka symbolických adres do strojového kódu procesoru PETR,
- zpětný překladač ze strojového kódu do jazyka symbolických adres,
- konverzní program pro vytvoření záznamu na magnetofonu ve formátu PETR,
- konverzní program pro přečtení záznamu ve formátu PETR z magnetofonu.

K editaci zdrojového textu programu se používá editor, jenž je integrální součástí OS AMOS.

Jazyk symbolických adres

Jazyk symbolických adres obsahuje nejnutnější konstrukce pro pohodlné psaní programů ve strojově orientovaném jazyce - možnost symbolických jmen, symbolických názvů instrukcí, návěští a konstant. Jazyk zahrnuje všechny instrukce procesoru PETR a pět pseudoinstrukcí obvyklých snad ve všech asemblerech:

	ORG	x	nastaví hodnotu čítače adres na hodnotu x
	DS	x	vynechá x buněk operační paměti
	DATA	x	umístí konstantu x do paměťové buňky
jm	EQU	x	symbolickému jménu *jm* přiřadí hodnotu x
	END		ukončuje zdrojový text programu

x ... zde označuje výraz, *jm* ... symbolické jméno

V roli operandů ve výrazech mohou stát číselné konstanty (v desítkové, dvojkové i šestnáctkové soustavě), symbolická jména a znakové konstanty (odpovídají znakům zobrazovaným na segmentovém displeji). Jako operátory ve výrazech je možno používat pouze + a - (binární i unární).

Implementace křížových vývojových prostředků

Všechny čtyři programy (asembler, disasembler a oba konverzní programy) jsou napsány v asembleru mikroprocesoru 8080 a implementovány pod operačním systémem AMOS.

Překladač jazyka symbolických adres je realizován jako klasický dvouprůchodový asembler (viz např. [18]). Kromě základní funkce - generování strojového kódu - umí na požádání vypsat do závěru listingu tabulku symbolů (tj. tabulku všech použitých symbolických jmen a jejich hodnot). Z hlediska operačního systému jde o program, který pracuje se třemi soubory: z prvního čte zdrojový text, do druhého zapisuje generovaný kód a do třetího píše listing programu.

Zpětný překladač je velmi jednoduchý. Je jednoprůchodový, před instrukce neumisťuje návěští a negeneruje žádná symbolická jména. Kladem disasembleru je skutečnost, že jeho výstup (text programu v mnemotechnických názvech instrukcí) je bez dalších úprav přeložitelný asemblerem.

Oba konverzní programy jsou velmi krátké programy, které pouze převádějí (kopírují) data ze souboru pod operačním systémem do tvaru přijímaného řídicím programem stavebnice a naopak.

6. OSAZENÍ A OŽIVENÍ STAVEBNICE PETR

Stavebnice PETR představuje poměrně jednoduchý mikropočítačový systém osazený na jednostranném plošném spoji. Přesto by se do stavby mikropočítače PETR neměl pouštět úplný začátečník bez znalostí základních pojmů a základních znalostí o součástkách a jejich značení a když, tak jen v odborných kroužcích pod vedením odborných vedoucích.

Budete-li stavět mikropočítač sami, postupujte dle následujícího návodu a pracujte pomalu a pečlivě.

Stavebnice se skládá ze 4 desek plošných spojů. Odděleny jsou "čárkovaným plošným spojem" a mezera mezi jednotlivými deskami je umožňuje oddělit pilkou na železo.

Základní deska je označena názvem, ostatní menší destičky jsou označeny jednou, dvěma nebo třemi hvězdičkami.

K osazení desek potřebujeme běžné nářadí: mikropájku (u pistolového pájedla si udělejte tenký hrot ze slabšího drátu), pinzetu, štípačky, cínovou pájku, kalafunu, popř. odsávačku a vrtačku. Pájet musíte rychle, čistě a s minimálně nutným množstvím cínové pájky. Plošné spoje jsou poměrně husté a je třeba dávat pozor na zkraty cínovými můstky.

Svoji šikovnost při pájení si nejprve ověřte při stavbě logické sondy.

6.1 Osazení logické sondy

(deska označena ***)

Sonda nám poslouží pro oživení mikropočítače, při ladění programů (kontrola vstupních a výstupních signálů) i pro další práci s logickými obvody.

Osazování začínáme drobnějšími součástkami (odpory, diodami); vývody součástek ohýbáme pinzetou nebo kleštěmi podle roztečí v plošném spoji. Vývody neohýbáme těsně u čel součástek a součástky nedorážíme těsně k desce, ale osadíme s mezerou asi 2 mm (odpory a diody potom nejsou při pájení tolik tepelně namáhány).

Před vlastním osazením součástkami prohlédněte destičku plošných spojů, zda nejsou některé spoje přerušeny, popř. navzájem zkratovány neproleptanými můstky. Případné chyby odstraníme propojením tenkým drátem nebo proškrábnutím.

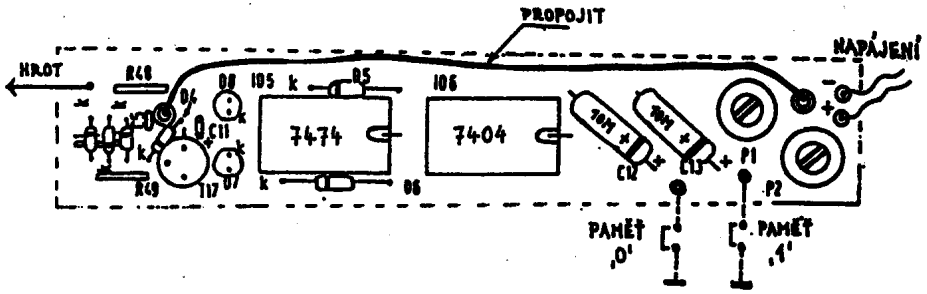
Pracujte pečlivě, výměna součástek a několikanásobné pájení obvykle vede k poškození plošného spoje. Pozor proto na polaritu diod, elektrolytických kondenzátorů a orientaci integrovaných obvodů. Při osazování se neřiďte pouze obrázkem rozložení součástek, ale také schematem zapojení. Nezapomeňte osadit jednu drátěnou propojku.

Pro nastavení sondy potřebujeme pevný zdroj 5V a proměnný zdroj 0 až 5V. Propojíme referenční (-) svorky obou zdrojů. K sondě připojíme napájecí napětí (odběr ze zdroje 20-30 mA) a na hrot sondy připojíme kladné proměnné napětí. V rozsahu proměnného napětí 0+0,8V má svítit dioda D7=úroveň L (nastavíme trimrem P2), v rozsahu 2,4+3,5V svítí dioda D8=úroveň H (nastavíme trimrem P1). Nastavení několikrát zopakujeme, nepřesnost jednoho nebo dvou desetin voltu není pro naše použití nijak kritická.

Nyní ještě zkontrolujeme paměť sondy. U vývodů elektrolytických kondenzátorů C12 a C13 jsou pájecí body, které při spojení se zemí umožní zapamatování jedničkového nebo nulového pulzu.

Spojíme-li se zemí paměť "1" (viz obr. 6) a hrotem se dotkneme napětí např. 5V, rozsvítí se dioda "H" a svítí trvale, i když hrot odpojíme, nebo připojíme na úroveň L. Zcela analogicky vyzkoušíme paměť "0".

Sondu umístíme do vhodné krabičky (např. do pouzdra od kuličkového pera), nebo si krabičku vyrobíme z odřezků cuprextitu a popíšeme. Na napájecí vodiče připojíme nástrčné konektory a vodiče označíme polaritou proti přepólování.

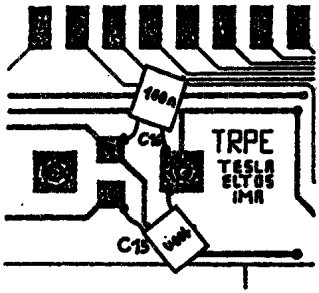


Obr. 6 Osazení logické sondy

6.2 Osazení mikropočítače PETR

U desky mikropočítače pečlivě prohlédneme plošný spoj včetně výstupního pole kontaktů (mezi krajními kontakty ▲ a + by mohl být zkrat, možný zbytek rohové značky plošného spoje). Do pájecích bodů Z1 až Z8 a DIN1 až DIN3 zaťukneme kontakty a dobře zapájíme (obr. 13, str.79). Ostatní kontakty napájíme na pájecí plochy výstupních signálů (viz obr. 8, osazení zdroje).

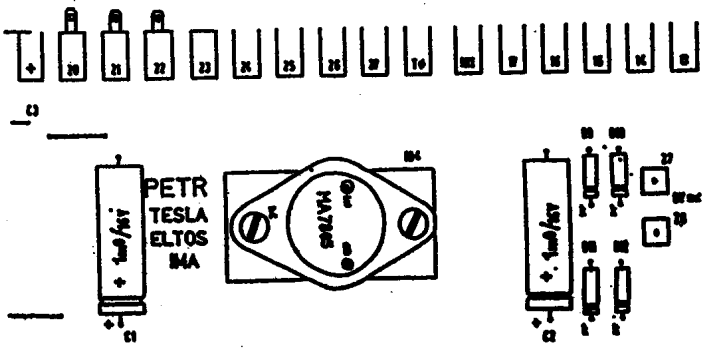
Dále osadíme 11 drátěných propojek a díly zdroje (D9+D12, C1+C4 a I04). Ze strany pájení připájíme na I04 kondenzátory C15, C16 dle obrázku 7.



Obr. 7 Umístění C15 a C16

Na svorky Z7 a Z8 připojíme pomocí nástrčných kontaktů napájecí vodiče, které připojíme na napětí 8V. Jako zdroj použijeme napáječ pro vláčky (můžeme použít zdroj střídavý nebo stejnosměrný, neboť zde nezáleží na polaritě).

Na výstupních svorkách mikropočítače (2 krajní levé a 2 krajní pravé) potom zkontrolujeme napětí 5V. Tyto svorky slouží pro napájení vnější elektroniky a logické sondy.



Obr. 8 Osazení zdroje a kontaktů

6.2.1 Osazení a oživení obvodu displeje

Displej tvořený diodami D13 až D15 je umístěn na destičce označené * .

Na destičce opět zkontrolujeme plošný spoj a osadíme 3 drátěné propojky. Poté osadíme displej D13 až D15 (orientace displeje je označena klíčem - levý dolní roh), viz obr. 9.

Na desku mikropočítače osadíme součástky dle obr. 10, tzn. objímku IO1, odpory R1 až R29 a tranzistory T1 až T14 Objímka IO má orientaci, kterou dodržte.

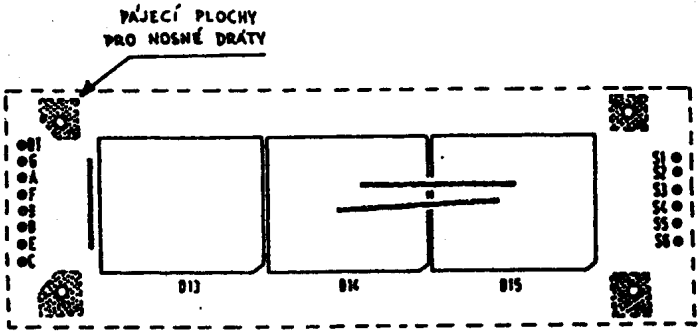
Pozn.: Pájecí body RT, TIN,TOUT se v základní verzi stavebnice PETR nevyužívají.

Nyní k připevnění destičky displeje na základní desku. Displej je se základní deskou spojen 14 vodiči. Vodiče S_1+S_6 slouží pro sepnutí jedné ze šesti číslicových znakovek, vodiče označené DT, G, A, F, B, D, E, C spínají jednotlivé segmenty (viz schema zapojení v příloze).

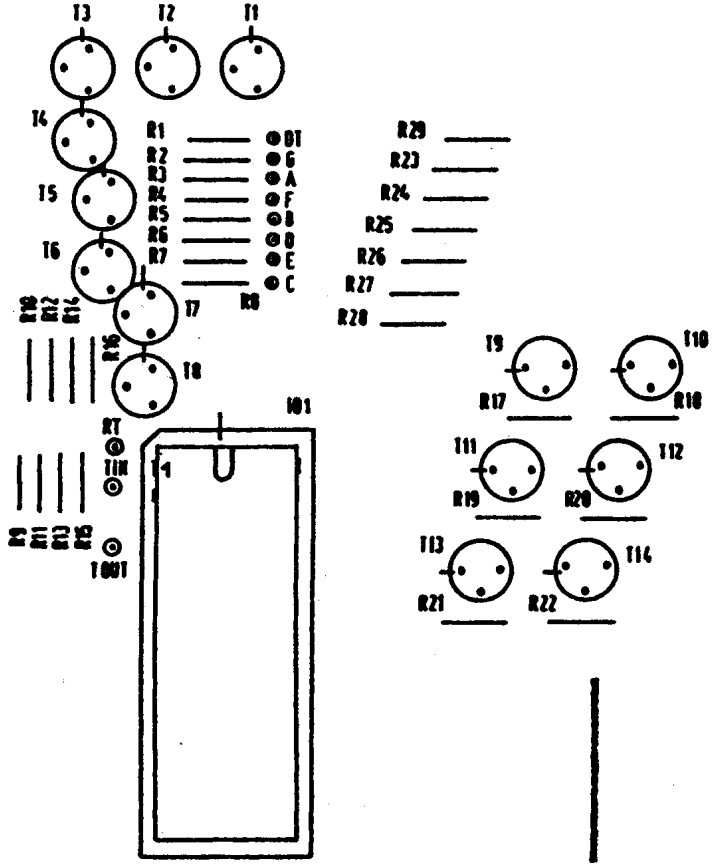
Na destičku připájíme propojovací, asi 7 cm dlouhé vodiče těchto signálů a druhé konce vodičů si odizolujeme pro připojení do základní desky. Na velké pájecí plošky na destičce displeje připájíme 4 silnější měděné dráty dlouhé asi 4 a 5 cm, které poslouží jako nosné sloupky displeje. Před zapojením nosných drátů do základní desky si nejprve vyzkoušíme výšku a sklon destičky displeje nad základní deskou tak, aby při umístění mikropočítače do skříňky byl displej přesně uprostřed okénka (nosné dráty volte raději delší, pro dotvarování polohy displeje).

Nyní připojíme i signálové vodiče displeje. Pájecí otvory destičky displeje i desky mikropočítače si přesně odpovídají. Po připojení displeje si ověříme jeho správnou funkci. K mikropočítači připojíme napájecí napětí a připravíme si dva vodiče připojené na minus pól mikropočítače (oba krajní body kontaktního pole). Nyní první vodič připojíme do kontaktu č.39 objímky IO1 a druhým vodičem se postupně krátce dotkneme vývodů č.29, 30, 31, 32, 33, 34, 35, 36. Postupně se tak rozsvítí segmenty levé krajní znakovky. První vodič nyní připojíme do kontaktu č.38 (a postupně do 37,5,2,1) a manipulaci druhým vodičem opakujeme.

Otestujeme tak všechny segmenty displeje i obvody spínacích tranzistorů.



Obr. 9 Osazení destičky displeje



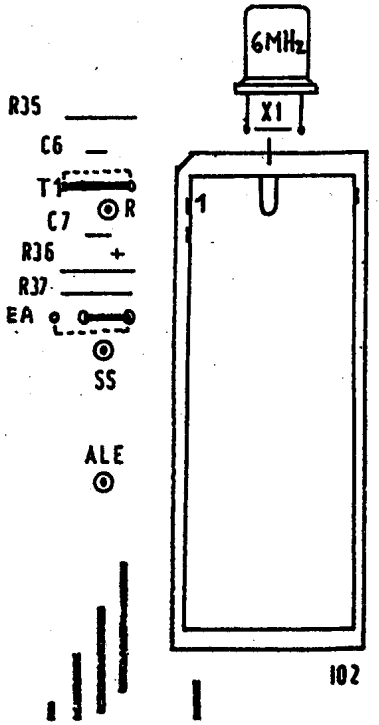
Obr. 10 Osazení spínacích obvodů displeje

Při zjištění jakékoliv chyby se ji pochopitelně pokusíme najít podle schematu obvodového zapojení a odstranit. Jinak dále osazovat desku nemá význam.

6.2.2 Osazení obvodů mikropočítače

Na základní desku osadíme objímku IO2, krystal X1, odpory R35 až R37, kondenzátor C6 a elektrolyt C7 podle obr. 11.

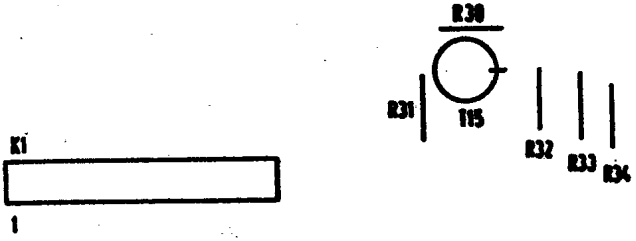
Pájecí body vstupního signálu T1 umožňují tento signál připojit na zem nebo + pól zdroje. Stav tohoto signálu umožňuje modifikaci řídicího programu. U základní verze stavebnice PETR je tento signál připojen na úroveň H podle obr. 11. Obdobný přepínač má i vstupní signál EA. Signál umožňuje zakázat vnitřní program mikropočítače. Pak je nutné připojit destičku s vnější pamětí a vlastním řídicím programem. V základní verzi stavebnice PETR je signál EA připojen na úroveň L (viz obr. 11). Podrobnější popis mapování paměti je uveden v odst. 4.2.



Obr. 11 Osazení obvodů mikropočítače

Pozn.: Pájecí body R, SS, ALE nejsou v základní verzi stavebnice využity, v jiných aplikacích stavebnice mohou sloužit pro připojení krokovacího přípravku.

Nyní již můžeme vyzkoušet mikropočítač v jeho základní funkci. Do objímek opatrně zasuneme oba integrované obvody (pozor na orientaci) a připojíme napájecí napětí. Na levém krajním displeji se objeví nápovědný znak "C". Je-li tomu tak, zbývá osadit obvody klávesnice (konektor K1, R30 až R34, T15), viz obr. 12 a potom obvody pro styk s magnetofonem.



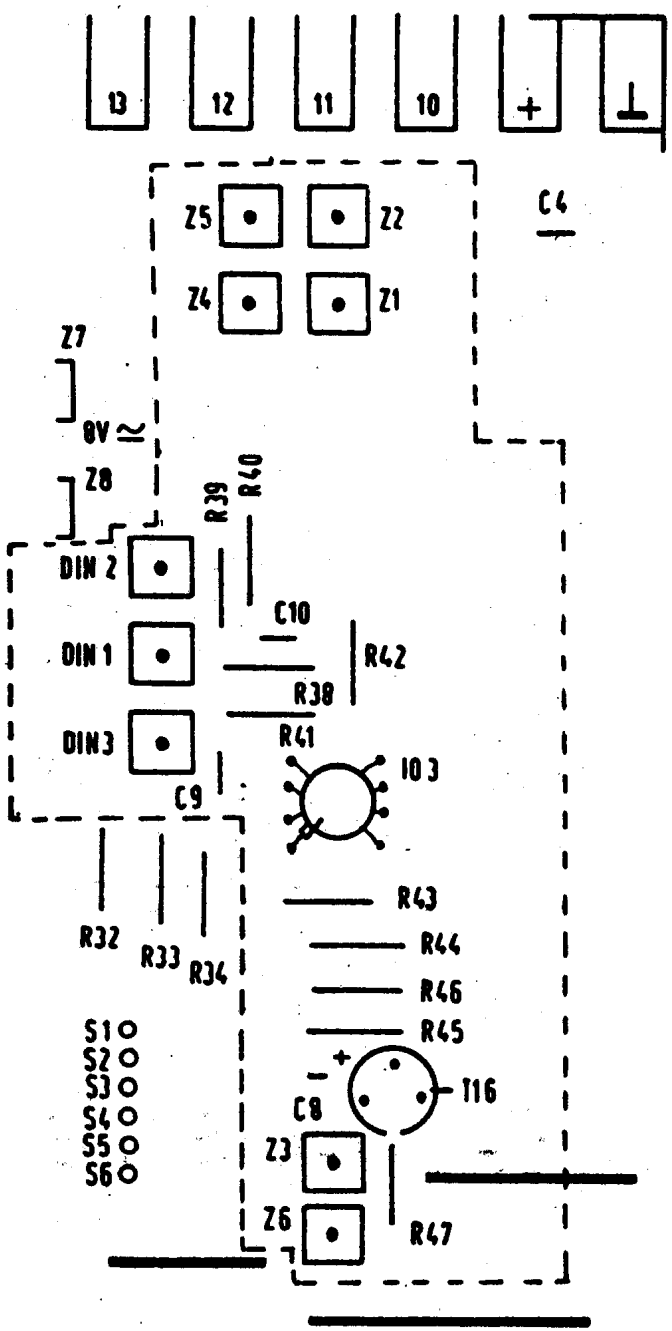
Obr. 12 Obvod klávesnice

6.2.3 Osazení obvodu magnetofonu

Pro nahrávání a přehrávání se využívají 2 linky z kontaktního pole označené 10 a 11. Tyto linky jsou součástí portu P1 a lze je běžně využívat instrukcemi P1IN a P1OUT. Jsou však vedeny přes přepínač P, kterým vždy před použitím magnetofonu tyto linky připojíme k obvodům nahrávání/přehrávání a poté je opět přepojíme zpět k výstupnímu poli kontaktů.

Přepínač je připojen podle schematu vodiči s nástrčnými konektory do kontaktů označených Z1 až Z6 .

Konektor DIN pro připojení nahrávací šňůry magnetofonu s třemi dutinkami je připojen do kontaktů DIN1, DIN2, DIN3 (čísla dutinek konektoru a kontaktů na základní desce si odpovídají).



Obr. 13 Označení obvodů magnetofonu

Součástky osadíme dle obr. 13, pozor na orientaci IO3 (klíč označuje vývod č. 8). Správnost funkce obvodů magnetofonu by bylo sice možné ověřit i samostatně (pomocí osciloskopu a nf generátoru), ale my správnost funkce ověříme až při zkompletování stavebnice přímo s magnetofonem pomocí příkazů SAVE a LOAD.

Mikropočítač je prakticky osazen a zbývá jej umístit do skříňky. Základní desku vložíme do spodního dílu skříňky a připojíme napájecí vodiče do kontaktů Z7 a Z8. Na vrchní díl skříňky připevníme klávesnici (má samolepicí spodek), jejíž plochý kabel provlékneme otvorem vrchního dílu.

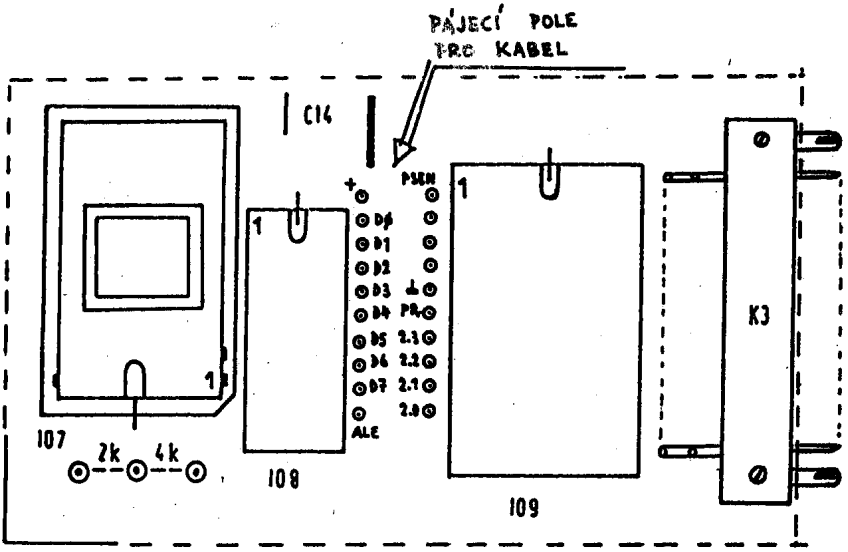
Nyní kablíky od přepínače a konektoru DIN umístěné na vrchním dílu připojíme nástrčnými konektory do odpovídajících kontaktů (je výhodné si vodiče u nástrčných konektorů předem popsat čísly kontaktů) a při téměř uzavřené skříňce zasuneme kabel klávesnice do konektoru K1 (kabel je poměrně krátký a jeho připojení chce trochu trpělivosti).

Skříňku uzavřeme a po připojení napájení nyní můžeme vyzkoušet celou klávesnici s funkcemi jednotlivých příkazů.

6.3 Osazení vnější paměti

Zbyla nám destička (označ. ****) umožňující modifikaci stavebnice. Rozšíření stavebnice o vnější paměť programu a linky expanderu si zároveň vynucuje úpravu (nebo celé přepracování) řídicího programu a jeho umístění do EPROM v pozici IO7 (obr.14). To ovšem předpokládá dobrou znalost jednočipového mikropočítače MHB 8048 (včetně programování v asembleru, možnosti práce na vývojovém systému a emulátoru) a zcela se vymyká z rámce popisu této příručky. Pro informaci lze uvést osazení destičky vnější paměti a expanderu (obr. 14).

Základní desku mikropočítače PETR je pro připojení této destičky nuto osadit konektorem K2. Konektor K2 obsahuje všechny systémové vodiče pro připojení i jiných rozšiřujících obvodů. Programové vybavení základní verze mikropočítače PETR již např. předpokládá možnost připojení druhého obvodu MHB 8155 a tím zdvojnásobení velikosti paměti stavebnice PETR.



Obr. 14 Osazení vnější paměti

Naznačené úpravy řídícího programu a obvodového zapojení jsou určeny jen pro zkušené uživatele. Na druhé straně jim však stavebnice poskytuje základ universálního mikroprocesorového systému s mikropočítačem 8048

Pozn.: Vnější zapojení, která budete mikropočítačem ovládat, stavte pečlivě a nejprve na univerzální desku, kde si ověříte správnost zapojení.

Výstupní pole kontaktů umožňuje připojit *jednu zátěž TTL*. Při připojení vnější aplikace mějte na paměti, že po zapnutí mikropočítače nebo po stisknutí tlačítka (RST) se signály P10 až P17 chovají jako vstupní s vysokou impedancí. Linky portu P1 lze programovat jako vstupní nebo jako výstupní s jistým omezením - viz popis instrukcí P1IN a P1OUT.

Signály P20 až P27 jsou směrovány vždy jako výstupní a tlačítko (RST) neovlivní jejich stav, pouze vypnutí a zapnutí mikropočítače je nastaví na počáteční úroveň L (obvod I02 je nulován pouze náběhem zdroje).

Tyto vlastnosti výstupních a vstupních linek je nutno při návrhu řízeného obvodu respektovat. Pozor na zkraty na těchto linkách ! Zničení integrovaných obvodů je snadné a jejich cena není malá.

Uvažujte i spotřebu přídavné vnější aplikace. Zdroj pro vláčky při napájení mikropočítače a současné třeba logické sondy a složitějšího vnějšího obvodu již nemusí být dostatečně tvrdý a je vhodné si změřit napájecí napětí na svorkách kontaktního pole. Je-li menší než 4,75V nemáme jistotu bezchybného chodu.

Potom je vhodné napájet aplikační zapojení samostatným zdrojem a spojit pouze země mikropočítače a této aplikace.

Stabilizátor napětí stavebnice (IO4) je v optimálním případě schopen dodat proud max. 1A což pro mikropočítač, logickou sondu i aplikaci ve většině případů stačí, ale musíme použít výkonnější zdroj než je napáječ pro vláčky.

7. PŘÍKLADY PROGRAMŮ

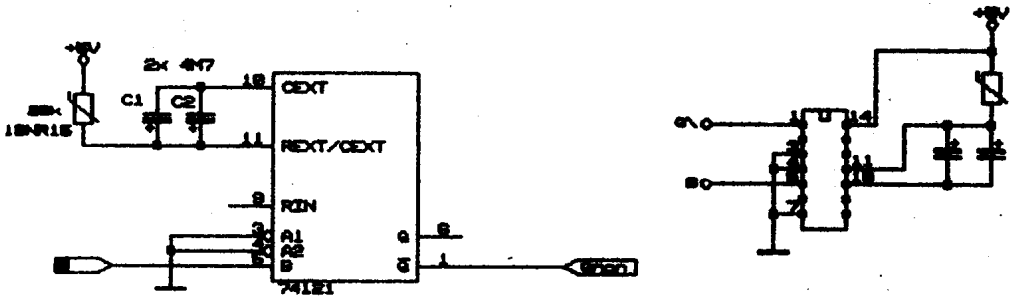
7.1 Digitální teploměr

Námět:

Navrhněte a naprogramujte jednoduchý digitální teploměr na principu teplotně závislého monostabilního klopného obvodu.

Řešení:

Monostabilní klopný obvod (MKO) zapojíme z integrovaného obvodu UCY 74121 a termistoru 13NR15 (20 kΩ) podle následujícího schematu:



Náběžná hrana jednotkového impulsu, vstupujícího do MKO způsobí po určité době stejnou změnu i na výstupu Q (na výstupu $\overline{Q}$ je negace tohoto impulzu).

Doba, za kterou se změní stav na výstupu, je určena hodnotami kapacity a odporu připojených součástek (kondenzátory musí být časově stálé, nejlépe tantalové).

Bude-li se měnit teplota termistoru, bude se měnit i jeho odpor, a tím i doba zpoždění výstupního impulzu za vstupním.

Změřením tohoto zpoždění a vhodnou úpravou (přepočtem) lze tedy určit teplotu termistoru.

Dále je uvedena jedna z možných verzí programu. Pro výstup jednotkového pulzu z počítače do MKO se využívá linka P14, časové zpoždění se měří na lince P15. Aby bylo možné číst informaci z linky P15, musí se aktivovat logickou jedničkou (viz popis instrukce P1IN).

```
adr   kód      instrukce     komentář
-------------------------------------------------------
000   04.000   LDC    000   ;nulování střadače
001   17.004   P1OUT   4    ;nulování 4.bitu P1
002   03.000   NOP    000   ;zpomalení reakce MKO na změnu teploty
003   04.001   LDC    001   ;1 do střadače
004   17.004   P1OUT   4    ;změna z 0 na 1 - čelo impulzu
005   17.005   P1OUT   5    ;aktivování 5.bitu P1
006   04.000   LDC    000   ;nulování střadače
007   17.004   P1OUT   4    ;ukončení impulzu na 4.bitu - týl
```

Měření doby zpoždění výstupního signálu za vstupním:

Ke střadači budeme přičítat jedničku tak dlouho, dokud se na lince P15 neobjeví úroveň H (nikoliv L, protože je použit z MKO výstup $\overline{Q}$).

```
adr   kód      instrukce     komentář
-------------------------------------------------------
008   07.100   ADD    100   ;přičtení 1 ke střadači
009   06.101   STA    101   ;uschování střadače na 101
010   16.005   P1IN     5   ;načtení hodnoty z 5.bitu P1
011   10.100   AEQ    100   ;je to 0 ?
012   05.101   LDA    101   ;vrácení načítané hodnoty do střadače
013   11.015   JF     015   ;ano - ukončení čítání
014   09.008   JMP    008   ; ne - zpět na nové čtení portu
```

Zobrazení znaku "°C" na levém okraji displeje:

```
adr   kód      instrukce     komentář
-------------------------------------------------------
015   04.030   LDC    030   ;kód znaku "°" do střadače
016   02.006   DISP   006   ;a zobrazení
017   04.108   LDC    108   ;kód znaku "C" do střadače
018   02.005   DISP   005   ;a zobrazení
```

Teploměr je cejchován pro měření teploty v rozsahu od 15 do 40°C, a proto je nutné před přepočtem provést kontrolu načítané hodnoty (zda leží v intervalu, který odpovídá teplotám od 15 do 40°C). Pro kontrolu slouží omezovací konstanty uložené v buňkách na adrese 117 (MIN) a 118 (MAX). Tyto hodnoty, právě tak jako obsah přepočtové tabulky, jsou závislé na způsobu zapojení MKO a na použitém typu termistoru.

```
adr   kód       instrukce     komentář
-------------------------------------------------------------
019   05.101    LDA    101    ;ACC:=načítaná hodnota
020   13.117    ALT    117    ;je menší než 35 ?
021   11.112    JF     112    ;ano => výpis chybového hlášení
022   12.118    AGT    118    ;je větší než 90 ?
023   11.112    JF     112    ;ano => výpis chybového hlášení
```

Podle načítané hodnoty měříme teplotu:

Obsah adresy 101, což je načítaná hodnota, použijeme jako adresu, na které bude uložena příslušná hodnota teploty již ve °C. K tomu je vhodná instrukce LDAI.

```
adr   kód       instrukce     komentář
-------------------------------------------------------------
024   19.101    LDAI   101    ;přiřazené načítání hodnoty k teplotě
025   02.000    DISP   000    ;tisk teploty
026   09.000    JMP    000    ;zpět na nové měření
```

Na displeji se zobrazí: 

, kde xx je změřená teplota a program cyklicky pokračuje znovu od začátku.

Úsek pro zpracování chyby:

```
adr   kód       instrukce     komentář
-------------------------------------------------------------
112   04.002    LDC    002    ;kód znaku "-"
113   02.001    DISP   001
114   02.002    DISP   002
115   02.003    DISP   003    ;tisk znaku "-" do datového pole
116   09.000    JMP    000    ;zpět na nové měření
```

Neleží-li teplota v přípustném rozsahu, zobrazí se zpráva o chybě a měří se znovu od začátku.

Zbývá ještě vyhradit místa pro uložení dat:

```
100   00.001    ;přičítaná hodnota
101   xx.xxx    ;zde se ukládá načítaná hodnota
102   00.000
 |
117   00.035    ;minimální přípustná načítaná hodnota
118   00.090    ;maximální přípustná načítaná hodnota
```

A tabulka pro přepočet načítaných hodnot na teplotu ve °C :

adr.	teplota-data	adr.	teplota-data	adr.	teplota-data
035	00.040	054	00.028	073	00.020
036	00.039	055	00.028	074	00.020
037	00.038	056	00.027	075	00.020
038	00.037	057	00.027	076	00.020
039	00.036	058	00.026	077	00.019
040	00.035	059	00.026	078	00.019
041	00.035	060	00.025	079	00.019
042	00.034	061	00.025	080	00.018
043	00.034	062	00.024	081	00.018
044	00.033	063	00.024	082	00.018
045	00.033	064	00.023	083	00.017
046	00.032	065	00.023	084	00.017
047	00.032	066	00.022	085	00.017
048	00.031	067	00.022	086	00.016
049	00.031	068	00.022	087	00.016
050	00.030	069	00.021	088	00.016
051	00.030	070	00.021	089	00.015
052	00.029	071	00.021	090	00.015
053	00.029	072	00.021		

Postup při odvození tabulky:

Do počítače napíšeme program od adresy 000 do 014.
Dále doplníme:

```
adr   kód      instrukce     komentář
.....................................................

015   02.000   DISP  000
016   09.000   JMP   000
100   00.001               ;příčítací konstanta
```

Po zapnutí programu se na displeji objeví načítaná hodnota. Na teploměru, který je ve stejném prostředí jako termistor, odečteme příslušnou teplotu. Teplotu postupně měníme a odečítáme další hodnoty pro přepočtovou tabulku.

Tento program ukazuje na jednu z variant použití tohoto zapojení s UCY 74121.

Další možností použití je zaměnění termistoru fotorezistorem, čímž získáme měřič osvětlení.

7.2 Zobrazování na displeji

Námět:

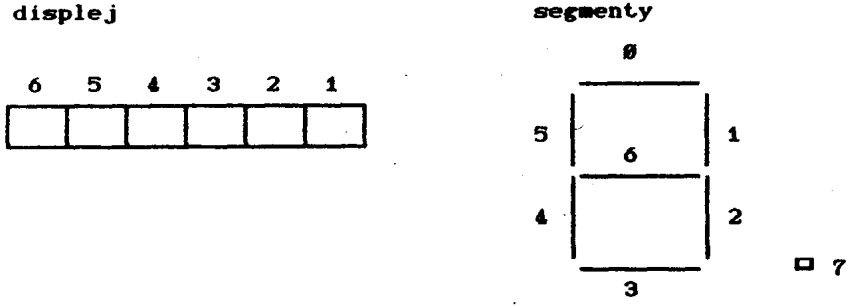
Utvořte program, který umožní na každé pozici displeje zobrazit libovolnou kombinaci segmentů a sestavovat tak různé grafické znaky.

Návod k řešení:

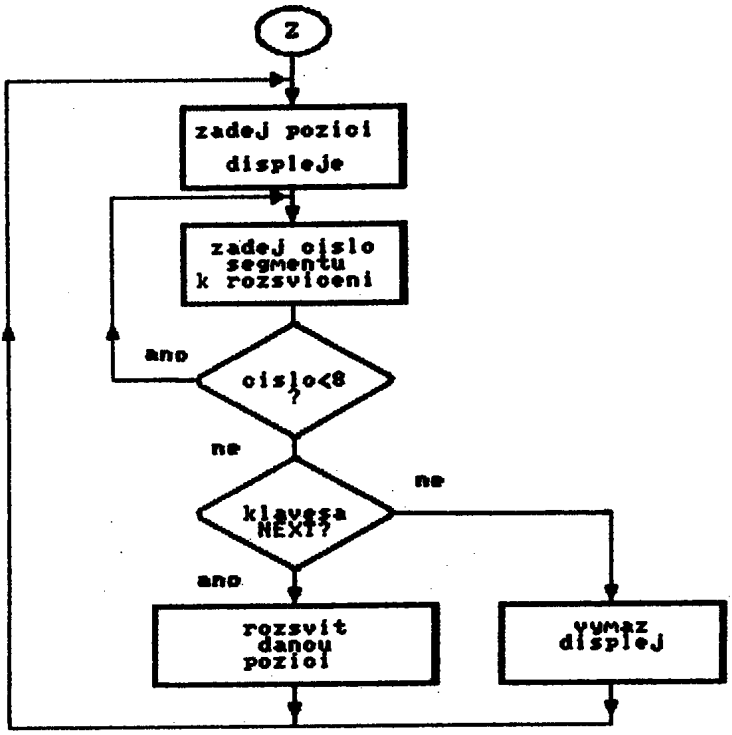
Pro zobrazování na displej využijeme instrukci DISP, jejímž parametrem lze určit pozici zobrazovaného znaku. Pozici znaku budeme zadávat z klávesnice. Po zadání pozice, kam se má znak zobrazit, budeme postupně vkládat čísla segmentů, která mají svítit. Po vložení všech čísel segmentů, které chceme rozsvítit, ukončíme vytváření zobrazovaného znaku omezovačem (NEXT) .

Při zadání jiného čísla segmentu než 0 až 7 se znak na vybrané pozici smaže a program pokračuje znovu od začátku.

Přiřazení pozic displeje a čísel segmentů je:



Algoritmus programu je jednoduchý - například:



Varianta programu:

Program je uložen od adresy 000 do adresy 019. Na adrese 050 je konstanta 128, která má význam pro testování stisknutého tlačítka instrukcí KEY.

Na adrese 053 je kódová adresa přepočtové tabulky, nutná pro zjištění kódu daného segmentu.

```
adr   kód       instrukce    komentář
------------------------------------------------------------------
000   26.000    KEY          ;čtení pozice displeje
001   10.050    AEQ  050     ;stisknuta klávesa ?
002   11.000    JF   000     ;ne, čekat
003   06.020    STA  020     ;ano, uložit jako parametr
                             ; instrukce DISP
004   26.000    KEY          ;čtení čísla segmentu
005   10.050    AEQ  050
006   11.004    ÚŘ   004     ;čekat na stisk tlačítka
007   13.051    ALT  051     ;číslo v rozsahu 0 až 7 ?
008   11.013    JF   013     ;ano
009   10.052    AEQ  052     ; NEXT ?
010   11.019    JF   019     ;ano, zobrazit znak
011   04.000    LDC  000     ; ne, smazat displej
012   09.020    JMP  020     ;a znovu od začátku
013   07.053    ADD  053     ;zjištění
019   06.015    STA  015     ; váhy
015   05.xxx    LDA  xxx     ; segmentu podle tabulky
016   07.054    ADD  054     ;přičtení dalšího segmentu
017   06.054    STA  054     ; ke kódu znaku
018   09.004    JMP  004     ;a číst číslo dalšího segmentu
019   05.054    LDA  054
020   02.xxx    DISP xxx     ;zobrazení znaku
021   04.000    LDC  000     ;počáteční nulování
022   06.054    STA  054     ; dalšího znaku
023   09.000    JMP  000     ;a znovu od začátku
------------------------------------------------------------------
050   00.128                 ; kód volné klávesnice
051   00.007                 ; horní přípustná mez čísel segmentů
052   00.012                 ; kód tlačítka NEXT
053   00.100                 ; ukládací adresa přepočtové tabulky
054   00.000                 ; kód znaku k zobrazení
------------------------------------------------------------------
                  ; tabulka vah segmentů
100   00.004                 ; segment 0
101   00.016                 ; segment 1
102   00.128                 ; segment 2
103   00.032                 ; segment 3
104   00.064                 ; segment 4
105   00.008                 ; segment 5
106   00.002                 ; segment 6
107   00.001                 ; segment 7
```

7.3 Sčítání čísel

Námět:

Vytvořte program, který sečte dvě čísla a výsledek zobrazí na displeji. Čísla mohou ležet v intervalu <0,999> a budou zadávána z klávesnice. Zadávané sčítance se ukončí stiskem klávesy (NEXT).

Je vhodné, aby program nereagoval na jiné klávesy než (0) až (9) a (NEXT).

Řešení:

Sčítanec vytvoříme sestavením z jednotlivých číslic (kódů tlačítek 0 až 9). Při stisku klávesy se její kód zobrazí v pravé části displeje (jednotkový řád sčítance). Při zadání další číslice se zobrazené cifry posouvají vlevo po displeji a poslední (stovkový řád) "vypadává". Je-li stisknuta klávesa (NEXT), zadávání sčítance se ukončí. Pro zobrazení cifry na vybraném místě displeje musíme vytvořit tabulku kódů číslic 0 až 9 - v programu na adr. 000 až 009.

Princip sčítání: sečteme jednotlivé řády, je-li výsledek větší než 9 odečteme 10 a k vyššímu řádu přičteme 1 (přenos). Výsledek zobrazíme.

Příklad zadávání čísla:

displej	3	2	1
nestisknuta klávesa	0	0	0
stisknuta "1"	0	0	1
stisknuta "2"	0	1	2
srisknuta "3"	1	2	3
stisknuta "4"	2	3	4

Začátek programu je na adrese 10. Proto je před spuštěním třeba nastavit čítač instrukcí PC na 10.

```
adr   kód        instrukce      komentář
------------------------------------------------------------------
000   00.252                    ; kód:   "0"
001   00.144                    ;        "1"
002   00.118                    ;        "2"
003   00.182                    ;        "3"
004   00.154                    ;        "4"
005   00.174                    ;        "5"
006   00.238                    ;        "6"
007   00.148                    ;        "7"
008   00.254                    ;        "8"
009   00.190                    ;        "9"
010   23.050     CALL  050      ; 1.sčítanec
011   05.112     LDA   112      ;přenesení cifer sčítance do
012   06.115     STA   115      ;jiného místa paměti (na adr.115)
013   05.113     LDA   113
014   06.116     STA   116
015   05.114     LDA   114
016   06.117     STA   117
017   23.050     CALL  050      ; 2.sčítanec
018   05.115     LDA   115
019   07.112     ADD   118      ;součet jednotkových řádů
020   23.042     CALL  042      ;je menší než 10 ?
021   11.023   ┌ JF    023      ;ano - skok
022   23.100   │ CALL  100      ;odečtení 10 a uložení pro
023   19.119   └→LDAI  119      ; přičtení k vyššímu řádu
024   02.001     DISP  001      ;zobrazení jednotkového řádu
025   05.116     LDA   116      ; výsledku
026   07.113     ADD   113      ;součet desítkových řádů
027   23.042     CALL  042
028   11.030   ┌ JF    030
029   23.100   │ CALL  100
030   19.119   └→LDAI  119
031   02.002     DISP  002      ;zobrazení desítkového řádu
032   05.117     LDA   117      ; výsledku
033   07.114     ADD   114      ;součet stovkových řádů
034   23.042     CALL  042
035   11.037   ┌ JF    037
036   23.100   │ CALL  100
037   19.119   └→LDAI  119
038   02.003     DISP  003      ;zobrazení stovkového řádu
039   19.122     LDAI  122      ; výsledku
040   02.004     DISP  004      ;je-li výsledek větší než 999,
                                ;zobrazení 1 na 4.místě displeje
                                ;(tisíce)
041   09.041   ┌→JMP   041      ; stop
------------------------------------------------------------------
042   07.122     ADD   122      ;korekce (přičtení 1-nižší řád>9
                                ; nebo 0)
043   13.118     ALT   118      ;je menší než 10?
044   06.119     STA   119      ;(nastaví FLAG)
045   04.000     LDC   000      ;bez přenosu do vyššího řádu
046   06.122     STA   122
047   24.000     RET
```

```
adr   kód        instrukce     komentář
------------------------------------------------------------------------
050   04.252     LDC   252     ;podprogram pro čtení čísla
051   02.004     DISP  004     ;zobrazení 0 na 4.místě displeje
052   04.000     LDC   000
053   02.000     DISP  000     ;nulování datového pole displeje
054   06.112     STA   112     ;nulování adres, ne kterých je
055   06.113     STA   113     ; sčítanec
056   06.114     STA   114
057   06.122     STA   122
058   23.090  ┌─→CALL  090     ;čtení cifry
059   10.110  │  AEQ   110     ;stisknuto tlačítko NEXT ?
060   11.083  │ ┌JF    083     ;ano - konec
061   12.120  │ │AGT   120     ;stisknuta jiná mlávesa než 0...9?
062   11.058  │ └JF    058     ;ano - zpět
063   06.112  │  STA   112
064   19.112  │  LDAI  112
065   02.001  │  DISP  001     ;tisk jednotkového řádu sčítance
066   19.113  │  LDAI  113
067   02.002  │  DISP  002     ;tisk desítkového řádu sčítance
068   19.114  │  LDAI  114
069   02.003  │  DISP  003     ;tisk stovkového řádu sčítance
070   23.090  │ ┌→CALL  090
071   10.110  │ │ AEQ   110
072   11.083  │ │ JF    083
073   12.120  │ │ AGT   120
074   11.070  │ └ JF    070
075   06.111  │  STA   111
076   05.113  │  LDA   113     ;rotace řádů sčítance v paměti
077   06.114  │  STA   114
078   05.112  │  LDA   112
079   06.113  │  STA   113
080   05.111  │  LDA   111
081   06.112  │  STA   112
082   09.064  │  JMP   064
083   24.000  └→ RET
------------------------------------------------------------------------
090   26.000  ┌→KEY            ;podprogram pro čtení znaku
091   10.121  │ AEQ   121
092   11.090  └ JF    090
093   24.000    RET

100   04.001    LDC   001     ;korekce při přenosu
101   06.122    STA   122     ;do vyššího řádu
102   05.119    LDA   119
103   08.118    SUB   118     ;odečtení 10
104   06.119    STA   119
105   24.000    RET
------------------------------------------------------------------------
                ; pomocná data
110   00.012    ; konstanta pro testování klávesy NEXT
111   xx.xxx    ; pomocná buňka
112   xx.xxx    ; 2.sčítanec
113   xx.xxx    ;    - " -
114   xx.xxx    ;    - " -
115   xx.xxx    ; 1.sčítanec
116   xx.xxx    ;    - " -
117   xx.xxx    ;    - " -
118   00.010    ; konstanta 10
119   xx.xxx    ; pomocná buňka
120   00.009    ; konstanta pro porovnání velikosti cifry
121   00.128    ; konstanta pro test klávesnice
122   xx.xxx    ; místo pro uložení přenosu při sčítání
```

7.4 Putující segment na displeji

Námět:

Vytvořte program, který rozsvítí na 6. displeji 6. segment a bude ho cyklicky posouvat vpravo.

Řešení:

Program je velice jednoduchý. Jediné, na co nesmíte zapomenout, je mazání displeje po posuvu na další pozici.

Pokuste se tento program vytvořit co nejkratší. Jedna z možných variant je tato:

```
adr   kód       instrukce    komentář
.....................................................

000   05.100 ┌─→ LDA   100
001   06.009 │   STA   009    ;nastavení čísla displeje
002   06.012 │   STA   012
003   08.102 │   SUB   102    ;odečte 1
004   12.101 │   AGT   101    ;rozsvícen segment na 1.displeji ?
005   11.007 │┌─ JF    007    ;ne - skok
006   04.006 │└─ LDC   006    ;jinak znovu od 6.pozice
007   06.100 └─→ STA   100
008   04.002 │   LDC   002    ;vzorek k rozsvícení
009   02.xxx │   DISP  xxx    ;operand se získá průběžně
010   03.050 │   NOP   050    ;prodleva
011   04.000 │   LDC   000
012   02.xxx │   DISP  xxx    ;smazání displeje
013   03.050 │   NOP   050    ;opět s prodlevou
014   09.000 └── JMP   000    ;a pořád dokola



100   00.006      ; pozice displeje
101   00.000      ; omezovač cyklického posunu pozice
102   00.001      ; konstanta pro posun pozice displeje
```

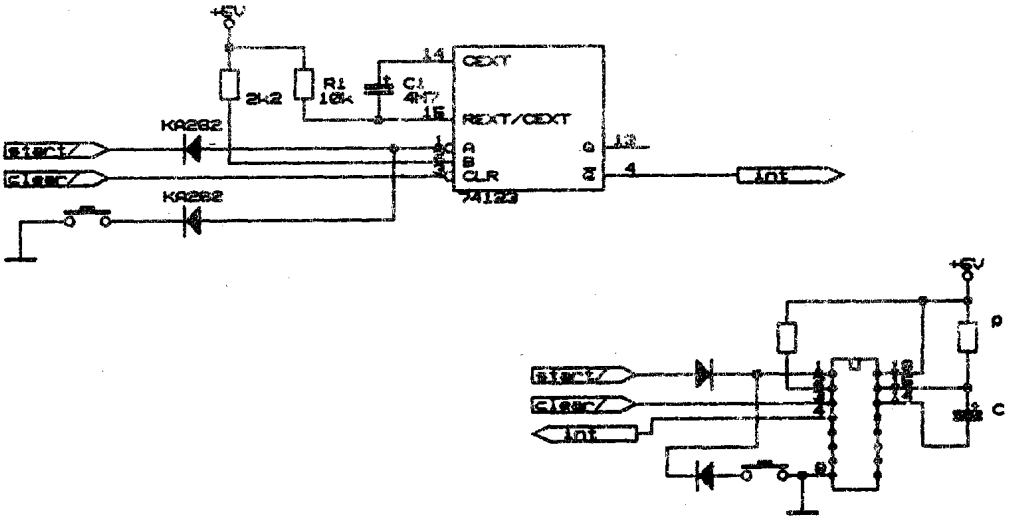
7.5 Zapojení vnějšího přerušení

Námět:

Zajistěte generování přerušovacího pulzu vhodné délky - cca 10 μs. Podrobně ji viz odstavec 2.1 a popis instrukce INT i.

Řešení:

Pulz bude generován monostabilním klopným obvodem UCY74123 zapojeným podle níže uvedeného obrázku.



Vývod INT lze přímo připojit na stejnojmennou svorku mikropočítače PETR. Stisknutím tlačítka se na výstupu INT monostabilního klopného obvodu vyrobí pulz úrovně 0. Jeho délka je přibližně 10 μs a lze ji ovlivnit změnou hodnot kondenzátoru C1 a odporu R1.

Při stejném zapojení lze vyrobit přerušovací pulz i programem nebo na žádost HW z vnější aplikace. K tomu slouží vstup START, který sestupnou hranou spustí monostabilní klopný obvod.

Programem lze i zkrátit přednastavenou délku přerušovacího pulzu opět sestupnou hranou na vstupu CLEAR. Dále je uveden příklad na obsluhu přerušení s využitím vývodů START a CLEAR . Při každém přerušení se inkrementuje (zvyšuje o 1) čítač, jehož hodnota je každou sekundu vypisována hlavním programem na displej. Přitom START je připojen na linku P10 a CLEAR na linku P11.

Program spusťte od adresy 001 !

```
adr   kód       instrukce     komentář
------------------------------------------------------
000   23.020    CALL  INTR    ;volání programu pro obsluhu
                              ; přerušení
```

Hlavní program pro výpis čítače:

```
adr   kód       instrukce     komentář
------------------------------------------------------
001   02.255    DISP  255     ;zhasnout celý displej
002   04.000    LDC   0       ;hodnota čítače bude ve střadači,
                              ; na začátku je ACC:=0
003   27.001    INT   1       ;povolit vnější přerušení
004   02.000 ┌─→DISP  0       ;vypsat obsah čítače na displej
005   03.255 │  NOP   255     ;čekat cca 1 sekundu
006   03.255 │  NOP   255
007   03.255 │  NOP   255
008   03.016 │  NOP   16
009   06.016 │  STA   016     ;uschovat čítač
010   05.017 │  LDA   017
011   17.000 │  P1OUT 0       ;START (tj. P10) := 0
               │                ;spustí pulz na vstupu INT
012   05.018 │  LDA   018
013   17.000 │  P1OUT 0       ;START := 1
014   05.016 │  LDA   016     ;a ještě obnovit hodnotu čítače !
015   09.004 └─ JMP   004
016   00.xxx    ; místo pro úschovu čítače
017   00.000    ; konstanta 0
018   00.001    ; konstanta 1
```

Obsluha přerušení může být například:

```
adr   kód        instrukce      komentář
020   06.016 INTR: STA 016      ; * uschovat čítač
021   05.017       LDA  17      ; *
022   17.001       P1OUT 1      ; * CLEAR (tj. P11) := 0
023   05.018       LDA  18      ; *
024   17.001       P1OUT 1      ; * CLEAR := 1
025   05.016       LDA  016     ; * obnovit čítač
026   07.018       ADD  018     ; zvýšit jej o 1
027   27.001       INT  1       ;znovu povolit další přerušení
                                ;  (důležité !)
028   24.000       RET          ;a návrat do hlavního programu
```

Poznámka:

1) V komentáři hvězdičkou * označené instrukce slouží ke zkrácení nulového pulzu na výstupu INT monostabilního klopného obvodu. To je důležité v případě, že jeho délka, daná hodnotou prvků C1 a R1, je větší než je délka trvání obsluhy přerušení. Pak by se totiž jedno a totéž přerušení přijmulo opakovaně (obdobně jako v pozn. 3).

2) Všimněte si, že po instrukci RET bude hlavní program pokračovat z místa, ve kterém byl přerušen. Proč ? Protože při přerušení se zpracovává vždy jedna vnucená instrukce z adresy 000, v našem příkladě je to instrukce CALL INTR. Ta volá podprogram na adrese INTR=020, zatím co návratovou adresu do přerušeného hlavního programu odloží do zásobníku. Tuto adresu pak využije instrukce RET.

3) Co se stane, zapojíme-li vstup INT trvale na úroveň 0 ? Náš příklad skončí chybou přetečení zásobníku. Proč ? Protože po každém povolení přerušení instrukcí INT 1 (na adrese 027) se ještě před provedením instrukce RET vyžádá znovu obsluha přerušení, instrukce CALL INTR na adrese 000 bude rekurzivně (tj. stále dokola) volat podprogram na adrese INTR=020 až se zásobník přeplní návratovými adresami a běh programu skončí chybou E 005 .

7.6 Převody soustav

Námět:

Vytvořte program, který převede číslo z desítkové do šestnáctkové soustavy.

Řešení:

Zadané číslo (v rozmezí 0 + 255) program přečte ze zvolené adresy (např. 100). Dělí ho 16, celočíselný výsledek dělení je vyšším řádem převedeného čísla, zbytek je řádem nižším.

Princip dělení: Od daného čísla odečítáme 16 tak dlouho, dokud výsledek není menší než dělitel, tj. 16. Počet odečtení zaznamenáváme do paměti na adresu 104. Výsledek zobrazíme pomocí instrukce LDAI *aaa* a tabulky kódů znaků.

```
adr   kód        instrukce     komentář
                 ; tabulka znaků
000   00.252                   ; "0"
001   00.144                   ; "1"
002   00.118                   ; "2"
003   00.182                   ; "3"
004   00.154                   ; "4"
005   00.174                   ; "5"
006   00.238                   ; "6"
007   00.148                   ; "7"
008   00.254                   ; "8"
009   00.190                   ; "9"
010   00.222                   ; "A"
011   00.234                   ; "B"
012   00.108                   ; "C"
013   00.242                   ; "D"
014   00.110                   ; "E"
015   00.078                   ; "F"
```

Vlastní program:

```
adr   kód        instrukce     komentář
.....................................................
020   04.000     LDC   000
021   06.104     STA   104     ;nulování paměti, ve které je
                               ;  vyšší řád výsledku
022   05.100     LDA   100     ;vyzvednutí převáděného čísla
023   13.102   ┌→ALT   102     ;je menší než 16 ?
024   11.032   │ JF    032     ;ano - skok na zobrazení
025   08.102   │ SUB   102     ;odečtení 16 (tj. základu hexa
                               ;  soustavy)
026   06.103   │ STA   103     ;uložení mezivýsledku
027   05.104   │ LDA   104
028   07.105   │ ADD   105     ;přičtení 1
029   06.104   │ STA   104     ;k vyššímu řádu výsledku
030   05.103   │ LDA   103     ;vrácení mezivýsledku do střadače
031   09.023   └ JMP   023     ;a znovu dokola
032   06.101   └→STA   101
033   19.101     LDAI  101
034   02.001     DISP  001     ;zobrazení nižšího řádu výsledku
035   19.104     LDAI  104
036   02.002     DISP  002     ;zobrazení vyššího řádu výsledku
037   09.037   ┌→JMP   037
  ⋮
100   xx.xxx                   ;číslo, určené na převod
101   xx.xxx                   ;nižší řád výsledku
102   00.016                   ;základ šestnáctkové soustavy
103   xx.xxx                   ;mezivýsledek
104   xx.xxx                   ;vyšší řád výsledku
105   00.001
```

Před spuštěním programu je nutné nastavit čítač instrukce PC na adresu 020 (počáteční adresa našeho programu).

Tabulku znaků je možné posunout do jiného místa paměti (např. od adresy 090). V tomto případě k výsledku dělení či zbytku přičteme 90. Ukázka je v programu "HODINY" - odst. 7.7).

7.7. Hodiny

Námět:

Vytvořte program, který bude přibližně měřit čas v sekundách, minutách a hodinách.

Řešení:

V programu vytvoříme smyčku, jejíž délka je přibližně 1 sekunda. V této smyčce přičítáme 1 k čítači sekund na adrese 100 tak dlouho, dokud výsledek nebude 60. Je-li již výsledek 60 vynulujeme obsah této adresy a zvýšíme o 1 čítač minut na adrese 101. Obdobně postupujeme i při dočítání 60ti minut a 24 hodin.

Zobrazení času: Protože tato část programu je poněkud složitější, vysvětlíme ji podrobněji.

Naměřené sekundy, minuty či hodiny není možné zobrazit najednou, ale postupně po jednotlivých cifrách.

Dále je uveden postup při zobrazení sekund (zobrazení minut a hodin je obdobné): Naměřené sekundy (dvojciferné číslo) potřebujeme rozdělit na 2 cifry - desítky a jednotky. Proto tuto hodnotu dělíme 10 (postup dělení je popsán v programu "Převody soustav" - odst. 7.6), celočíselný výsledek dělení udává desítky sekund a zobrazí se na displej 2. Zbytek po dělení udává jednotky sekund a zobrazí se na 1.displeji. Dělení a zobrazení naměřeného času je v podprogramu od adresy 040.
Aby mohl být tento podprogram využit i pro zobrazení minut a hodin na 3. až 6. displeji je třeba přepsat čísla displejů (tj. operandy v instrukcích DISP na adresách 053 a 058) vždy před skokem do tohoto podprogramu. (Provádí se instrukcemi STA na adresách 071, 073, 077, 079, 083, 085 ...).

Tabulka znaků je posunuta od adresy 090 přičtením konstanty 90 k zobrazované cifře.

```
adr   kód       instrukce     komentář
------------------------------------------------------------------

000   04.000 ┌─→ LDC   000     ;nulování hodin
001   06.100 │   STA   100
002   06.101 │   STA   101
003   06.102 │   STA   102
004   06.121 │   STA   121
005   23.070 │ ┌→ CALL  070
006   03.255 │ │   NOP   255   ;nastavení přesnosti chodu
007   03.255 │ │   NOP   255
008   03.255 │ │   NOP   255
009   03.000 │ │   NOP   000
010   05.100 │ │   LDA   100   ;počet sekund do ACC
011   07.110 │ │   ADD   110   ;zvýšení o 1 sekundu
012   06.100 │ │   STA   100   ;uložení
013   13.111 │ │   ALT   111   ;již 60 sekund ?
014   11.005 │ ├─ JF    005   ;ne - zpět
015   04.000 │ │   LDC   000   ;nulování sekund
016   06.100 │ │   STA   100
017   05.101 │ │   LDA   101   ;počet minut do ACC
018   07.110 │ │   ADD   110   ;zvýšení o 1 minutu
019   06.101 │ │   STA   101   ;uložení
020   13.111 │ │   ALT   111   ;již 60 minut ?
021   11.005 │ ├─ JF    005   ;ne - zpět
022   04.000 │ │   LDC   000   ;nulování minut
023   06.101 │ │   STA   101
024   05.102 │ │   LDA   102   ;počet hodin do ACC
025   07.110 │ │   ADD   110   ;zvýšení o 1 hodinu
026   06.102 │ │   STA   102   ;uložení
027   13.112 │ │   ALT   112   ;již 24 hodin ?
028   11.005 │ └─ JF    005   ;ne - zpět
029   09.000 └──── JMP   000   ;nulování času, nové měření
------------------------------------------------------------------
```

Podprogram pro zobrazení dvou cifer:

```
040   06.120      STA   120   ;uschování
041   13.113 ┌──→ ALT   113   ;menší než 10 ?
042   11.050 │ ┌─ JF    050   ;ano - skok
043   08.113 │ │   SUB   113   ;odečtení 10
044   06.120 │ │   STA   120   ;uschování
045   05.121 │ │   LDA   121   ;přičtení 1 k obsahu paměti 121
046   07.110 │ │   ADD   110
047   06.121 │ │   STA   121
048   05.120 │ │   LDA   120
049   09.041 │ └─ JMP   041   ;zpět
050   07.114 │ ┌→ ADD   114   ;přesun na tabulku znaků
051   06.120 │ │   STA   120   ;  (začíná na adrese 90)
052   19.120 │ │   LDAI  120
053   02.xxx │ │   DISP  xxx   ;operand (číslo displeje) se
054   05.121 │ │   LDA   121   ;  doplní průběžně
055   07.114 │ │   ADD   114
056   06.121 │ │   STA   121
057   19.121 │ │   LDAI  121
058   02.xxx │ │   DISP  xxx
059   04.000 │ │   LDC   000   ;nulování pomocných adres
060   06.120 │ │   STA   120
061   06.121 │ │   STA   121
062   24.000 │ │   RET
```

Podprogram pro zobrazení času:

```
adr    kód         instrukce      komentář
-------------------------------------------------------------------
070   04.001       LDC    001     ;číslo displeje
071   06.053       STA    053     ;doplnění operandu v instr. DISP
072   04.008       LDC    002
073   06.058       STA    058
074   05.100       LDA    100     ;počet sekund do ACC
075   23.040       CALL   040     ;podprogram pro zobrazení
076   04.003       LDC    003
077   06.053       STA    053
078   04.004       LDC    004
079   06.058       STA    058
080   05.101       LDA    101     ;počet minut do ACC
081   23.040       CALL   040
082   04.005       LDC    005
083   06.053       STA    053
084   04.006       LDC    006
085   06.058       STA    058
086   05.102       LDA    102     ;počet hodin do ACC
087   23.040       CALL   040
088..24.000        RET
-------------------------------------------------------------------
                tabulka znaků
090   00.252                      ; "0"
091   00.144                      ; "1"
092   00.118                      ; "2"
093   00.182                      ; "3"
094   00.154                      ; "4"
095   00.174                      ; "5"
096   00.238                      ; "6"
097   00.148                      ; "7"
098   00.254                      ; "8"
099   00.190                      ; "9"
100   xx.xxx                      ;sekundy
101   xx.xxx                      ;minuty
102   xx.xxx                      ;hodiny
 ¦     ¦
110   00.001
111   00.060
112   00.024
113   00.010
114   00.090      ; konstanta pro posun na tabulku znaků
 ¦     ¦          ; (tj. adresa tabulky)
120   xx.xxx      ; pomocné výsledky
121   xx.xxx
```

Při spuštění programu z adresy 000 dojde k vynulování času. Chceme-li hodiny spustit od času, který nastavíme v adresách 100 + 102 , musíme na adresu 121 vložit 000 a spustit program od adresy 005.

LITERATURA

/1/ KOSMOS-Computer-Praxis, Das universelle Mikroprozessor-System, Franckh'sehe Verlagshandlung Stuttgart.

/2/ Barták, Kurt: Mikrořadiče MCS-48. Tesla ELTOS ČSVTS, Pardubice 1983.

/3/ Nohel, Jiří - Machačka, Ivo: Základní instrukce mikroprocesoru 8048. Tesla ELTOS o.p., Praha 1983.

/4/ Nohel, Jiří a kol.: Asembler 8048. Tesla ELTOS-IMA, Praha 1984.

/5/ Kruml, Jaroslav: Podklady pro technické podmínky integrovaného obvodu N-MOS MHB 8048/8035 mikropočítač [výzkumná zpráva]. Tesla VÚST, Praha 1984.

/6/ Černoch, Michal - Stehno, Zdeněk - Vybulková, Vlasta: Mikropočítač 8048. Sdělovací technika, srpen 1983, č. 8, s. 283-300.

/7/ Švéda, Miroslav: Programování pro jednočipové mikropočítače jako zvláštní disciplina. Tesla ELTOS-IMA. Mikrosystém 1/1985, s. 11-13.

/8/ Trpišovský, Tomáš - Zeman, Vojtěch - a kol.: Emulátor TEMS 49 Uživatelská příručka. Tesla ELTOS-IMA, Praha 1985.

/9/ Trpišovský, Tomáš - Couf, Petr: FL48 - knihovna podprogramů pro 3-bytovou binární aritmetiku v pohyblivé řádové čárce pro jednočipový mikropočítač 8048. Tesla ELTOS-IMA, Praha 1986.

/10/ Trpišovský, T. - Zeman, V.: Emulátor TEMS 49 popis rozšířené verze V-02. Tesla ELTOS-IMA, Praha 1986.

/11/ Mužík, V. - a kol.: Uživatelská příručka mikropočítačů řady 48. Knižnice ČSVTS Mikroprocesorová technika. Praha 1985.

/12/ Mužík, v. - a kol.: Příručka programování mikropočítačů řady 48. Knižnice ČSVTS Mikroprocesorová technika. Praha 1986.

/13/ Horák, V.: Jednočipové mikropočítače řady 8048. Amatérské rádio A/č.7/86 (s. 257-259), A/č.8/86 (s. 301-303), A/č.9/86 (s. 339-341).

/14/ Operační systém AMOS verze 4.0, příručka uživatele, KKI MFF UK, 1987.

/15/ Vlček, J. - Bajbar, J. - Korbel', P.- Smolka, J.: Technické prostriedky a aplikácie jednočipových mikropočítačov série 8048. ČSVTS Tesla VRUSE, Bratislava 1985.

/16/ Korbel', P. - Smolka, J. - Bajbar, J. - Vlček, J.: Programovanie jednočipových mikropočítačov série 8048. ČSVTS Tesla VRUSE, Bratislava 1985.

/17/ Amatérské rádio č. 8/80 (s. 292-293).

/18/ Donovan, J.: Systémové programovanie, slovenský překlad Alfa, 1983.

/19/ Horák, V. - Trpišovský, T. - Couf,P.: Jednočipové mikropočítače a jejich využití. In: Sborník semináře MOP'87. MFF UK Praha, Vinné 1987.

PŘÍLOHA 1: Výpis programu PETR

```
DEBUG ASM48 PV02.SRC PAGEWIDTH(95) NOOBJECT XREF


ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V3.0                PAGE    1


 LOC  OBJ          LINE        SOURCE STATEMENT

                     1 $MACROFILE NOGEN NOCOND
                     2 $SAVE NOLIST
                   231 $RESTORE
                   232 ;
                   233 ;*******************************************************
                   234 ;*                                                     *
                   235 ;*   RIDICI  PROGRAM  PRO  STAVEBNICI    P  E  T  R    *
                   236 ;*                                                     *
                   237 ;*       P.C.  16. 6.1986                              *
                   238 ;*       T.T.   2. 7.1986  REV.                        *
                   239 ;*       P.C.  13.10.1986  OVLADAC MGF V.1             *
                   240 ;*       P.C.   6. 2.1987  REV., SNIZENI ROZSAHU       *
                   241 ;*       P.C.  12. 3.1987  OVLADAC MGF V.2             *
                   242 ;*       P.C.  20. 8.1987  UPRAVA PRO NOVY HARDWARE    *
                   243 ;*       T.T./P.C. XII/87  REV. OVEROVACI VERZE 1 KB   *
                   244 ;*       T.T.  II/88       REV. -) PV02.SRC            *
                   245 ;*                                                     *
                   246 ;*******************************************************
                   247 ;
                   248 ;       XTAL 6 MHZ
                   249 ;
                   250 ;       H  A  R  D  W  A  R  E
                   251 ;
                   252 ;       T1=1    REZIM  "PETR"
                   253 ;       T1=0    REZIM  "EXTERNI ROM"
                   254 ;       ADRESY V EXTERNI ROM:
0400               255 EXTRES  EQU     400H            ;RESET
0403               256 EXTINT  EQU     403H            ;EXTERNAL INTERRUPT
0407               257 EXTTIM  EQU     407H            ;TIMER/COUNTER INTERRUPT
                   258 ;
                   259 ; OBSAZENI PORTU 8048
                   260 ;       P10-P17 .. EXTERNAL  ("PETR" - PORT 1)
                   261 ;
                   262 ;       P20-P23 .. PRO ROZSIROVANI SYSTEMU
                   263 ;        NAVIC P23 .. IO/M (OBOU) 8155
                   264 ;              P22 .. CE VNITRNI 8155
                   265 ;              P21 .. CE PRIDAVNE 8155
                   266 ;       P24     .. VYSTUP MUX7 (POUZE PRO KLAVESNICI)
                   267 ;       P25-P27 .. VSTUPNI LINKY MATICE KLAVESNICE
0008               268 P55IOM  EQU     00001000B       ;ZAPOJENI
0004               269 P55CE0  EQU     00000100B       ;JEDNOTLIVYCH
0002               270 P55CE1  EQU     00000010B       ;VODICU
0010               271 MUX7    EQU     00010000B       ;BRANY P2
                   272 ;
                   273 ;       T0      .. EXTERNAL  ("PETR" - LINKA T)
                   274 ;       INT     .. EXTERNAL  ("PETR" - LINKA INT)
                   275 ;
                   276 ; OBSAZENI PORTU 8155
                   277 ;       PA0-PA7 .. EXTERNAL  ("PETR" - PORT 2)
                   278 ;       PB0-PB7 .. SEGMENTY DISPLEJE
                   279 ;       PC0-PC5 .. VYSTUP MUX1-MUX6 (DISPLEJ A KLAVESNICE)
                   280 ;
                   281 ; PRIPOJENI  MAGNETOFONU
                   282 ;       MGF OUTPUT = P11
```

```
LOC  OBJ          LINE        SOURCE STATEMENT

                   283 ;       MGF INPUT  = P10
0002               284 MGBITW  EQU     00000010B       ;PRO ZAPIS
0001               285 MGBITR  EQU     00000001B       ;PRO CTENI
                   286 ;
                   287 ;
                   288 ;       S  O  F  T  W  A  R  E
                   289 ;
                   290 ;       OBSAZENI REGISTRU
                   291 ;
                   292 ;       RB0 - PRACOVNI REGISTRY
                   293 ;           R4  "PETR" PRIZNAK F
                   294 ;           R6  AKTUALNI ADRESA MEM
                   295 ;           R7  "PETR" HODNOTA PC
                   296 ;
                   297 ;       RB1 - PRERUSENI OD CASOVACE
                   298 ;           R0  USCHOVA STAVU LINEK P20-P23
                   299 ;           R1  PRACOVNI SMERNIK
                   300 ;           R2  CITAC MUX
                   301 ;           R3  DETEKCE KLAVESY "STOP"
001B               302 STOP?   EQU     1BH             ;SMERNIK NA R3
                   303 ;           R4  CITAC CHVENI PRI DETEKCI TLACITKA
                   304 ;           R5  POSLEDNI STISKNUTE TLACITKO
                   305 ;           R6  DETEKOVANE TLACITKO
001E               306 KEY     EQU     1EH             ;SMERNIK NA R6
                   307 ;           R7  USCHOVA ACC
                   308 ;
                   309 ;       OBSAZENI INTERNI RAM
                   310 ;
0020               311 VID0    EQU     20H             ;VIDEOREGISTRY. PRAVY
0021               312 VID1    EQU     VID0+1          ; .
0022               313 VID2    EQU     VID0+2          ; .
0023               314 VID3    EQU     VID0+3          ; .
0024               315 VID4    EQU     VID0+4          ; .
0025               316 VID5    EQU     VID0+5          ; LEVY
0026               317 SP      EQU     26H             ;"PETR" STACK POINTER
0027               318 ACC     EQU     27H             ;"PETR" ACC
0028               319 TIM128  EQU     28H             ;CASOVAC 1.28 MS
0029               320 P1D     EQU     29H             ;KOPIE DAT NA P1 (8048)
002A               321 P2D     EQU     2AH             ;KOPIE DAT NA P2 (PA 8155)
002B               322 BUFF    EQU     2BH             ;BUFFER CISLIC 2BH - 30H
0031               323 STKMIN  EQU     31H             ;OBLAST PRO ZASOBNIK
003A               324 STKMAX  EQU     3AH             ;  10 POLOZEK
                   325 ;       3BH-3FH ... VOLNE
                   326 ;
                   327 ;       PRIZNAKY
                   328 ;       F0 .. PRITOMNOST ROZSIRUJICI 8155
                   329 ;       F1 .. ZADOST O PRERUSENI PROCESORU PETR
                   330 ;
                   331 ;       KONSTANTY
                   332 ;
00F0               333 KTIM    EQU     0F0H            ;KONSTANTA PRO CASOVAC
                   334                                 ;  (1.28 MS, CCA 800HZ)
0003               335 CHVENI  EQU     3               ;POCET CTENI NUTNYCH PRO
                   336                                 ;   DETEKCI TLACITKA
                   337 ;
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

0000             338 P55R    EQU     0               ;RIZENI 8155
0001             339 P55A    EQU     1               ;PORT A
0002             340 P55B    EQU     2               ;PORT B
0003             341 P55C    EQU     3               ;PORT C
                 342 ;
                 343 ;       CISLA CHYB
                 344 ;
0001             345 ERR1    EQU     1               ;PRETECENI PRI DEC-->BIN
0002             346 ERR2    EQU     2               ;ADRESA MIMO PAMET
0003             347 ERR3    EQU     3               ;NEZNAMA INSTRUKCE
0004             348 ERR4    EQU     4               ;OPERAND MIMO ROZSAH
0005             349 ERR5    EQU     5               ;PREPLNENI ZASOBNIKU (CALL)
0006             350 ERR6    EQU     6               ;PRAZDNY ZASOBNIK (RET)
0007             351 ERR7    EQU     7               ;CHYBA CTENI Z MGF
                 352 ;
                 353 ;       KONSTANTY PRO ZOBRAZOVANI
                 354 ;
                 355 ;DEFINICE ZAPOJENI SEGMENTU NA JEDNOTLIVE LINKY PORTU B
                 356 ;
0004             357 S0      EQU     00000100B       ;   ---- S0 ----
0010             358 S1      EQU     00010000B       ; !             !
0080             359 S2      EQU     10000000B       ; S5            S1
0020             360 S3      EQU     00100000B       ; !             !
0040             361 S4      EQU     01000000B       ;   ---- S6 ----
0008             362 S5      EQU     00001000B       ; !             !
0002             363 S6      EQU     00000010B       ; S4            S2
0001             364 S7      EQU     00000001B       ; !             !
0001             365 ZTECKA  EQU     S7              ;   ---- S3 ----      S7
                 366 ;
00FC             367 ZZ0     EQU     S0+S1+S2+S3+S4+S5
0090             368 ZZ1     EQU     S1+S2
0076             369 ZZ2     EQU     S0+S1+S3+S4+S6
00B6             370 ZZ3     EQU     S0+S1+S2+S3+S6
009A             371 ZZ4     EQU     S1+S2+S5+S6
00AE             372 ZZ5     EQU     S0+S2+S3+S5+S6
00EE             373 ZZ6     EQU     S0+S2+S3+S4+S5+S6
0094             374 ZZ7     EQU     S0+S1+S2
00FE             375 ZZ8     EQU     S0+S1+S2+S3+S4+S5+S6
00BE             376 ZZ9     EQU     S0+S1+S2+S3+S5+S6
00DE             377 ZZA     EQU     S0+S1+S2+S4+S5+S6
005E             378 ZZP     EQU     S0+S1+S4+S5+S6
006E             379 ZZE     EQU     S0+S3+S4+S5+S6
004E             380 ZZF     EQU     S0+S4+S5+S6
00AE             381 ZZS     EQU     ZZ5
00C2             382 ZZM     EQU     S2+S4+S6
00BA             383 ZZH     EQU     S1+S2+S4+S5+S6
006C             384 ZZC     EQU     S0+S3+S4+S5
00E2             385 ZRUN    EQU     S2+S3+S4+S6
                 386 ;
                 387 ; KODY TLACITEK
                 388 ;
0000             389 TL0     EQU     0               ;SKUPINA CISLIC
0001             390 TL1     EQU     TL0+1           ;MUSI BYT ZA SEBOU
0002             391 TL2     EQU     TL0+2
0003             392 TL3     EQU     TL0+3
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

0004            393 TL4     EQU     TL0+4
0005            394 TL5     EQU     TL0+5
0006            395 TL6     EQU     TL0+6
0007            396 TL7     EQU     TL0+7
0008            397 TL8     EQU     TL0+8
0009            398 TL9     EQU     TL0+9
                399 ;
000A            400 TLPREV  EQU     10              ;TYTO TRI KODY
000B            401 TLEND   EQU     11              ;MUSI NASLEDOVAT
000C            402 TLNEXT  EQU     12              ;ZA SEBOU !
                403 ;
000D            404 TLMEM   EQU     13              ;PRIKAZOVE KLAVESY
000E            405 TLACC   EQU     14              ;MUSI BYT ZA SEBOU
000F            406 TLPC    EQU     15
0010            407 TLRUN   EQU     16
0011            408 TLSTEP  EQU     17
0012            409 TLSAVE  EQU     18
0013            410 TLLOAD  EQU     19
                411 ;
000B            412 TLSTOP  EQU     TLEND           ;TLACITKO VE FUNKCI STOP
0080            413 NOKEY   EQU     80H
                414 ;
                415         DATABLK 10
0300            461+        ORG     768
                465 ;
                466 ;       TABULKA PRO ZOBRAZOVANI CISLIC
                467 ;       UMISTENA OD ZACATKU PAGE 3
                468 ;
0300 FC         469 TABCIS: DB      ZZ0,ZZ1,ZZ2,ZZ3,ZZ4
0301 90
0302 76
0303 B6
0304 9A
0305 AE         470         DB      ZZ5,ZZ6,ZZ7,ZZ8,ZZ9
0306 EE
0307 94
0308 FE
0309 BE
                471         ;
                472         SIZECHK
000A            475+SIZE    SET     10
                476+;
                477+;++++++++++++++++++++++++++++++++++++
                478+;
                489         CBLK48  160
0000            496+        ORG     0
                500 ;
                501 ;*************************************************************
                502 ;*                                                           *
                503 ;*              VEKTOR  RESTART  A  PRERUSENI                *
                504 ;*                                                           *
                505 ;*************************************************************
                506 ;
                507 ; ADRESA 0 - RESET
0000 0485       508         JMP     RESET
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

0002 00          509         NOP
                 510 ; ADRESA 3 - EXT INT
0003 5681        511         JT1     INTINT          ;PRERUSENI PETR
0005 8403        512         JMP     EXTINT
                 513 ; ADRESA 7 - TIMER/COUNTER INT
0007 560B        514         JT1     TIMER
0009 8407        515         JMP     EXTTIM
                 516 ;
                 517 ;*******************************************************
                 518 ;*                                                     *
                 519 ;*              PRERUSENI  OD  CASOVACE                *
                 520 ;*                                                     *
                 521 ;*******************************************************
                 522 ;
000B D5          523 TIMER:  SEL     RB1
000C AF          524         MOV     R7,A            ;USCHOVA ACC
000D 23F0        525         MOV     A,#KTIM
000F 62          526         MOV     T,A             ;CASOVAC
                 527 ; USCHOVA STAVU LINEK P20 - P23 (NA KONCI HO OBNOVI)
0010 0A          528         IN      A,P2
0011 A8          529         MOV     R0,A
                 530 ; OBSLUHA DISPLEJE
                 531 ; 1. ZHASNOUT SVITICI ZNAK, MAPOVAT VNITRNI 8155, IO
0012 74F1        532         CALL    ZHAS0
                 533 ; 2. POSUN MULTIPLEXU. NYNI JE LINKA MUX7=1
0014 EA1A        534         DJNZ    R2,TIM2         ;CITAC MULTIPLEXU
                 535 ;       PRO MUX=7: NASTAVIT LINKU P24 (VYSTUP MUX7) := 0
0016 9AEF        536         ANL     P2,#NOT MUX7
0018 BA07        537         MOV     R2,#7           ;REINCIALIZACE CITACE
                 538 TIM2:   ;NASTAVENI LINEK MUX1-MUX6 (BRANA C 8155)
001A FA          539         MOV     A,R2            ;CITAC MULTIPLEXU
001B 0379        540         ADD     A,#LOW TABMUX-1
001D A3          541         MOVP    A,@A            ;A=DATA K ZAPISU
001E B903        542         MOV     R1,#P55C        ;  NA BRANU C 8155
0020 91          543         MOVX    @R1,A           ;ZAPIS MUX1-MUX6
                 544 ; 3. ROZSVICENI ZNAKU NA NOVE POZICI
                 545 ;       (V PRIPADE MUX=7 JE DISPLEJ NEAKTIVNI,
                 546 ;        NA HODNOTE DAT ZAPSANYCH NA BRANU B NEZALEZI)
0021 FA          547         MOV     A,R2            ;CITAC MUX
0022 031F        548         ADD     A,#VID0-1
0024 A9          549         MOV     R1,A
0025 F1          550         MOV     A,@R1           ;DATA NA DISPLEJ
0026 37          551         CPL     A
0027 B902        552         MOV     R1,#P55B
0029 91          553         MOVX    @R1,A           ;ROZSVITI DISPLEJ
                 554 ;
                 555 ;       CTENI STAVU TLACITEK
                 556 ;
002A 0A          557 TIM3:   IN      A,P2
002B 37          558         CPL     A
002C 53E0        559         ANL     A,#11100000B    ;MASKA LINEK P25-P27
002E 97          560         CLR     C
002F B903        561         MOV     R1,#3
0031 F7          562 TIM4:   RLC     A
0032 F640        563         JC      TIM6            ;JE TLACITKO
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

0034 E931        564         DJNZ    R1,TIM4
                 565         ; NENI TLACITKO
0036 FD          566         MOV     A,R5            ;POSLEDNI TLACITKO
0037 5307        567         ANL     A,#7
0039 DA          568         XRL     A,R2            ;NA STEJNE LINCE MUX ?
003A 965A        569         JNZ     TIM8            ;NE, NIC
                 570         ; TLACITKO BYLO PUSTENO
003C BD80        571 TIM5:   MOV     R5,#NOKEY       ;LASTKEY := NEPLATNE
003E 045A        572         JMP     TIM8
                 573         ; NASEL STISKNUTE TLACITKO
0040 963C        574 TIM6:   JNZ     TIM5            ;JSOU STISKNUTA DVE
0042 F9          575         MOV     A,R1            ;PORADI TLACITKA (1..3)
0043 E7          576         RL      A
0044 E7          577         RL      A               ;*8
0045 E7          578         RL      A
0046 4A          579         ORL     A,R2            ;JESTE CITAC MUX
                 580         ;V A JE KOD TLACITKA  9 .. 30
0047 2D          581         XCH     A,R5            ;AKTUALIZUJE LASTKEY
0048 DD          582         XRL     A,R5            ;STEJNE JAKO PREDTIM ?
0049 C64D        583         JZ      TIM7
004B BC03        584         MOV     R4,#CHVENI      ;NOVE, ZNOVU CITAT CHVENI
004D FC          585 TIM7:   MOV     A,R4
004E C65A        586         JZ      TIM8            ;JIZ DETEKOVANO
0050 EC5A        587         DJNZ    R4,TIM8         ;CITAC CHVENI
                 588         ;DETEKCE TLACITKA
0052 FD          589         MOV     A,R5            ;LASTKEY
0053 035A        590         ADD     A,#TABTL-9      ;PTR DO TABULKY
0055 A3          591         MOVP    A,@A            ;PREKODOVANI TLACITKA
0056 AE          592         MOV     R6,A            ;KOD TLACITKA
                 593         ;V R6 (KEY) OCEKAVA KOD TLACITKA HLAVNI PGM
                 594 ;TEST STISKU "STOP":
0057 D30B        595         XRL     A,#TLSTOP       ;TL. VE VYZNAMU STOP
0059 AB          596         MOV     R3,A            ;R3:=0 .. DETEKOVANO
                 597 TIM72:
                 598 ;
                 599 ; VYSTUP Z CASOVACE. OBNOVA LINEK P10 A P24
                 600 ;
005A B928        601 TIM8:   MOV     R1,#TIM128      ;CITA V CASOVACI 1.28 MS
005C 11          602         INC     @R1
                 603 ;
005D F8          604         MOV     A,R0            ;STAV P2 PRI PRERUSENI
005E 43F0        605         ORL     A,#11110000B    ;P24..P27 := 1
0060 3A          606         OUTL    P2,A            ;OBNOVA P20..P23
0061 FF          607         MOV     A,R7            ;OBNOVA ACC
0062 93          608         RETR
                 609 ;
                 610 ;       TABULKA PRO PREKODOVANI TLACITEK
                 611 ;
                 612 TABTL:
                 613 ;KLAVESNICE:            CITAC MUX =
                 614 ;          6      5      4      3      2      1      7
                 615 ;       ---------------------------------------------
                 616 ;P25 (- ! 7  ! 8  ! 9  ! 0   ! SAVE ! LOAD !HW.RES!
                 617 ;P26 (- ! 4  ! 5  ! 6  ! ACC ! PC  ! RUN  ! STEP !
                 618 ;P27 (- ! 1  ! 2  ! 3  ! MEM ! PREV ! NEXT ! END  !
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

                 619 ;       ------------------------------------------------
                 620 ;ORGANIZACE TABULKY:
                 621 ;       MUX=NEDEF  1   2   3   4   5   6   7
                 622 ;P25                .   .   .   .   .   .   X
                 623 ;P26            X   .   .   .   .   .   .   .
                 624 ;P27            X   .   .   .   .   .   .   .
                 625 ;
0063 13          626  DB       TLLOAD,TLSAVE,TL0   ,TL9   ,TL8   ,TL7   ,NOKEY
0064 12
0065 00
0066 09
0067 08
0068 07
0069 80
006A 80          627  DB NOKEY,TLRUN ,TLPC  ,TLACC ,TL6   ,TL5   ,TL4   ,TLSTEP
006B 10
006C 0F
006D 0E
006E 06
006F 05
0070 04
0071 11
0072 80          628  DB NOKEY,TLNEXT,TLPREV,TLMEM ,TL3   ,TL2   ,TL1   ,TLEND
0073 0C
0074 0A
0075 0D
0076 03
0077 02
0078 01
0079 0B
                 629 ;
                 630 ;TABULKA PRO KODOVANI VYSTUPU MULTIPLEXU NA BRANE C 8155
007A F7          631 TABMUX: DB      11110111B       ;MUX=1  (SVITI VPRAVO)
007B EF          632         DB      11101111B       ;   2          .
007C DF          633         DB      11011111B       ;   3          .
007D FE          634         DB      11111110B       ;   4          .
007E FD          635         DB      11111101B       ;   5          .
007F FB          636         DB      11111011B       ;   6  (SVITI VLEVO)
0080 FF          637         DB      11111111B       ;   7  (DISPLEJ ZHASNUTY)
                 638 ;
                 639 ;*******************************************************
                 640 ;*                                                     *
                 641 ;*              OBSLUHA  VNEJSIHO  PRERUSENI           *
                 642 ;*                                                     *
                 643 ;*******************************************************
                 644 ;
0081 A5          645 INTINT: CLR     F1
0082 B5          646         CPL     F1              ;F1 := 1
0083 15          647         DIS     I
0084 93          648         RETR
                 649         SIZECHK
0085             652+SIZE    SET     133
                 653+;
                 654+;++++++++++++++++++++++++++++++++++++
                 655+;
```

```
 LOC  OBJ          LINE        SOURCE STATEMENT

                    666          CBLK48  57
*0085               673+         ORG     133
                    677 ;
                    678 ;*****************************************************
                    679 ;*                                                   *
                    680 ;*              RESET  SYSTEMU                       *
                    681 ;*                                                   *
                    682 ;*****************************************************
                    683 ;
                    684 RESET:  ;INICIALIZACE VNITRNI 8155 (DISPLEJ)
 0085 B900          685          MOV     R1,#P55R        ;RIDICI PORT 8155
 0087 230F          686          MOV     A,#0FH          ;POVEL: A=B=C=OUT
 0089 74F5          687          CALL    WR55IO          ;VYSTUP, MAPUJE 8155 IO
 008B 19            688          INC     R1
 008C 23FF          689          MOV     A,#0FFH
                    690          ;MOVX   @R1,A           ;PA(--FF NIC !
 008E 19            691          INC     R1
 008F 91            692          MOVX    @R1,A           ;PB(--FF (ZHASNE DISP.)
 0090 19            693          INC     R1
 0091 91            694          MOVX    @R1,A           ;MULTIPLEX NEAKTIVNI
                    695 ;
                    696 ;INICIALIZACE HODNOT V INTERNI RAM
 0092 B820          697          MOV     R0,#VID0        ;NULUJE:
 0094 BB08          698          MOV     R3,#8           ;  VIDEOREGISTRY,
 0096 74EA          699          CALL    CLEAR           ;  ACC, SP
 0098 AC            700          MOV     R4,A            ;  PRIZNAK F
 0099 AE            701          MOV     R6,A            ;  ADRESU MEM
 009A AF            702          MOV     R7,A            ;  PC
 009B 37            703          CPL     A               ;A:=0FFH
 009C A0            704          MOV     @R0,A           ;P1D
 009D 18            705          INC     R0
 009E A0            706          MOV     @R0,A           ;P2D
                    707 ;INICIALIZACE CASOVACE, A=0FFH
 009F 62            708          MOV     T,A             ;1. TIK = 80 MIKROSEC
 00A0 D5            709          SEL     RB1
 00A1 BA01          710          MOV     R2,#1           ;CITAC MUX
 00A3 AB            711          MOV     R3,A            ;DETEKCE "STOP"
                    712 ;;       MOV     R5,A            ;LAST KEY := 0FFH
                    713                                  ;(NENI NUTNE)
 00A4 AE            714          MOV     R6,A            ;KEY := 0FFH
 00A5 C5            715          SEL     RB0
                    716          ; ODSKOK DO EXTERNI ROM, JE-LI PRIPOJENA
 00A6 56AA          717          JT1     INTRES          ;POKRACUJE ZDE
 00A8 8400          718          JMP     EXTRES          ;EXTERNAL RESET
                    719 ;
                    720 ;        RESET SYSTEMU "PETR", R0="OBECNA" HODNOTA
                    721 ;
                    722 INTRES:
                    723 ;(PO RESETU JE F0=0, F1=0)
                    724 ;TEST MAME-LI JEDNU NEBO DVE 8155
 00AA 23F5          725          MOV     A,#NOT (P55IOM+R55CE1)
 00AC 3A            726          OUTL    P2,A            ;MAPUJE POUZE DRUHOU 8155
 00AD F8            727          MOV     A,R0
 00AE 91            728          MOVX    @R1,A           ;ZAPISE NEJAKOU HODNOTU
 00AF 81            729          MOVX    A,@R1           ;A PRECTE JI ZPET
```

```
LOC  OBJ          LINE        SOURCE STATEMENT

00B0 D8            730          XRL     A,R0            ;PRECETL TOTEZ ?
00B1 96B4          731          JNZ     RES3            ;NE -) NENI DRUHA 8155
00B3 95            732          CPL     F0              ;MAME OBE 8155 !
                   733 RES3:    ;NULUJE RAM 8155
00B4 9AF1          734          ANL     P2,#NOT (P55IOM+P55CE0+P55CE1)
00B6 27            735          CLR     A          ;MAPUJE OBE 8155, PAMET
00B7 A9            736          MOV     R1,A
00B8 91            737 RES1:    MOVX    @R1,A
00B9 E9B8          738          DJNZ    R1,RES1
                   739 ;RESET PROVEDEN
00BB 55            740          STRT    T
00BC 2415          741          JMP     PRIKAZ          ;SKOK DO RIDICI SMYCKY
                   742          ;
                   743          SIZECHK
0039               746+SIZE     SET     57
                   747+;
                   748+;++++++++++++++++++++++++++++++++++
                   749+;
                   760          CBLK48  256
0100               772+         ORG     256
                   776 ;
                   777 ;TEST PRIPUSTNOSTI ADRESY V A, NICI R2
0100 B604          778 TSADR:   JF0     TSOK            ;F0 = MAME DVE 8155
0102 F205          779          JB7     TSERR
0104 83            780 TSOK:    RET
0105 BA02          781 TSERR:   MOV     R2,#ERR2
                   782 ;
                   783 ;*************************************************************
                   784 ;*                                                           *
                   785 ;*                OBSLUHA  CHYBY                             *
                   786 ;*                                                           *
                   787 ;*************************************************************
                   788 ;
                   789 ; CISLO CHYBY V R2
0107 BE00          790 ERRORO:  MOV     R6,#0
0109 25            791 ERROR:   EN      TCNTI
010A 74E6          792          CALL    CLRD5           ;MAZE DISPLEJ
010C B06E          793          MOV     @R0,#ZZE        ;'E' VLEVO
010E FA            794          MOV     A,R2
010F 74D2          795          CALL    ZOBR3           ;ZOBRAZI CISLO CHYBY
                   796          ; CEKA NA STISK END
0111 74C2          797 CEKEND:  CALL    CTICIS
0113 9611          798          JNZ     CEKEND
                   799 ;
                   800 ;*************************************************************
                   801 ;*                                                           *
                   802 ;*                RIDICI  SMYCKA  PRIKAZU                    *
                   803 ;*                                                           *
                   804 ;*************************************************************
                   805 ;
0115 25            806 PRIKAZ:  EN      TCNTI
0116 74E6          807          CALL    CLRD5           ;MAZE DISPLEJ
0118 B06C          808          MOV     @R0,#ZZC        ;'C' VLEVO
011A 74BD          809 PRIK1:   CALL    CTIKL1          ;CEKA NA KLAVESU
011C 03F3          810          ADD     A,#-TLMEM       ;CY IFF PRIKAZ
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

011E E61A         811          JNC     PRIK1
0120 0325         812          ADD     A,#LOW TAB1
0122 B825         813          MOV     R0,#VID5          ;LEVY VIDEOREGISTR
0124 B3           814          JMPP    @A                ;ROZSKOK PODLE PRIKAZU
                  815 ;
                  816 ;        TABULKA PRIKAZU
                  817 ;
                  818 ;PORADI MUSI SOUHLASIT S PORADIM KODU KLAVES
0125 3F           819 TAB1:    DB      LOW XMEM,LOW XACC,LOW XPC
0126 60
0127 6F
0128 2C           820          DB      LOW XRUN,LOW XSTEP
0129 37
012A 9A           821          DB      LOW MGSAVE,LOW MGLOAD
012B D1
                  822 ;
                  823 ;*********************************************************
                  824 ;*                                                       *
                  825 ;*              PRIKAZY  RUN, STEP                       *
                  826 ;*                                                       *
                  827 ;*********************************************************
                  828 ;
012C B0E2         829 XRUN:    MOV     @R0,#ZRUN         ;ZPRAVA NA DISPLEJ
012E A5           830          CLR     F1                ;RUSI PRIJATE PRERUSENI
012F 541B         831 XR2:     CALL    INSTR
0131 B81B         832          MOV     R0,#STOP?
0133 14FC         833          CALL    CTIKL3            ;TEST DETEKCE TL. STOP
0135 962F         834          JNZ     XR2
                  835 ;BREAK:            ;DO REZIMU "STEP"
                  836 ;
0137 541B         837 XSTEP:   CALL    INSTR             ;PROVEDE 1 INSTRUKCI
0139 74CD         838          CALL    ZOBRPC            ;ZOBRAZI PC
013B 241A         839          JMP     PRIK1             ;CEKA NA PRIKAZ,
                  840                                    ;  ANIZ BY MAZAL DISPLEJ
                  841 ;
                  842 ;*********************************************************
                  843 ;*                                                       *
                  844 ;*              PRIKAZ  MEM                              *
                  845 ;*                                                       *
                  846 ;*********************************************************
                  847 ;
                  848 ;DISPLEJ JE SMAZANY, R0=VID5
013D 74E6         849 XMEM3:   CALL    CLRD5
013F B0C2         850 XMEM:    MOV     @R0,#ZZM          ;ZOBRAZIT 'M'
0141 B906         851          MOV     R1,#6             ;ADRESA MEMORY - R6
0143 34F5         852          CALL    ZC3A              ;ZOBRAZENI + VSTUP ADRESY
                  853          ;A = OMEZOVAC
                  854          ;END JE OSETRENO, JINAK ZOBRAZI OBSAH PAMET.BUNKY
0145 FE           855          MOV     A,R6
0146 14BF         856          CALL    CVADR             ;KONVERZE NA 8155 ADRESU
                  857          ;ZOBRAZENI SOUCASNE HODNOTY
0148 81           858          MOVX    A,@R1
0149 74D2         859          CALL    ZOBR3
014B 19           860          INC     R1
014C 81           861          MOVX    A,@R1
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

014D 74D6        862          CALL    ZOBR2
                 863          ;VSTUP NOVE HODNOTY
014F BD05        864          MOV     R5,#5          ;MAX. 5 CISLIC
0151 7417        865          CALL    CISLO          ;VSTUP CISLA
0153 F65A        866          JC      XMEM4          ;BEZE ZMENY
0155 C9          867          DEC     R1
0156 91          868          MOVX    @R1,A          ;ZAPIS HODNOTY
0157 19          869          INC     R1
0158 FB          870          MOV     A,R3
0159 91          871          MOVX    @R1,A
015A 34FC        872 XMEM4:   CALL    TSTEND         ;OMEZOVAC DO A, TEST END
                 873 ;PRO PREV A NEXT: POSUN ADRESY A POKRACOVANI
                 874 ;A=-1 PRO PREV, +1 PRO NEXT
015C 6E          875          ADD     A,R6           ;POSUN ADRESY
015D AE          876          MOV     R6,A
015E 243D        877          JMP     XMEM3
                 878 ;
                 879 ;*****************************************************
                 880 ;*                                                   *
                 881 ;*      PRIKAZY  ZMENY  STAVU  PROCESORU             *
                 882 ;*                                                   *
                 883 ;*****************************************************
                 884 ;
                 885 ;       PRIKAZ ACC
                 886 ;
0160 74E6        887 XACC:    CALL    CLRD5
0162 B0DE        888          MOV     @R0,#ZZA
0164 B927        889          MOV     R1,#ACC
                 890          ; R1=ACC
0166 740A        891          CALL    ZC3            ;VSTUP NOVE HODNOTY
0168 F66B        892          JC      XACC1          ;JEN OMEZOVAC
016A A1          893          MOV     @R1,A          ;ZAPIS NOVE HODNOTY ACC
016B 34FC        894 XACC1:   CALL    TSTEND         ;OMEZOVAC DO A, TEST END
016D F288        895          JB7     XSP            ;PREV
                 896 ;
                 897 ;       PRIKAZ PC
                 898 ;
016F 74CD        899 XPC:     CALL    ZOBRPC
0171 B907        900          MOV     R1,#7          ;ADRESA PC (R7)
0173 34F5        901          CALL    ZC3A           ;ZOBRAZENI A ZMENA
                 902          ;A = OMEZOVAC
0175 F260        903          JB7     XACC           ;PREV
                 904 ;
                 905 ;       ZMENA HODNOTY FLAGU
                 906 ;
0177 74E6        907 XF:      CALL    CLRD5
0179 B04E        908          MOV     @R0,#ZZF
017B FC          909          MOV     A,R4           ;SOUCASNA HODNOTA
017C 7411        910 XF1:     CALL    ZC1            ;ZOBRAZENI + VSTUP
017E F684        911          JC      XF3            ;JEN OMEZOVAC
0180 14DE        912          CALL    T01            ;TEST A=0..1, BIT DO CARRY
0182 F7          913          RLC     A
0183 AC          914          MOV     R4,A           ;NOVA HODNOTA
                 915 XF2:
0184 34FC        916 XF3:     CALL    TSTEND         ;OMEZOVAC DO A, TEST END
```

```
LOC  OBJ          LINE        SOURCE STATEMENT

0186 F26F          917         JB7     XPC             ;PREV
                   918 ;
                   919 ;       ZMENA STACK POINTERU
                   920 ;
0188 74E6          921 XSP:    CALL    CLRD5
018A B0AE          922         MOV     @R0,#ZZS
018C B926          923         MOV     R1,#SP
018E F1            924         MOV     A,@R1
018F 7411          925         CALL    ZC1             ;ZOBRAZENI + VSTUP
0191 F694          926         JC      XSP1            ;BEZE ZMENY
0193 A1            927         MOV     @R1,A           ;ZMENA HODNOTY
0194 34FC          928 XSP1:   CALL    TSTEND          ;OMEZOVAC DO A,TEST END
0196 F277          929         JB7     XF              ;PREV
0198 2460          930         JMP     XACC            ;NEXT
                   931 ;
                   932 ;*****************************************************
                   933 ;*                                                   *
                   934 ;*              PRIKAZY  SAVE  A  LOAD               *
                   935 ;*                                                   *
                   936 ;*****************************************************
                   937 ;
                   938 ;STRUKTURA ZAZNAMU:
                   939 ;  - ZAVADECI TON (BITY 0)
                   940 ;  - SYNCHRONZACNI BITY 1,0,0
                   941 ;  - SYNHRONIZACNI BYTE 'Y' (01011001B)
                   942 ;  - DELKA ZAZNAMU (POCET BYTU V DATOVE CASTI), 2 BYTY
                   943 ;                 (NEJPRVE H-BYTE, POTOM L-BYTE)
                   944 ;  ... VLASTNI DATA ...
                   945 ;  - KONTROLNI SOUCET  1 BYTE
                   946 ;
                   947 ;POUZITI REGISTRU - PLATI PRO SAVE I LOAD
                   948 ;  R0 .. CITAC PRES OBVODY 8155
                   949 ;  R1 .. ADRESA V RAMCI OBVODU 8155
                   950 ;  R2 .. KONTROLNI SOUCET
                   951 ;  R3 .. VYSILANY BYTE (PPRG. WBYTE)
                   952 ;  R4 .. CITAC BITU V BYTU (PPRG. WBYTE)
                   953 ;        POLARITA VSTUPNIHO SIGNALU (PPRG. SYNCHRO, RBIT)
                   954 ;  R5 .. CITAC PRODLEVY (PPRG. WBIT,RBIT)
                   955 ;  R6, R7 NEMENI
                   956 ;PRI NAVRATU DOSADI DO R4 (FLAG F PROCESORU "PETR") 0
                   957 ;
                   958 ;       ZAPIS  NA  MAGNETOFON  ( SAVE )
                   959 ;
019A 74F0          960 MGSAVE: CALL    ZHASNI          ;ZHASNE DISPLEJ
                   961 ;UVODNI TON = 256 BITU 0
019C BC00          962         MOV     R4,#0
019E 7482          963 MGSV1:  CALL    WBIT0           ;ZAZNAM BITU 0
01A0 EC9E          964         DJNZ    R4,MGSV1
                   965 ;SYNCHRONIZACNI BITY: 1,0,0
01A2 2304          966         MOV     A,#100B
01A4 7492          967         CALL    WBYTE
                   968 ;INICIALIZACE NAHRAVACI ADRESY A KONTROLNIHO SOUCTU
01A6 27            969         CLR     A
01A7 A9            970         MOV     R1,A            ;NAHRAVACI ADRESA
01A8 AA            971         MOV     R2,A            ;KONTROLNI SOUCET
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

                 972 ;SYNCHRONIZACNI ZNAK & DRUH NAHRAVKY
01A9 2359        973         MOV     A,#'Y'
01AB 7492        974         CALL    WBYTE
                 975 ;ZJISTI ROZSAH OPERACNI PAMETI POCITACE "PETR"
                 976 ;VYSLE DELKU ZAZNAMU V BYTECH
01AD 2301        977         MOV     A,#1
01AF B6B2        978         JF0     MGSV2
01B1 07          979         DEC     A
                 980 MGSV2:  ;A=0 PRO JEDNU 8155, =1 PRO DVE 8155
01B2 A8          981         MOV     R0,A
01B3 7492        982         CALL    WBYTE           ;H-BYTE DELKY
01B5 F8          983         MOV     A,R0
01B6 67          984         RRC     A
01B7 A7          985         CPL     C
01B8 67          986         RRC     A
01B9 7492        987         CALL    WBYTE           ;L-BYTE DELKY
01BB 18          988         INC     R0
                 989 ;R0 = POCET OBVODU 8155 (1 NEBO 2)
01BC 23F3        990         MOV     A,#NOT (P55IOM+P55CE0)
01BE 3A          991 MGSV3:  OUTL    P2,A            ;MAPUJE 8155
01BF 81          992 MGSV4:  MOVX    A,@R1
01C0 7492        993         CALL    WBYTE
01C2 E9BF        994         DJNZ    R1,MGSV4
                 995 ;CYKLUS PRES OBVODY 8155
01C4 23F5        996         MOV     A,#NOT (P55IOM+P55CE1)
01C6 E8BE        997         DJNZ    R0,MGSV3
                 998 ;ZAPIS KONTROLNIHO SOUCTU
01C8 FA          999         MOV     A,R2
01C9 37         1000         CPL     A
01CA 17         1001         INC     A
01CB 7492       1002         CALL    WBYTE           ;ZAPIS CHKSUM
                1003 ;PRO JISTOTU ZAVERECNY TON - BYTE 00H
01CD 7491       1004         CALL    WBYTE0
01CF 24EC       1005         JMP     MGEND           ;R2=0. DULEZITE !
                1006 ;
                1007 ;       CTENI  Z  MAGNETOFONU  ( LOAD )
                1008 ;
01D1 74F0       1009 MGLOAD: CALL    ZHASNI
01D3 B859       1010         MOV     R0,#'Y'
01D5 7465       1011         CALL    SYNCHRO         ;SYNCHRONIZACE
01D7 A9         1012         MOV     R1,A            ;A:=0
                1013 ;
01D8 74AF       1014         CALL    RBYTE           ;H-BYTE DELKY, 0 NEBO 1
01DA 17         1015         INC     A
01DB A8         1016         MOV     R0,A            ;POCET OBVODU 8155
01DC 74AF       1017         CALL    RBYTE           ;L-BYTE DELKY,
                1018                                 ;  (ZDE BEZVYZNAMNY)
01DE 23F3       1019         MOV     A,#NOT (P55IOM+P55CE0)
                1020 ;NACTENI OBSAHU PAMETI SYSTEMU
01E0 3A         1021 MGLD1:  OUTL    P2,A            ;MAPUJE 8155
01E1 74AF       1022 MGLD2:  CALL    RBYTE           ;CTE 1 BYTE
01E3 91         1023         MOVX    @R1,A
01E4 E9E1       1024         DJNZ    R1,MGLD2
                1025 ;
01E6 23F5       1026         MOV     A,#NOT (P55IOM+P55CE1)
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

01E8 E8E0       1027        DJNZ    R0,MGLD1        ;CYKLUS PRES OBVODY 8155
                1028 ;ZKONTROLUJE CHKSUM
01EA 74AF       1029        CALL    RBYTE           ;KONTROLNI SOUCET
01EC FA         1030 MGEND: MOV     A,R2
01ED BC00       1031        MOV     R4,#0           ;PRIZNAK F := 0
01EF C615       1032        JZ      PRIKAZ          ;OK
01F1 BA07       1033        MOV     R2,#ERR7
01F3 2409       1034        JMP     ERROR
                1035 ;
                1036 ;ZOBRAZENI A ZMENA HODNOTY BYTU NA ADRESE R1,
                1037 ;TEST, ZDA NOVA HODNOTA MUZE BYT ADRESOU (NE=)CHYBA)
                1038 ;V A VRATI OMEZOVAC (CISLO HO NECHA V R2)
01F5 740A       1039 ZC3A:  CALL    ZC3
01F7 F6FC       1040        JC      TSTEND          ;POUZE OMEZOVAC
01F9 3400       1041        CALL    TSADR           ;MJ. TEST ROZSAHU
01FB A1         1042        MOV     @R1,A           ;NOVA HODNOTA
01FC FA         1043 TSTEND: MOV    A,R2            ;OMEZOVAC
01FD C615       1044        JZ      PRIKAZ          ;END - UKONCENI PRIKAZU
01FF 83         1045        RET
                1046        SIZECHK
0100            1049+SIZE   SET     256
                1050+;
                1051+;++++++++++++++++++++++++++++++++++
                1052+;
                1063        CBLK48  256
0200            1080+       ORG     512
                1084 ;
                1085 ; TABULKA INSTRUKCI - MUSI BYT NA ZACATKU STRANKY !
                1086 ;
0200 59         1087 TAB2:  DB      LOW IHLT        ; 1 .. HALT
0201 D5         1088        DB      LOW IDISP       ; 2 .. DISPLAY
0202 4E         1089        DB      LOW INOP        ; 3 .. NOP, PRODLEVA
0203 4B         1090        DB      LOW ILDC        ; 4 .. ACC:=KONST
0204 40         1091        DB      LOW ILDA        ; 5 .. ACC:=MEM
0205 46         1092        DB      LOW ISTA        ; 6 .. MEM:=ACC
0206 6D         1093        DB      LOW IADD        ; 7 .. ACC:=ACC+MEM
0207 72         1094        DB      LOW ISUB        ; 8 .. ACC:=ACC-MEM
0208 6A         1095        DB      LOW IJMP        ; 9 .. JMP
0209 89         1096        DB      LOW CMPO        ;10 .. CMP ACC=MEM
020A 61         1097        DB      LOW IJF         ;11 .. JUMP ON FLAG
020B 95         1098        DB      LOW ICMP1       ;12 .. CMP ACC>MEM
020C 90         1099        DB      LOW ICMP2       ;13 .. CMP ACC<MEM
020D 7B         1100        DB      LOW INOT        ;14 .. ACC:=NOT ACC
020E 7F         1101        DB      LOW IAND        ;15 .. ACC:=ACC AND MEM
020F B9         1102        DB      LOW IP1IN       ;16 .. ACC:=P1
0210 C5         1103        DB      LOW IP1OUT      ;17 .. P1:=ACC
0211 CC         1104        DB      LOW IP2OUT      ;18 .. P2:=ACC
0212 3E         1105        DB      LOW ILDAI       ;19 .. ACC:=@MEM
0213 44         1106        DB      LOW ISTAI       ;20 .. @MEM:=ACC
0214 68         1107        DB      LOW IJMPI       ;21 .. JMP @MEM
0215 84         1108        DB      LOW IOR         ;22 .. ACC:=ACC OR MEM
0216 9C         1109        DB      LOW ICALL       ;23 .. CALL
0217 AA         1110        DB      LOW IRET        ;24 .. RET
0218 65         1111        DB      LOW IJT         ;25 .. JMP ON T
0219 EE         1112        DB      LOW IKEY        ;26 .. A:=STAV KLAVESNIC
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

021A F4         1113          DB      LOW IINT        ;27 .. INT DIS/EN
                1114          ;
001B            1115 MAXINSTR EQU     $-TAB2  ;POCET INSTRUKCI
                1116 ;
                1117 ;*******************************************************
                1118 ;*                                                     *
                1119 ;*      INTERPRET  INSTRUKCI  PROCESORU  "PETR"        *
                1120 ;*                                                     *
                1121 ;*******************************************************
                1122 ;
                1123 ;   R7=PC, R5=NOVE PC  BEHEM ZPRACOVANI INSTRUKCE
                1124 ;   R0=ACC, R1=ADRESA ADRESOVE CASTI INSTRUKCE V EXT. RAM
                1125 ;   ZACHOVAVA R6
                1126 ;
                1127 ; INSTR = PROVEDENI 1 INSTRUKCE NA ADRESE PC (R7)
                1128 ;         NEBO PRIJETI POZADAVKU NA PRERUSENI
                1129 ;
021B FF         1130 INSTR:  MOV     A,R7            ;PC
021C AD         1131         MOV     R5,A
021D 27         1132         CLR     A               ;PRO PRIPAD INT
021E 7624       1133         JF1     INST1
0220 FD         1134         MOV     A,R5            ;NENI INT
0221 1D         1135         INC     R5              ;POSUN PC
0222 4425       1136         JMP     INST2
0224 A5         1137 INST1:  CLR     F1
0225 14BF       1138 INST2:  CALL    CVADR           ;FYZICKA ADRESA DO R1
0227 19         1139         INC     R1
0228 81         1140         MOVX    A,@R1           ;OPERACNI ZNAK
0229 03E4       1141         ADD     A,#-MAXINSTR-1
022B 031B       1142         ADD     A,#MAXINSTR     ;CY IFF JE INSTRUKCE
022D BA03       1143         MOV     R2,#ERR3
022F E63C       1144         JNC     IERROR          ;CHYBA - NENI TO INSTRUKCE
0231 C9         1145         DEC     R1              ;A = KOD INSTRUKCE-1
0232 B827       1146         MOV     R0,#ACC
0234 543B       1147         CALL    INST0           ;ROZSKOK A PROVEDENI
0236 FD         1148 NEWPC:  MOV     A,R5
0237 3400       1149         CALL    TSADR           ;TEST PRIPUSTNOSTI ADRESY
0239 AF         1150         MOV     R7,A            ;NOVE PC
023A 83         1151         RET
                1152 ;
                1153 ;ROZSKOK PODLE KODU INSTRUKCE
023B B3         1154 INST0:  JMPP    @A
                1155 ;
023C 2409       1156 IERROR: JMP     ERROR
                1157 ;
                1158 ;       INTERPRETACE JEDNOTLIVYCH INSTRUKCI
                1159 ;
023E 14BE       1160 ILDAI:  CALL    LDADR
0240 14CF       1161 ILDA:   CALL    LDOPN           ;CTE OPERAND
0242 A0         1162         MOV     @R0,A
0243 83         1163         RET
                1164         ;
0244 14BE       1165 ISTAI:  CALL    LDADR
0246 14BE       1166 ISTA:   CALL    LDADR
                1167 ;;      CLR     A
```

```
LOC  OBJ          LINE,       SOURCE STATEMENT

                  1168 ;;       INC     R1
                  1169 ;;       MOVX    @R1,A           ;OPER.ZNAK:=0 (DATA)
                  1170 ;;       DEC     R1
0248 F0           1171          MOV     A,@R0
0249 91           1172          MOVX    @R1,A
024A 83           1173          RET
                  1174          ;
024B 81           1175 ILDC:    MOVX    A,@R1
024C A0           1176          MOV     @R0,A
024D 83           1177          RET
                  1178          ;
024E 81           1179 INOP:    MOVX    A,@R1
024F C658         1180          JZ      NOP0
0251 18           1181          INC     R0              ;R0 = TIM128
                  1182 ;PRODLEVA (A) * 1.28 MILISEKUND
0252 60           1183          ADD     A,@R0
0253 AA           1184          MOV     R2,A
0254 FA           1185 NOP1:    MOV     A,R2
0255 D0           1186          XRL     A,@R0
0256 9654         1187          JNZ     NOP1
0258 83           1188 NOP0:    RET
                  1189          ;
0259 74E6         1190 IHLT:    CALL    CLRD5           ;
025B B0DA         1191          MOV     @R0,#ZZH        ;'H' VLEVO NA DISPLEJ
                  1192 ;;       CALL    NEWPC           ;NOVE PC (PRO POKRACOVANI)
025D 74D1         1193          CALL    ZPC1            ;ZOBRAZI PC
025F 2411         1194          JMP     CEKEND          ;ROVNOU DO RIDICI SMYCKY
                  1195          ;(*** POSUNUJE STACK O 2 POLOZKY ***)
                  1196 ;
                  1197 ;       SKOKY
                  1198 ;
0261 FC           1199 IJF:     MOV     A,R4            ;SKOK, JE-LI FLAG
0262 966A         1200          JNZ     IJMP
0264 83           1201          RET                     ;JINAK NIC
0265 366A         1202 IJT:     JT0     IJMP            ;SKOK, JE-LI LINKA T=1
0267 83           1203          RET                     ;JINAK NIC
0268 14BE         1204 IJMPI:   CALL    LDADR
                  1205 ;NEPODMINENY SKOK
026A 81           1206 IJMP:    MOVX    A,@R1           ;CILOVA ADRESA SKOKU
026B AD           1207          MOV     R5,A            ;NOVE PC
026C 83           1208          RET
                  1209 ;
                  1210 ;       ARITMETICKE OPERACE
                  1211 ;
026D 14CB         1212 IADD:    CALL    LDOPF0
026F 60           1213          ADD     A,@R0
0270 4478         1214          JMP     IADD2
                  1215          ;
0272 14CB         1216 ISUB:    CALL    LDOPF0
0274 2A           1217          XCH     A,R2
0275 37           1218          CPL     A
0276 6A           1219          ADD     A,R2
0277 37           1220          CPL     A               ;A:=A-R2,CY,ZERO
0278 A0           1221 IADD2:   MOV     @R0,A
0279 4499         1222          JMP     IADD3           ;PRETECENI -) SET FLAG
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

                1223 ;
                1224 ;       LOGICKE OPERACE
                1225 ;
027B F0         1226 INOT:   MOV     A,@R0
027C 37         1227         CPL     A
027D A0         1228         MOV     @R0,A
027E 83         1229         RET
                1230         ;
027F 14CF       1231 IAND:   CALL    LDOPM
0281 50         1232         ANL     A,@R0
0282 A0         1233         MOV     @R0,A
0283 83         1234         RET
                1235         ;
0284 14CF       1236 IOR:    CALL    LDOPM
0286 40         1237         ORL     A,@R0
0287 A0         1238         MOV     @R0,A
0288 83         1239         RET
                1240 ;
                1241 ;       POROVNANI
                1242 ;
0289 14CB       1243 CMP0:   CALL    LDOPF0          ;POROVNANI ACC=MEM
028B D0         1244         XRL     A,@R0
028C 968F       1245         JNZ     CMP4
028E 1C         1246 CMP6:   INC     R4              ;R4!=1 .. SET FLAG
028F 83         1247 CMP4:   RET
                1248         ;
0290 14CB       1249 ICMP2:  CALL    LDOPF0          ;POROVNANI ACC<MEM
0292 2A         1250         XCH     A,R2            ;VYMENA OPERANDU
0293 4497       1251         JMP     CMP5            ;POKRACUJE JAKO ACC>MEM
                1252         ;
0295 14CB       1253 ICMP1:  CALL    LDOPF0          ;POROVNANI ACC>MEM
0297 37         1254 CMP5:   CPL     A
0298 6A         1255         ADD     A,R2            ;CARRY IFF A(MEM)<R2(ACC)
0299 F68E       1256 IADD3:  JC      CMP6            ;SET F0 + RET
029B 83         1257         RET
                1258 ;
                1259 ;       CALL & RET
                1260 ;
029C C8         1261 ICALL:  DEC     R0              ;R0=SP
029D F0         1262         MOV     A,@R0
029E D30A       1263         XRL     A,#STKMAX-STKMIN+1      ;PLNY ZASOBNIK ?
02A0 BA05       1264         MOV     R2,#ERR5        ;CISLO PRIPADNE CHYBY
02A2 C63C       1265         JZ      IERROR          ;CHYBA - PLNY ZASOBNIK
02A4 F0         1266         MOV     A,@R0
02A5 10         1267         INC     @R0             ;ZVYSI UKAZATEL SP
02A6 54B2       1268         CALL    ISTK2           ;PC DO ZASOBNIKU
02A8 446A       1269         JMP     IJMP
                1270         ;
02AA C8         1271 IRET:   DEC     R0              ;R0=SP
02AB F0         1272         MOV     A,@R0
02AC BA06       1273         MOV     R2,#ERR6
02AE C63C       1274         JZ      IERROR          ;CHYBA PRAZDNY ZASOBNIK
02B0 07         1275         DEC     A               ;SNIZI SP
02B1 A0         1276         MOV     @R0,A
                1277 ISTK2:  ;VYMENA HODNOTY PC A VRCHOLU ZASOBNIKU
```

```
LOC  OBJ          LINE        SOURCE STATEMENT

                  1278          ;A = REL. ADRESA VRCHOLU ZASOBNIKU
02B2 0331         1279          ADD     A,#STKMIN
02B4 A8           1280          MOV     R0,A            ;ADRESA VRCHOLU ZASOBNIKU
02B5 FD           1281          MOV     A,R5            ;PC
02B6 20           1282          XCH     A,@R0           ;VYMENA
02B7 AD           1283          MOV     R5,A            ;ADRESA ZE ZASOBNIKU
02B8 83           1284          RET
                  1285 ;
                  1286 ;        IN ,  OUT
                  1287 ;
02B9 14D3         1288 IP1IN:   CALL    LT08            ;R2:=OPERAND, CARRY
02BB 09           1289          IN      A,P1
02BC F6C3         1290          JC      STORA           ;CELY BYTE
02BE F7           1291 IP1B2:   RLC     A
02BF EABE         1292          DJNZ    R2,IP1B2
02C1 27           1293          CLR     A               ;PRECTENY BIT V CARRY
02C2 F7           1294          RLC     A               ;A = PRECTENY BIT
02C3 A0           1295 STORA:   MOV     @R0,A
02C4 83           1296          RET
                  1297          ;
02C5 BA29         1298 IP1OUT:  MOV     R2,#P1D         ;OBRAZ DAT NA PORTU
02C7 14E6         1299          CALL    POUT            ;A=NOVA DATA, R1:=R2
02C9 A1           1300          MOV     @R1,A           ;ZAPIS OBRAZU DAT
02CA 39           1301          OUTL    P1,A            ;DATA NA PORT
02CB 83           1302          RET
                  1303          ;
02CC BA2A         1304 IP2OUT:  MOV     R2,#P2D         ;OBRAZ DAT NA PORTU
02CE 14E6         1305          CALL    POUT            ;A=NOVA DATA, R1:=R2
02D0 A1           1306          MOV     @R1,A           ;ZAPIS OBRAZU DAT
02D1 B901         1307          MOV     R1,#P55A
02D3 64F5         1308          JMP     WR55IO          ;ZAPIS DAT NA PORT
                  1309 ;
                  1310 ;        ZOBRAZENI
                  1311 ;
02D5 81           1312 IDISP:   MOVX    A,@R1           ;OPERAND
02D6 C6E7         1313          JZ      IZOBR1          ;ZOBRAZIT ACC
02D8 17           1314          INC     A               ;TEST OPERAND = 255
02D9 C6EA         1315          JZ      IZOBR0          ;SMAZAT DISPLEJ
                  1316 ;PRIMY ZAPIS DO VIDEOREGISTRU (DISP 1..6)
02DB 03F8         1317          ADD     A,#-8           ;TEST OPERAND = 1..6
02DD BA04         1318          MOV     R2,#ERR4
02DF F63C         1319          JC      IERROR          ;CHYBNY OPERAND
02E1 0326         1320          ADD     A,#VID0+6
02E3 A9           1321          MOV     R1,A
02E4 F0           1322          MOV     A,@R0           ;OBSAH ACC
02E5 A1           1323          MOV     @R1,A           ;PRIMY ZAPIS DO VIDEOREG.
02E6 83           1324          RET
                  1325 ;
                  1326 ;ZOBRAZENI ACC DEKADICKY (DISP 0)
02E7 F0           1327 IZOBR1:  MOV     A,@R0           ;OBSAH ACC
02E8 64D2         1328          JMP     ZOBR3
                  1329 ;
                  1330 ;VYMAZ DISPLEJE (DISP 255)
                  1331 CLDISP:
02EA 74E6         1332 IZOBR0:  CALL    CLRD5           ;VYMAZ PETI PRAVYCH POZIC
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

02EC A0         1333        MOV     @R0,A           ;VYMAZ LEVE POZICE
02ED 83         1334        RET
                1335 ;
                1336 ;      CTENI STAVU KLAVESNICE Z PROGRAMU
                1337 ;
02EE 14FA       1338 IKEY:  CALL    CTIKL
02F0 B827       1339        MOV     R0,#ACC
02F2 A0         1340        MOV     @R0,A           ;ACC := KEY
02F3 83         1341        RET
                1342 ;
                1343 ;      INTERRUPT  DIS / EN
                1344 ;
02F4 81         1345 IINT:  MOVX    A,@R1
02F5 14DE       1346        CALL    T01             ;TEST 0..1
02F7 15         1347        DIS     I
02F8 E6FB       1348        JNC     IINT0
02FA 05         1349        EN      I
02FB 83         1350 IINT0: RET
                1351        SIZECHK
00FC            1354+SIZE   SET     252
                1355+;
                1356+;++++++++++++++++++++++++++++++++++++
                1357+;
                1368 ;
                1369 ;*************************************************************
                1370 ;*                                                           *
                1371 ;*      PODRPOGRAMY  PRO  INTERPRETACI  INSTRUKCI            *
                1372 ;*                                                           *
                1373 ;*************************************************************
                1374 ;
                1375        CBLK48  13
00BE            1382+       ORG     190
                1386 ;
                1387 ;*************************************************************
                1388 ;*                                                           *
                1389 ;*      TRANSFORMACE  LOG. ADRESA --> FYZ. ADRESA            *
                1390 ;*                                                           *
                1391 ;*************************************************************
                1392 ;
                1393 ;VSTUP:  A=LOGICKA ADRESA BUNKY PAMETI
                1394 ;VYSTUP: PRIMAPUJE PAMET ODPOVIDAJICIHO OBVODU 8155,
                1395 ;       V R1 VRATI ADRESU UVNITR PRIMAPOVANE STRANKY
                1396 ; NICI REGISTRY A,R2
                1397 ;
                1398 ;TRANSFORMACNI FUNKCE:
                1399 ;VNITRNI 8155:  FYZ. ADR. = 2*LOG.ADRESA
                1400 ;VNEJSI 8155:   FYZ. ADR. = 2*LOG.ADRESA+1
                1401 ;
00BE 81         1402 LDADR: MOVX    A,@R1
00BF 3400       1403 CVADR: CALL    TSADR           ;TEST PRIPUSTNOSTI
00C1 B9F5       1404        MOV     R1,#NOT (P55IOM+P55CE1) ;PRO ADR >=128
00C3 F2C7       1405        JB7     CVADO
00C5 B9F3       1406        MOV     R1,#NOT (P55IOM+P55CE0) ;PRO ADR <128
00C7 E7         1407 CVADO: RL      A
00C8 29         1408        XCH     A,R1
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

00C9 3A         1409        OUTL    P2,A            ;MAPUJE SPRAVNOU 8155
00CA 83         1410        RET
                1411        SIZECHK
000D            1414+SIZE   SET     13
                1415+;
                1416+;++++++++++++++++++++++++++++++++++
                1417+;
                1428        CBLK48  27
00CB            1435+       ORG     203
                1439 ;
                1440 ;*********************************************************
                1441 ;*                                                       *
                1442 ;*              CTENI  OPERANDU  INSTRUKCI               *
                1443 ;*                                                       *
                1444 ;*********************************************************
                1445 ;
                1446 ;VYSTUP:  A=HODNOTA OPERANDU MEM
                1447 ;PRO LDOPF0 NAVIC: FLAG := 0, V R2 VRATI ACC
                1448 ;
00CB BC00       1449 LDOPF0: MOV     R4,#0           ;NULUJE UZIV. FLAG
00CD F0         1450        MOV     A,@R0
00CE AA         1451        MOV     R2,A            ;ACC
                1452 ;
00CF 14BE       1453 LDOPM:  CALL    LDADR           ;ADRESA OPERANDU DO R1
                1454 ;;      INC     R1              ;CISLO PRIPADNE CHYBY
                1455 ;;      MOVX    A,@R1           ;OPER. ZNAK
                1456 ;;      JNZ     AERROR          ;NENULOVY => CHYBA
                1457 ;;      DEC     R1
00D1 81         1458        MOVX    A,@R1
00D2 83         1459        RET
                1460 ;
                1461 ;*********************************************************
                1462 ;*                                                       *
                1463 ;*              VYBER  OPERANDU  A  TEST 0..8            *
                1464 ;*                                                       *
                1465 ;*********************************************************
                1466 ;
                1467 ; VRATI CARRY IFF OPERAND JE 8
                1468 ; OPERAND VRATI V R2 8,...,1,0FFH
                1469 ; A:=STARE R2
00D3 81         1470 LT08:   MOVX    A,@R1
00D4 03F7       1471        ADD     A,#-9
00D6 F6E2       1472        JC      LERROR
                1473 ;TED JE A = -9..-1
00D8 0301       1474        ADD     A,#1            ;CARRY PRO OP.=8
00DA 37         1475        CPL     A
00DB 17         1476        INC     A
00DC 2A         1477        XCH     A,R2
00DD 83         1478 TO1RET: RET
                1479 ;
                1480 ;*********************************************************
                1481 ;*                                                       *
                1482 ;*              TEST,  ZDA  A = 0 NEBO 1                 *
                1483 ;*                                                       *
                1484 ;*********************************************************
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

             1485 ;
             1486 ;NENI-LI, OHLASI CHYBU 4
             1487 ;PRI NAVRATU:  CARRY := BIT A.0, A VYNULUJE
00DE 97      1488 T01:    CLR    C
00DF 67      1489         RRC    A                ;BIT A.0 V CARRY
00E0 C6DD    1490         JZ     T01RET           ;OSTATNI BITY A NULOVE
             1491 ;
             1492 ;SPOLECNA CHYBA - OPERAND MIMO POVOLENY ROZSAH
00E2 BA04    1493 LERROR: MOV    R2,#ERR4         ;CISLO CHYBY
00E4 2409    1494 AERROR: JMP    ERROR
             1495         SIZECHK
001B         1498+SIZE    SET    27
             1499+;
             1500+;++++++++++++++++++++++++++++++++++++
             1501+;
             1512         CBLK48 20
00E6         1519+        ORG    230
             1523 ;
             1524 ;*******************************************************
             1525 ;*                                                     *
             1526 ;*      PODPROGRAM  PRO  PROVEDENI  OPERACI  OUT       *
             1527 ;*                                                     *
             1528 ;*******************************************************
             1529 ;
             1530 ;   VSTUP: R0=ACC, R2 SMERNIK NA OBRAZ DAT NA PORTU
             1531 ;   R1 = SMERNIK NA OPERAND INSTRUKCE (V EXT. RAM)
             1532 ;   VYSTUP: V A DATA K VYSLANI NA PORT
             1533 ;
00E6 14D3    1534 POUT:   CALL   LT08             ;CTE OPERAND, TEST 0..8
00E8 A9      1535         MOV    R1,A             ;ADRESA OBRAZU DAT NA PORTU
00E9 F0      1536         MOV    A,@R0            ;LOAD ACC
00EA F6F9    1537         JC     POUT8            ;CELY PORT
00EC 14DE    1538         CALL   T01              ;TEST 0..1, CY:=BIT
00EE 2301    1539         MOV    A,#1
00F0 77      1540 POUT1:  RR     A
00F1 EAF0    1541         DJNZ   R2,POUT1
             1542 ;V A JE BIT 1 V ODPOVIDAJICI POLOZE
00F3 E6F7    1543         JNC    POUT2            ;NULOVAT BIT
00F5 41      1544         ORL    A,@R1            ;JEDNICKUJE BIT
00F6 83      1545         RET
00F7 37      1546 POUT2:  CPL    A
00F8 51      1547         ANL    A,@R1            ;NULUJE BIT
00F9 83      1548 POUT8:  RET
             1549         SIZECHK
0014         1552+SIZE    SET    20
             1553+;
             1554+;++++++++++++++++++++++++++++++++++++
             1555+;
             1566         CBLK48 7
030A         1612+        ORG    778
             1616 ;
             1617 ;*******************************************************
             1618 ;*                                                     *
             1619 ;*      ZOBRAZENI  BYTU,  VSTUP  NOVE  HODNOTY         *
             1620 ;*                                                     *
```

```
LOC  OBJ          LINE        SOURCE STATEMENT

                  1621 ;*******************************************************
                  1622 ;
                  1623 ;ADRESA ZOBRAZOVANEHO BYTU JE V R1
                  1624 ;
                  1625 ;POUZE ZOBRAZENI A VSTUP HODNOTY
030A F1           1626 ZC3:     MOV     A,@R1
030B 74D2         1627          CALL    ZOBR3
030D BD03         1628          MOV     R5,#3           ;3 CISLICE
030F 6417         1629          JMP     CISLO           ;VSTUP CISLA A NAVRAT
                  1630          SIZECHK
0007              1633+SIZE     SET     7
                  1634+;
                  1635+;++++++++++++++++++++++++++++++++++
                  1636+;
                  1647          CBLK48  80
0311              1693+         ORG     785
                  1697 ;
                  1698 ;*******************************************************
                  1699 ;*                                                     *
                  1700 ;*      ZOBRAZENI  A  VSTUP  1  CISLICE                *
                  1701 ;*                                                     *
                  1702 ;*******************************************************
                  1703 ;
                  1704 ;HODNOTA K ZOBRAZENI JE V A
0311 B820         1705 ZC1:     MOV     R0,#VIDO
0313 74D8         1706          CALL    ZOBR1           ;ZOBRAZENI HODNOTY
0315 BD01         1707          MOV     R5,#1           ;POCET VSTUPNICH CISLIC
                  1708          ;;JMP   CISLO
                  1709 ;
                  1710 ;*******************************************************
                  1711 ;*                                                     *
                  1712 ;*              CTENI  CISLA                           *
                  1713 ;*                                                     *
                  1714 ;*******************************************************
                  1715 ;
                  1716 ;    VSTUP:  R5 - POCET DOVOLENYCH CISLIC
                  1717 ;    VYSTUP: R2=OMEZOVAC ZA CISLEM,
                  1718 ;                (-1...PREV, 0...END, 1...NEXT)
                  1719 ;            CY IFF NEBYLA ZADANA HODNOTA
                  1720 ;    ZNICI R0,2,3,5. ZACHOVA R1,4,6,7
                  1721 ;    PRECTENOU HODNOTU VRATI V A(LOW),R3(HIGH)
                  1722 ;
0317 B82B         1723 CISLO:   MOV     R0,#BUFF
0319 74E8         1724          CALL    CLR5            ;NULUJE BUFFER CISLIC
031B 74C2         1725          CALL    CIICIS          ;CTE KLAVESU
031D F658         1726          JC      CISL4           ;JEN OMEZOVAC
                  1727          ;MAZE DISPLEJ KROME LEVEHO MISTA
031F 74E6         1728          CALL    CLRD5
                  1729          ;VYPIS NAPOVEDNE TECKY
0321 2320         1730          MOV     A,#VIDO
0323 6D           1731          ADD     A,R5
0324 A8           1732          MOV     R0,A            ;PTR NA POZICI TECKY
0325 F0           1733          MOV     A,@R0
0326 4301'        1734          ORL     A,#ZTECKA
0328 A0           1735          MOV     @R0,A           ;PUVODNI OBSAH + TECKA
```

```
LOC  OBJ       LINE       SOURCE STATEMENT

             1736       ;
0329 FA      1737       MOV     A,R2            ;ZNAK+(TLO-TLEND)
032A 030B    1738 CISL2: ADD     A,#TLEND-TLO    ;HODNOTA CISLICE
032C AA      1739       MOV     R2,A            ;ODLOZI HODNOTU
032D E3      1740       MOVP3   A,@A            ;ZOBRAZI CISLICI
032E B820    1741       MOV     R0,#VIDO        ;A POSUNE PREDCHOZI
0330 745D    1742       CALL    POSUN
0332 FA      1743       MOV     A,R2            ;NOVA CISLICE
0333 B82B    1744       MOV     R0,#BUFF        ;ZAPIS DO BUFFERU
0335 745D    1745       CALL    POSUN           ;A POSUN PREDCHOZICH
0337 74C2    1746       CALL    CTICIS          ;DALSI KLAVESA
0339 E62A    1747       JNC     CISL2           ;CYKLUS PRES CISLICE
             1748       ; CISLO NACTENO, KONVERZE DEC-->BIN
033B B830    1749       MOV     R0,#BUFF+5      ;PTR NA CISLICE
033D BB02    1750       MOV     R5,#2           ;POCET CISLIC HI
033F 7444    1751       CALL    CISL6           ;PREVOD DEC-->BIN
0341 AB      1752       MOV     R3,A            ;ULOZI HI
0342 BD03    1753       MOV     R5,#3           ;POCET CISLIC LOW
             1754       ; PREVOD DEC-->BIN. R0 UKAZATEL NA HORNI CISLICI,
             1755       ;                   R5 - POCET CISLIC.
             1756       ; ZNICI @R0,R5. VYSLEDEK DA DO REG. A
0344 27      1757 CISL6: CLR     A
0345 F259    1758 CISL5: JB7     CISL7           ;CHYBA - PRETECENI
0347 E7      1759       RL      A               ;*2
0348 A0      1760       MOV     @R0,A
0349 F259    1761       JB7     CISL7           ;PRETECENI
034B E7      1762       RL      A               ;*4
034C F259    1763       JB7     CISL7           ;PRETECENI
034E E7      1764       RL      A               ;*8
034F 60      1765       ADD     A,@R0           ;*10
0350 F659    1766       JC      CISL7           ;PRETECENI
0352 C8      1767       DEC     R0
0353 60      1768       ADD     A,@R0           ;+NOVA CISLICE
0354 F659    1769       JC      CISL7           ;PRETECENI
0356 ED45    1770       DJNZ    R5,CISL5        ;CYKLUS PRES CISLICE
0358 83      1771 CISL4: RET
             1772       ;
             1773       ; PRETECENI PRI PREVODU DEC-->BIN
0359 BA01    1774 CISL7: MOV     R2,#ERR1
035B 2409    1775       JMP     ERROR
             1776       ;
             1777       SIZECHK
004C         1780+SIZE  SET     76
             1781+;
             1782+;++++++++++++++++++++++++++++++++++++
             1783+;
             1794       CBLK48  8
035D         1840+      ORG     861
             1844 ;
             1845 ;     PODPROGRAM PRO CTENI CISLA
             1846 ;
             1847 ;POSUNE R5 BYTU POCINAJE @R0 O 1 MISTO SMEREM
             1848 ;K VYSSIM ADRESAM. DO 1.BYTU ZAPISE A.
             1849 ;ZNICI R2, POSUNE R0, OSTATNI REGISTRY ZACHOVA
035D 2D      1850 POSUN: XCH     A,R5
```

```
LOC  OBJ        LINE        SOURCE STATEMENT

035E AB         1851        MOV     R3,A            ;KOPIE DELKY
035F 2D         1852        XCH     A,R5
0360 20         1853 POS1:  XCH     A,@R0
0361 18         1854        INC     R0
0362 EB60       1855        DJNZ    R3,POS1
0364 83         1856        RET
                1857        SIZECHK
0008            1860+SIZE   SET     8
                1861+;
                1862+;++++++++++++++++++++++++++++++++++++
                1863+;
                1874 ;
                1875 ;*********************************************************
                1876 ;*                                                       *
                1877 ;*      PODPROGRAMY  PRO  KOMUNIKACI  S  PERIFERIEMI     *
                1878 ;*                                                       *
                1879 ;*********************************************************
                1880 ;
                1881 ;
                1882 ;*********************************************************
                1883 ;*                                                       *
                1884 ;*                 M A G N E T O F O N                   *
                1885 ;*                                                       *
                1886 ;*********************************************************
                1887 $SAVE NOLIST
                2315 $RESTORE
                2316 ;
                2317 ;*********************************************************
                2318 ;*                                                       *
                2319 ;*                 K L A V E S N I C E                   *
                2320 ;*                                                       *
                2321 ;*********************************************************
                2322 ;
                2323        CBLK48  6
00FA            2330+       ORG     250
                2334 ;
                2335 ; CTENI STAVU KLAVESNICE, JE-LI ZNAK, JEHO PREVZETI
                2336 ;
00FA B81E       2337 CTIKL:  MOV     R0,#KEY
00FC 2380       2338 CTIKL3: MOV     A,#NOKEY
00FE 20         2339        XCH     A,@R0
00FF 83         2340        RET
                2341        SIZECHK
0006            2344+SIZE   SET     6
                2345+;
                2346+;++++++++++++++++++++++++++++++++++++
                2347+;
                2358        CBLK48  5
03BD            2404+       ORG     957
                2408 ;
                2409 ;      CTENI ZNAKU Z KLAVESNICE S CEKANIM
                2410 ;      KOD PRECTENEHO TLACITKA VRATI V A.
                2411 ;
03BD 14FA       2412 CTIKL1: CALL    CTIKL
03BF F2BD       2413        JB7     CTIKL1          ;CEKACI SMYCKA
```

```
LOC  OBJ          LINE        SOURCE STATEMENT

03C1 83           2414          RET
                  2415          SIZECHK
0005              2418+SIZE     SET     5
                  2419+;
                  2420+;++++++++++++++++++++++++++++++++++++
                  2421+;
                  2432          CBLK48  11
03C2              2478+         ORG     962
                  2482 ;
                  2483 ; CTENI ZNAKU Z KLAVESNICE PRO POTREBY CTENI CISLA.
                  2484 ;
03C2 74BD         2485 CTICIS: CALL     CTIKL1          ;CEKA NA ZNAK
03C4 03F3         2486         ADD      A,#-TLMEM       ;TEST PRIKAZOVEHO TLACITKA
03C6 F6C2         2487         JC       CTICIS          ;PRIKAZY IGNORUJE
03C8 0303         2488         ADD      A,#3            ;CY IFF OMEZOVAC
03CA 07           2489         DEC      A
03CB AA           2490         MOV      R2,A
                  2491 ;VYSTUP: A=R2=PRECTENY ZNAK
                  2492 ; JE-LI TO OMEZOVAC, JE CARRY,
                  2493 ;       ZNAK = -1..PREV, 0..END, +1..NEXT.
                  2494 ; JE-LI TO CISLICE, ZNAK = HODNOTA CISLICE+(TLO-TLEND)
03CC 83           2495          RET
                  2496          SIZECHK
000B              2499+SIZE     SET     11
                  2500+;
                  2501+;++++++++++++++++++++++++++++++++++++
                  2502+;
                  2513 ;
                  2514 ;*********************************************************
                  2515 ;*                                                       *
                  2516 ;*                  Z O B R A Z O V A N I                *
                  2517 ;*                                                       *
                  2518 ;*********************************************************
                  2519 ;
                  2520          CBLK48  25
03CD              2566+         ORG     973
                  2570 ;
                  2571 ;*********************************************************
                  2572 ;*                                                       *
                  2573 ;*          VYSTUP CISEL V DESITKOVE SOUSTAVE            *
                  2574 ;*                                                       *
                  2575 ;*********************************************************
                  2576 ;
                  2577 ;       ZOBRAZENI PC
                  2578 ;
03CD 74E6         2579 ZOBRPC: CALL     CLRD5
03CF B05E         2580         MOV      @R0,#ZZP        ;'P' DO LEVEHO DISPLEJE
03D1 FF           2581 ZPC1:   MOV      A,R7            ;PC
                  2582 ;
                  2583 ; ZOBR3: ZOBRAZENI TROJICE VPRAVO NA DISPLEJI
                  2584 ; ZOBR2: ZOBRAZENI DVOJICE. NUTNO NASTAVIT R0 !
                  2585 ; ZOBR1: ZOBRAZENI 1 CISLICE. NUTNO NASTAVIT R0 !
                  2586 ;
03D2 B820         2587 ZOBR3:  MOV      R0,#VIDO        ;PTR DO VIDEOREGISTRU
03D4 74D8         2588         CALL     ZOBR1
```

```
LOC  OBJ          LINE        SOURCE STATEMENT

03D6 74D8         2589 ZOBR2:  CALL    ZOBR1
                  2590 ;
                  2591 ;  ZOBRAZI CISLICI A MOD 10; A:=A DIV 10. NICI R2
                  2592 ;
03D8 BAFF         2593 ZOBR1:  MOV     R2,#0FFH
03DA 03F6         2594 ZOBR11: ADD     A,#-10
03DC 1A           2595         INC     R2
03DD F6DA         2596         JC      ZOBR11
03DF 030A         2597         ADD     A,#10
                  2598         ;R2=A DIV 10, A=A MOD 10
03E1 E3           2599         MOVP3   A,@A
03E2 A0           2600         MOV     @R0,A           ;ZOBRAZI A MOD 10
03E3 18           2601         INC     R0
03E4 FA           2602         MOV     A,R2            ;A:=A DIV 10
03E5 83           2603         RET
                  2604         SIZECHK
0019              2607+SIZE    SET     25
                  2608+;
                  2609+;++++++++++++++++++++++++++++++++++++
                  2610+;
                  2621         CBLK48  10
03E6              2667+        ORG     998
                  2671 ;
                  2672 ;*******************************************************
                  2673 ;*                                                     *
                  2674 ;*              MAZANI  DISPLEJE                       *
                  2675 ;*                                                     *
                  2676 ;*******************************************************
                  2677 ;
03E6 B820         2678 CLRD5:  MOV     R0,#VID0
                  2679 ;NULOVANI 5 BYTU POCINAJE R0
03E8 BB05         2680 CLR5:   MOV     R3,#5
03EA 27           2681 CLEAR:  CLR     A
03EB A0           2682 CLR0:   MOV     @R0,A
03EC 18           2683         INC     R0
03ED EBEB         2684         DJNZ    R3,CLR0
03EF 83           2685         RET
                  2686         SIZECHK
000A              2689+SIZE    SET     10
                  2690+;
                  2691+;++++++++++++++++++++++++++++++++++++
                  2692+;
                  2703         CBLK48  11
03F0              2749+        ORG     1008
                  2753 ;
                  2754 ;*******************************************************
                  2755 ;*                                                     *
                  2756 ;*      ZHASNUTI DISPLEJE A ZABLOKOVANI JEHO OBSLUHY   *
                  2757 ;*                                                     *
                  2758 ;*******************************************************
                  2759 ;
03F0 35           2760 ZHASNI: DIS     TCNTI
03F1 B902         2761 ZHAS0:  MOV     R1,#P55B        ;ADRESA PORTU A (SEGMENTY)
03F3 23FF         2762         MOV     A,#0FFH         ;VSECHNY ZHASNOUT
03F5 8A0A         2763 WR55I0: ORL     P2,#P55IOM+P55CE1
```

```
 LOC  OBJ          LINE        SOURCE STATEMENT

 03F7 9AFB         2764          ANL     P2,#(NOT P55CE0);MAPUJE VNITRNI 8155, IO
 03F9 91           2765          MOVX    @R1,A           ;ZAPIS NA PORT
 03FA 83           2766          RET
                   2767          SIZECHK
 000B              2770+SIZE     SET     11
                   2771+;
                   2772+;++++++++++++++++++++++++++++++++++
                   2773+;
                   2784          END
```

USER SYMBOLS

??MIND	0020	?LENGT	000B	?SIZE	0016	?START	03F0	ACC	0027	AERROR	00E4
BUFF	002B	CBLK48	0005	CEKEJ	038E	CEKEND	0111	CHR2	03B9	CHVENI	0003
CISL2	032A	CISL4	0358	CISL5	0345	CISL6	0344	CISL7	0359	CISLO	0317
CLDISP	02EA	CLEAR	03EA	CLR0	03EB	CLR5	03E8	CLRD5	03E6	CMP0	0289
CMP4	028F	CMP5	0297	CMP6	028E	CODEBL	0004	CODEMB	0003	CTICIS	03C2
CTIKL	00FA	CTIKL1	03BD	CTIKL3	00FC	CVAD0	00C7	CVADR	00BF	DATABL	0001
DF	0000	ERR1	0001	ERR2	0002	ERR3	0003	ERR4	0004	ERR5	0005
ERR6	0006	ERR7	0007	ERROR	0109	ERROR0	0107	EXTINT	0403	EXTRES	0400
EXTTIM	0407	IADD	026D	IADD2	0278	IADD3	0299	IAND	027F	ICALL	029C
ICMP1	0295	ICMP2	0290	IDISP	02D5	IERROR	023C	IHLT	0259	IINT	02F4
IINT0	02FB	IJF	0261	IJMP	026A	IJMPI	0268	IJT	0265	IKEY	02EE
ILDA	0240	ILDAI	023E	ILDC	024B	INOP	024E	INOT	027B	INST0	023B
INST1	0224	INST2	0225	INSTR	021B	INTINT	0081	INTRES	00AA	IOR	0284
IP1B2	02BE	IP1IN	02B9	IP1OUT	02C5	IP2OUT	02CC	IRET	02AA	ISTA	0246
ISTAI	0244	ISTK2	02B2	ISUB	0272	IZOBR0	02EA	IZOBR1	02E7	K500	0062
K750	0094	KEY	001E	KP1	00FF	KTIM	00F0	LDADR	00BE	LDOPF0	00CB
LDOPM	00CF	LERROR	00E2	LOCBLK	0006	LT08	00D3	MAXINS	001B	MGBITR	0001
MGBITW	0002	MGEND	01EC	MGLD1	01E0	MGLD2	01E1	MGLOAD	01D1	MGSAVE	019A
MGSV1	019E	MGSV2	01B2	MGSV3	01BE	MGSV4	01BF	MUX7	0010	NEWPC	0236
NOKEY	0080	NOP0	0258	NOP1	0254	ORGPG0	0100	ORGPG1	0200	ORGPG2	02FC
ORGPG3	03FB	ORGPG4	0400	ORGPG5	0500	ORGPG6	0600	ORGPG7	0700	ORGPG8	0800
ORGPG9	0900	ORGPGA	0A00	ORGPGB	0B00	ORGPGC	0C00	ORGPGD	0D00	ORGPGE	0E00
ORGPGF	0F00	OUT500	038B	P1D	0029	P2D	002A	P55A	0001	P55B	0002
P55C	0003	P55CE0	0004	P55CE1	0002	P55IOM	0008	P55R	0000	POS1	0360
POSUN	035D	POUT	00E6	POUT1	00F0	POUT2	00F7	POUT8	00F9	PRIK1	011A
PRIKAZ	0115	RBIT	03A0	RBIT0	039E	RBIT1	03A6	RBYT1	03B1	RBYTE	03AF
REORG	0014	RES1	00B8	RES3	00B4	RESET	0085	RSYN0	0367	RSYN1	0369
RSYN2	036F	RSYN3	037B	S0	0004	S1	0010	S2	0080	S3	0020
S4	0040	S5	0008	S6	0002	S7	0001	SIZE	000B	SIZECH	0015
SP	0026	STKMAX	003A	STKMIN	0031	STOP?	001B	S1ORA	02C3	SYNCHR	0365
T01	00DE	T01RET	00DD	TAB1	0125	TAB2	0200	TABCIS	0300	TABMUX	007A
TABTL	0063	TIM128	0028	TIM2	001A	TIM3	002A	TIM4	0031	TIM5	003C
TIM6	0040	TIM7	004D	TIM72	005A	TIM8	005A	TIMER	000B	TL0	0000
TL1	0001	TL2	0002	TL3	0003	TL4	0004	TL5	0005	TL6	0006
TL7	0007	TL8	0008	TL9	0009	TLACC	000E	TLEND	000B	TLLOAD	0013
TLMEM	000D	TLNEXT	000C	TLPC	000F	TLPREV	000A	TLRUN	0010	TLSAVE	0012
TLSTEP	0011	TLSTOP	000B	TSADR	0100	TSERR	0105	TSOK	0104	TSTEND	01FC
VID0	0020	VID1	0021	VID2	0022	VID3	0023	VID4	0024	VID5	0025
WBIT	0383	WBIT0	0382	WBIT2	0387	WBYT2	0396	WBYTE	0392	WBYTE0	0391
WR55IO	03F5	XACC	0160	XACC1	016B	XF	0177	XF1	017C	XF2	0184
XF3	0184	XMEM	013F	XMEM3	013D	XMEM4	015A	XPC	016F	XR2	012F
XRUN	012C	XSP	0188	XSP1	0194	XSTEP	0137	ZC1	0311	ZC3	030A
ZC3A	01F5	ZHAS0	03F1	ZHASNI	03F0	ZOBR1	03D8	ZOBR11	03DA	ZOBR2	03D6
ZOBR3	03D2	ZOBRPC	03CD	ZPC1	03D1	ZRUN	00E2	ZTECKA	0001	ZZ0	00FC

ZZ1	0090	ZZ2	0076	ZZ3	00B6	ZZ4	009A	ZZ5	00AE	ZZ6	00EE
ZZ7	0094	ZZ8	00FE	ZZ9	00BE	ZZA	00DE	ZZC	006C	ZZE	006E
ZZF	004E	ZZH	00DA	ZZM	00C2	ZZP	005E	ZZS	00AE		

ASSEMBLY COMPLETE, NO ERRORS

```
??MIND    7#
?LENGT  417#   479    491#   656    668#   750    762#  1053   1065#  1358   1377#  1418
       1430#  1502   1514#  1556   1568#  1637   1649#  1784   1796#  1864   1895#  1976
       1988#  2057   2069#  2136   2148#  2222   2234#  2305   2325#  2348   2360#  2422
       2434#  2503   2522#  2611   2623#  2693   2705#  2774
?SIZE   189#   473    650    744   1047   1352   1412   1496   1550   1631   1778   1858
       1970   2051   2130   2216   2299   2342   2416   2497   2605   2687   2768
?START  463#   473    473    482    498#   650    650    659    675#   744    744    753
        774#  1047   1047   1056   1082#  1352   1352   1361   1384#  1412   1412   1421
       1437#  1496   1496   1505   1521#  1550   1550   1559   1614#  1631   1631   1640
       1695#  1778   1778   1787   1842#  1858   1858   1867   1941#  1970   1970   1979
       2034#  2051   2051   2060   2115#  2130   2130   2139   2194#  2216   2216   2225
       2280#  2299   2299   2308   2332#  2342   2342   2351   2406#  2416   2416   2425
       2480#  2497   2497   2506   2568#  2605   2605   2614   2669#  2687   2687   2696
       2751#  2768   2768   2777
ACC     318#   889   1146   1339
AERROR 1494#
BUFF    322#  1723   1744   1749
CBLK48   60#   489    666    760   1063   1375   1428   1512   1566   1647   1794   1893
       1986   2067   2146   2232   2323   2358   2432   2520   2621   2703
CEKEJ  2048#
CEKEND  797#   798   1194
CHR2   2121  2294#
CHVENI  335#   584
CISL2  1738#  1747
CISL4  1726   1771#
CISL5  1758#  1770
CISL6  1751   1757#
CISL7  1758   1761   1763   1766   1769   1774#
CISLO   865   1629   1723#
CLDISP 1331#
CLEAR   699  2681#
CLR0   2682#  2684
CLR5   1724  2680#
CLRD5   792   807    849    887    907    921   1190   1332   1728   2579  2678#
CMP0   1096  1243#
CMP4   1245  1247#
CMP5   1251  1254#
CMP6   1246#  1256
CODEBL   56#
CODEMB   48#    52#
CTICIS  797   1725   1746   2485#  2487
CTIKL  1338  2337#  2412
CTIKL1  809  2412#  2413   2485
CTIKL3  833  2338#
CVAD0  1405  1407#
CVADR   856   1138   1403#
DATABL   44#   415
DF        9#
ERR1    345#  1774
ERR2    346#   781
ERR3    347#  1143
ERR4    348#  1318   1493
ERR5    349#  1264
ERR6    350#  1273
ERR7    351#  1033
ERROR   791#  1034   1156   1494   1775
```

```
ERRORO  790#
EXTINT  256#   512
EXTRES  255#   718
EXTTIM  257#   515
IADD    1093   1212#
IADD2   1214   1221#
IADD3   1222   1256#
IAND    1101   1231#
ICALL   1109   1261#
ICMP1   1098   1253#
ICMP2   1099   1249#
IDISP   1088   1312#
IERROR  1144   1156#  1265   1274   1319
IHLT    1087   1190#
IINT    1113   1345#
IINTC   1348   1350#
IJF     1097   1199#
IJMP    1095   1200   1202   1206#  1269
IJMPI   1107   1204#
IJT     1111   1202#
IKEY    1112   1338#
ILDA    1091   1161#
ILDAI   1105   1160#
ILDC    1090   1175#
INOP    1089   1179#
INOT    1100   1226#
INST0   1147   1154#
INST1   1133   1137#
INST2   1136   1138#
INSTR    831    837   1130#
INTINT   511    645#
INTRES   717    722#
IOR     1108   1236#
IP1B2   1291#  1292
IP1IN   1102   1288#
IP1OUT  1103   1298#
IP2OUT  1104   1304#
IRET    1110   1271#
ISTA    1092   1166#
ISTAI   1106   1165#
ISTK2   1268   1277#
ISUB    1094   1216#
IZOBR0  1315   1332#
IZOBR1  1313   1327#
K500    1889#  2047
K750    1890#  2201
KEY      306#  2337
KP1     1891#  2040
KTIM     333#   525
LDADR   1160   1165   1166   1204   1402#  1453
LDOPF0  1212   1216   1243   1249   1253   1449#
LDOPM   1161   1231   1236   1453#
LERROR  1472   1493#
LOCBLK    64#   416    490    667    761   1064   1376   1429   1513   1567   1648   179#
        1894   1987   2068   2147   2233   2324   2359   2433   2521   2622   2704
LTOB    1288   1470#  1534
MAXINS  1115#  1141   1142
```

```
MGBITR  285#  1949
MGBITW  284#  2044
MGEND   1005  1030#
MGLD1   1021#  1027
MGLD2   1022#  1024
MGLOAD   821   1009#
MGSAVE   821    960#
MGSV1    963#   964
MGSV2    978    980#
MGSV3    991#   997
MGSV4    992#   994
MUX7     271#   536
NEWPC   1148#
NOKEY    413#   571    626    627    628   2338
NOP0    1180   1188#
NOP1    1185#  1187
ORGPG0    27#   419    493    494    664#   670    671    758#   764   1067   1379   1380
         1426#  1432   1433   1510#  1516   1517   1564#  1570   1651   1798   1897   1990
         2071   2150   2236   2327   2328   2356#  2362   2436   2524   2625   2707
ORGPG1    28#   769    770   1061#  1072   1575   1656   1803   1902   1995   2076   2155
         2241   2367   2441   2529   2630   2712
ORGPG2    29#  1077   1078   1366#  1580   1661   1808   1907   2000   2081   2160   2246
         2372   2446   2534   2635   2717
ORGPG3    30#   458    459    487#  1609   1610   1645#  1690   1691   1792#  1837   1838
         1872#  1936   1937   1984#  2029   2030   2065#  2110   2111   2144#  2189   2190
         2230#  2275   2276   2313#  2401   2402   2430#  2475   2476   2511#  2563   2564
         2619#  2664   2665   2701#  2746   2747   2782#
ORGPG4    31#   436   1587   1668   1815   1914   2007   2088   2167   2253   2379   2453
         2541   2642   2724
ORGPG5    32#
ORGPG6    33#
ORGPG7    34#
ORGPG8    35#
ORGPG9    36#
ORGPGA    37#
ORGPGB    38#
ORGPGC    39#
ORGPGD    40#
ORGPGE    41#
ORGPGF    42#
OUT500  2043   2045#
P1D      320#  1298
P2D      321#  1304
P55A     339#  1307
P55B     340#   552   2761
P55C     341#   542
P55CE0   269#   734    990   1019   1406   2764
P55CE1   270#   725    734    996   1026   1404   2763
P55ION   268#   725    734    990    996   1019   1026   1404   1406   2763
P55R     338#   685
POS1    1853#  1855
POSUN   1742   1745   1850#
POUT    1299   1305   1534#
POUT1   1540#  1541
POUT2   1543   1546#
POUT8   1537   1548#
PRIK1    809#   811    839
```

```
TIM2     534    538#
TIM3     557#
TIM4     562#   564
TIM5     571#   574
TIM6     563    574#
TIM7     583    585#
TIM72    597#
TIM8     569    572    586    587    601#
TIMER    514    523#
TL0      389#   390    391    392    393    394    395    396    397    398    626    1738
TL1      390#   628
TL2      391#   628
TL3      392#   628
TL4      393#   627
TL5      394#   627
TL6      395#   627
TL7      396#   626
TL8      397#   626
TL9      398#   626
TLACC    405#   627
TLEND    401#   412    628    1738
TLLOAD   410#   626
TLMEM    404#   628    810    2486
TLNEXT   402#   628
TLPC     406#   627
TLPREV   400#   628
TLRUN    407#   627
TLSAVE   409#   626
TLSTEP   408#   627
TLSTOP   412#   595
TSADR    778#   1041   1149   1403
TSERR    779    781#
TSOK     778    780#
TSTEND   872    894    916    928    1040   1043#
VID0     311#   312    313    314    315    316    548    697    1320   1705   1730   1741
         2587   2678
VID1     312#
VID2     313#
VID3     314#
VID4     315#
VID5     316#   813
WBIT     2040#  2125
WBIT0    963    2039#
WBIT2    2043#
WBYT2    2123#  2127
WBYTE    967    974    982    987    993    1002   2121#
WBYTE0   1004   2120#
WR55I0   687    1308   2763#
XACC     819    887#   903    930
XACC1    892    894#
XF       907#   929
XF1      910#
XF2      915#
XF3      911    916#
XMEM     819    850#
XMEM3    849#   877
XMEM4    866    872#
```

ISIS-II ASSEMBLER SYMBOL CROSS REFERENCE, V2.1

XPC	819	899#	917		
XR2	831#	834			
XRUN	820	829#			
XSP	895	921#			
XSP1	926	928#			
XSTEP	820	837#			
ZC1	910	925	1705#		
ZC3	891	1039	1626#		
ZC3A	852	901	1039#		
ZHASO	532	2761#			
ZHASNI	960	1009	2760#		
ZOBR1	1706	2588	2589	2593#	
ZOBR11	2594#	2596			
ZOBR2	862	2589#			
ZOBR3	795	859	1328	1627	2587#
ZOBRPC	838	899	2579#		
ZPC1	1193	2581#			
ZRUN	385#	829			
ZTECKA	365#	1734			
ZZ0	367#	469			
ZZ1	368#	469			
ZZ2	369#	469			
ZZ3	370#	469			
ZZ4	371#	469			
ZZ5	372#	381	470		
ZZ6	373#	470			
ZZ7	374#	470			
ZZ8	375#	470			
ZZ9	376#	470			
ZZA	377#	888			
ZZC	384#	808			
ZZE	379#	793			
ZZF	380#	908			
ZZH	383#	1191			
ZZM	382#	850			
ZZP	378#	2580			
ZZS	381#	922			

CROSS REFERENCE COMPLETE

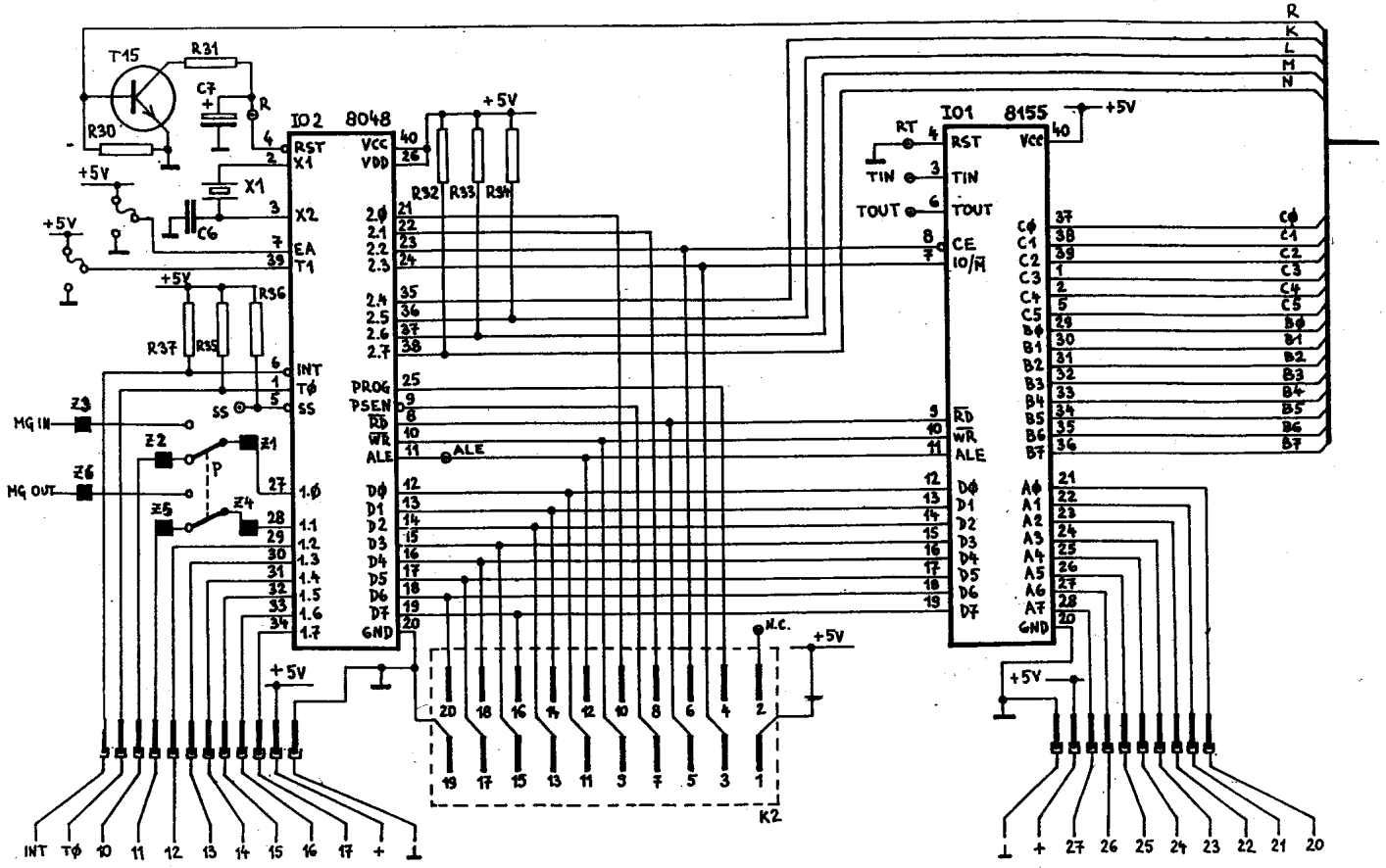
PŘÍLOHA 2:

Schema:

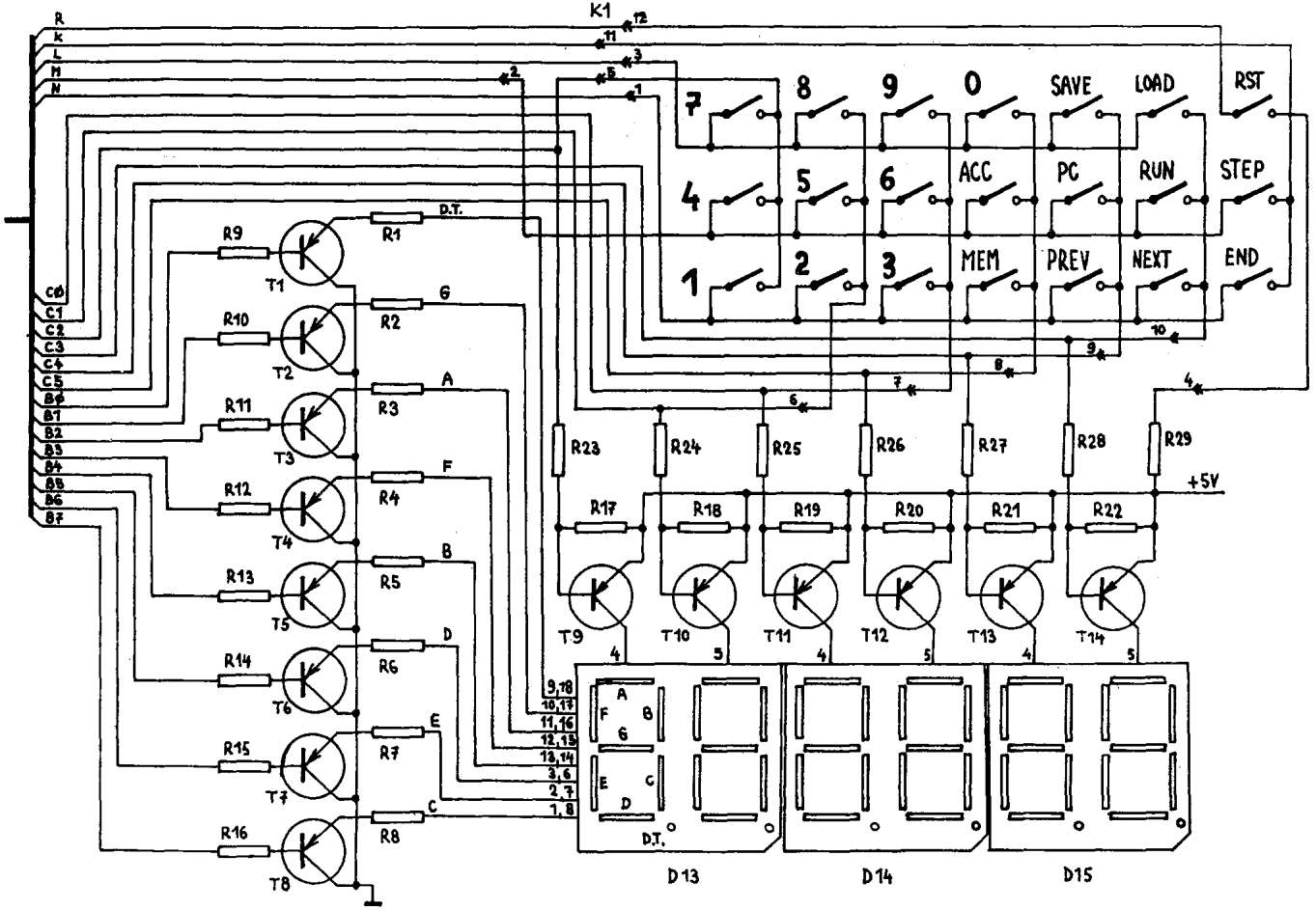
- zapojení mikropočítače PETR
- obvod klávesnice a displeje
- stabilizátoru napětí
- obvod připojení magnetofonu
- zapojení logické sondy
- zapojení vnější paměti s expanderem
- rozložení součástek na základní desce
- blokové schema mikropočítače PETR

Seznam součástek:

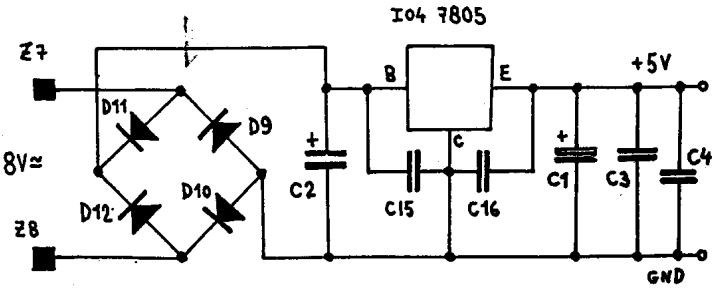
- stavebnice PETR
- logické sondy ke stavebnici PETR
- pro rozšíření stavebnice PETR



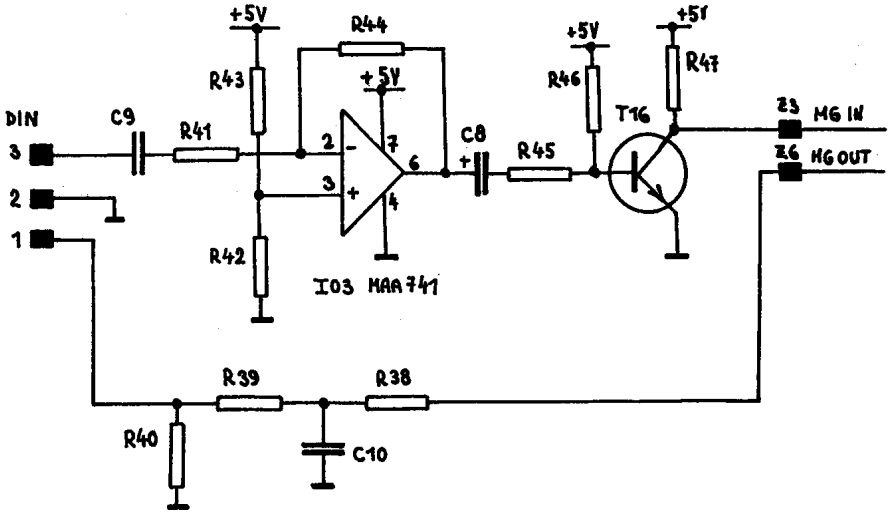
Schema zapojení mikropočítače PETR



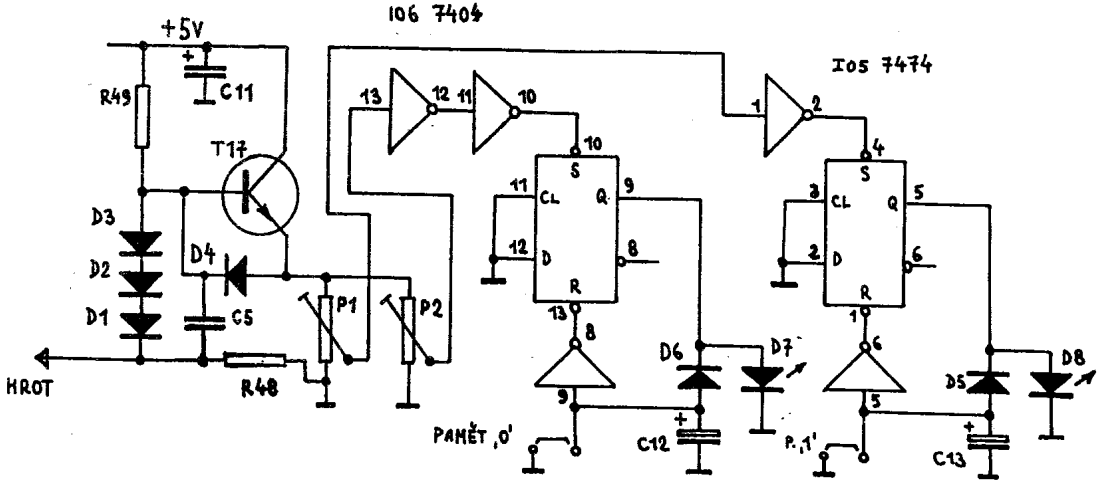
Schoma obvodu klávesnice a displeje



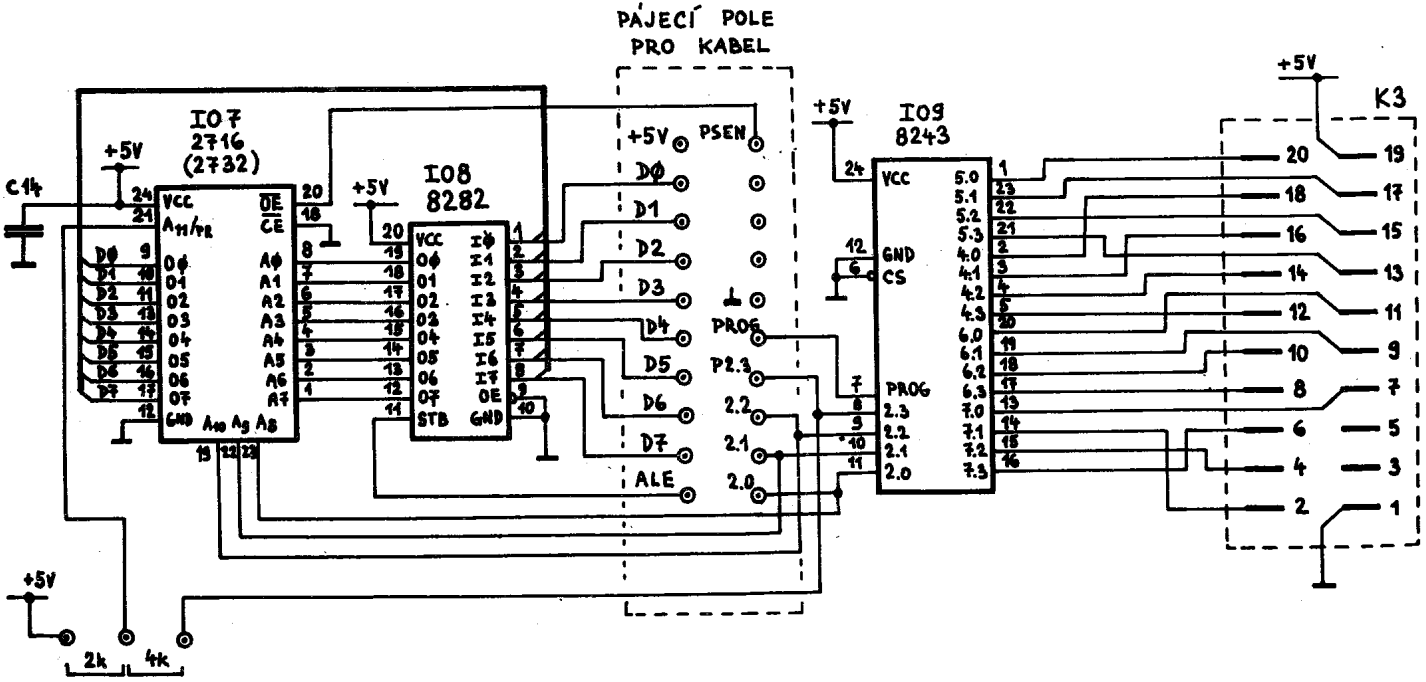
Schema stabilizátoru napětí



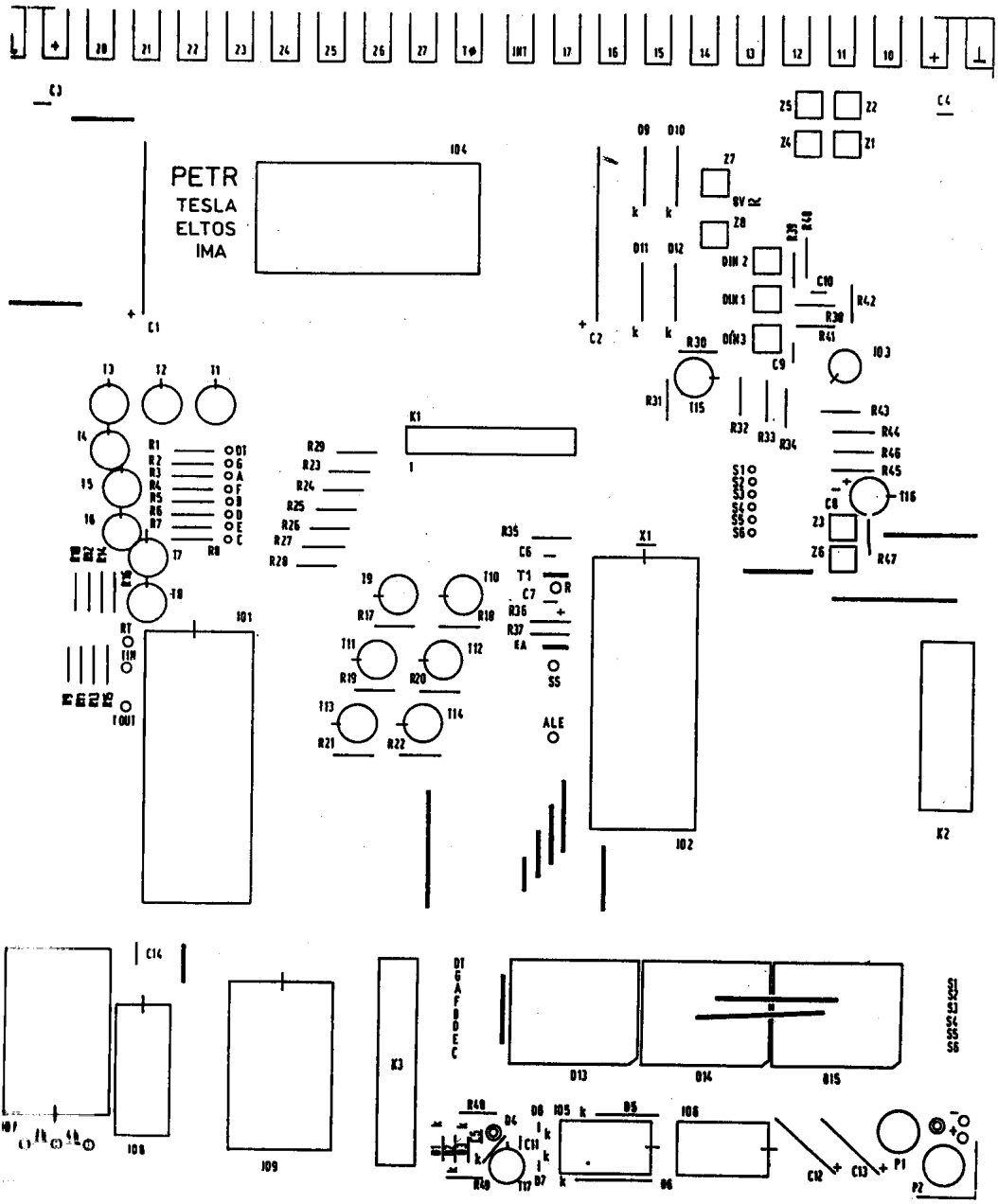
Schema obvodu připojení magnetofonu



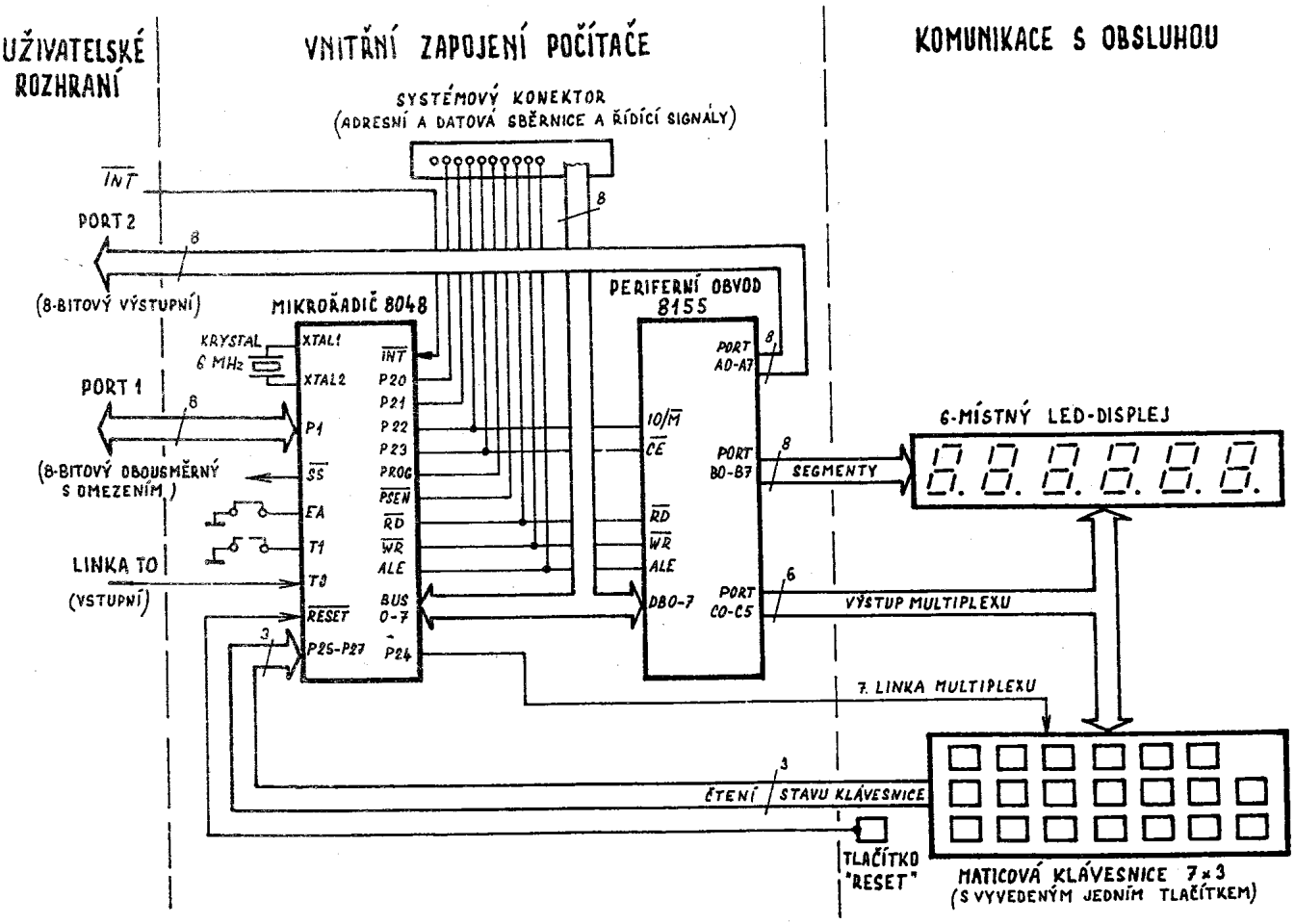
Schema logické sondy



Schema zapojení vnější paměti s expanderem



Rozložení součástek na základní desce



Blokové schema mikropočítače PETR

Seznam součástek stavebnice PETR

Integrované obvody:

1 ks	IO1	MHB 8155
1 ks	IO2	MHB 8048 (paměť programu maskou)
1 ks	IO3	MAA 741
1 ks	IO4	MA 7805

Ostatní polovodiče:

8 ks	T1-T8	tranzistor BC 178
2 ks	T15,16	tranzistor KC 508
6 ks	T9-T14	tranzistor KF 517
4 ks	D9-D12	dioda KY 132/80
3 ks	D13-D15	dvoumístná jednotka VQE 24E

Odpory TR 191:

9 ks	R1-R8,R31	...	56R
20ks	R9-R28	...	1k
1 ks	R29	...	330R
3 ks	R30,R39,R45	...	4k7
3 ks	R32-R34	...	3k3
4 ks	R35-R37,R47	...	2k2
1 ks	R38	...	47k
1 ks	R40	...	100
2 ks	R41,R42	...	10k
1 ks	R43	...	6k8
1 ks	R44	...	1M
1 ks	R46	...	22k

Kondenzátory:

2 ks	C1,C2	TF008	1000M/16V
5 ks	C3,C4,C9,C15,C16	TK 783	100n
1 ks	C6	TK 754	22p
2 ks	C7,C8	TE 135	1M/40V
1 ks	C10	TK 783	4n7

Ostatní součástky:

1 ks	X1 krystal 6MHz (malé provedení)
2 ks	objímka IO 40 pin
1 ks	konektor klávesnice (1/2 objímky IO 24 pin)
1 ks	chladič pro IO4
1 ks	přepínač (označ. P) 946 221 01 1
1 ks	konektor DIN pevný (3 dutinky)
1 ks	foliová klávesnice TS 523 (modifikovaný potisk)
35ks	koncovka 4FA 898 03
40ks	kontakt 4FA 062 06
1 ks	skříňka (Tesla Bratislava)
1 ks	deska plošných spojů (označ. TRPE)
1 m	vodič plochý PNLY 30 žil (0,15 mm^2)
0,5m	vodič jednoduchý s izol. PVC (vnitřní průměr 0,7 mm)

Seznam součástek logické sondy k stavebnici PETR

Integrované obvody:

1 ks IO5 MH 7474
1 ks IO6 MH 7404

Ostatní polovodiče:

1 ks T17 tranzistor KC508
1 ks D4 dioda KA 261
5 ks D1-D3,D5,D6 dioda KA 206
1 ks D7 dioda LED zelená LQ 1702
1 ks D8 dioda LED červená LQ 1102

Ostatní součástky:

2 ks R48,R49 odpor TR 191 ... 15k
2 ks P1,P2 odpor. trimr TP 095 680R
1 ks C5 kondenzátor TK 754 22p
1 ks C11 elektrolyt TE 131 6M8/6,3V
2 ks C12,C13 elektrolyt TE 981 10M/6V
1 ks deska plošných spojů
(oddělený díl desky stavebnice PETR)

Seznam součástek pro rozšíření stavebnice PETR

Integrované obvody:

1 ks IO7 paměť EPROM K573RF5(2716) popř.
K573RF3(2732)
1 ks IO8 MH 8282
1 ks IO9 MHB 8243

Ostatní součástky:

1 ks C14 TK 783 100n
1 ks objímka IO 24 pin
1 ks K2 konektor FRB TX 511 20 11
(na desce mikropočítače PETR)
1 ks K3 konektor FRB TY 511 20 11
1 ks konektor FRB TY 511 20 13 (pro kabel)
1 ks deska plošných spojů
(oddělený díl desky stavebnice PETR)

PŘÍLOHA 3.

Odpovědi na otázky v textu

?1 Bude čítač adres PC po provedení jedné instrukce ukazovat vždy na v pořadí další následující instrukci zapsanou v programu ?

Nebude. Například po provedení instrukce skoku (JMP) může být další instrukce ke zpracování umístěna v programu kdekoliv. Obdobných instrukcí jako JMP je více a označují se jako řídicí instrukce (viz odst. 2.2).

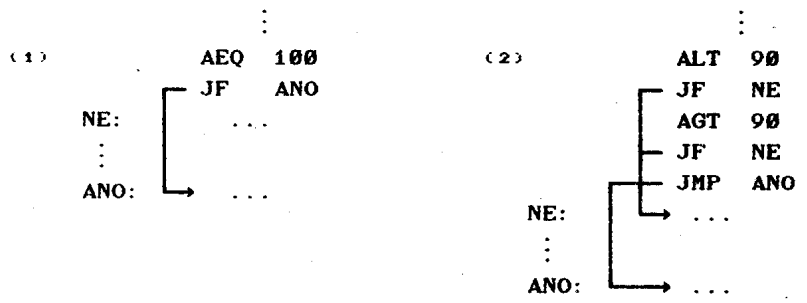
?2 Kolik podprogramů lze do sebe nejvýše vnořit tak, aby nedošlo k přeplnění zásobníku ?

Deset, neboť zásobník mikropočítače PETR má deset položek.

?3 Kolik kroků může nejvýše obsahovat program pro mikropočítač PETR ?

Počet kroků programu není omezen. Pokud jste odpověděli, že 128 resp. 256 uvědomte si, že počet kroků programu nesouvisí ani tak s velikostí paměti jako spíše s typem instrukcí. Např. jednoinstrukční program CYKL: JMP CYKL je nekonečný cyklus, při kterém procesor neustále dokola provádí tuto jedinou instrukci. Tento případ se někdy označuje jako dynamický stop procesoru. Dynamický proto, že procesor nepřetržitě pracuje, stop proto, že se přitom z adresy CYKL nikam jinam nedostane.

?4 Jsou tyto dva úseky programů funkčně stejné ?



Nejsou. Byly by ekvivalentní v případě, že by porovnávaly obsah střadače s obsahem téže buňky paměti! Navíc řešení (2) je nejen delší, ale i časově náročnější.

25 Kdy se v tomto programu nastavuje příznak F, který testuje instrukce JF ?

Při provádění instrukce AEQ, která testuje rovnost mezi obsahem střadače a srovnávací konstantou 001.

26 Tabulku pro zobrazení písmen a symbolů si laskavě, doplňte sami podle výše uvedeného návodu. Přitom tvary písmen lze volit různě, např. c lze zobrazit jako  nebo  apod. Některá písmena se zobrazují těžko, např. k jako  , některá zobrazit nejde – např. x .

písmeno	a	b	c	d	e	f	g	h	i	j	k	l	m
tvar													
hodnota	222	234	98	242	110	78	236	218	144	240	74	104	220

písmeno	n	o	p	r	s	t	u	v	x	y	z		
tvar													
hodnota	194	226	94	66	174	106	224			154			

symbol	+	-	?	!	°								
tvar													
hodnota	146	2	87	145	30								

číslice	0	1	2	3	4	5	6	7	8	9			
tvar													
hodnota	252	144	118	182	154	174	238	148	254	190			

?7 Jaký bude obsah střadače v okamžiku, kdy se program zastaví na instrukci HALT, a proč ?

Obsah střadače bude ACC=0, neboť k přetečení dojde poprvé při ACC=256. Do střadače se tedy dosadí hodnota ACC:=256-256=0 a nastaví se F:=1, viz popis instrukce ADD.

?8 Popište funkci takto zapsané instrukce:

adr	kód	instrukce	komentář
003	21.003	JMPI 003	; ?

Takto zapsaná instrukce má význam skoku sama na sebe. Program se na ní zacyklí.

?9 Upravte program tak, aby se po stisknutí tlačítek (0), (1), (2), nebo (3) zastavil !

Pro řešení lze využít posloupnosti kódů, které uvedená tlačítka generují - viz obr. 5.:

```
adr   kód          instrukce     komentář
000   26.000       KEY           ;přečíst stav klávesnice do ACC
001   10.010       AEQ  010      ;je stisknuté tlačítko ?
002   11.000       JF   000      ;ne, číst znovu
003   13.011       ALT  011      ;ukončit program ?
004   11.007       JF   007      ;ano, ale jen pro daná tlačítka
005   02.000       DISP 000      ;jinak zobrazit kód na displej
006   09.000       JMP  000      ;a číst znovu z klávesnice
007   01.000       HALT          ;stop

011   00.004       ;omezovač pro ukončení programu
```

?10 Jaký výsledek bude mít instrukce LDAI ležící na adrese dané operandem, tedy například:

```
adr        kód          instrukce
001        19.001       LDAI  001     ; ???
```

Výsledek bude stejný jako u instrukce LDC 001 , po provedení bude tedy obsah střadače ACC=001 !

?11 Napište úsek programu pro kopírování obsahu portu P1 na port P2.

Program může mít např. tento tvar:

```
adr   kód       instrukce   komentář
000   04.255    LDC   255   ; ACC := 1111 1111
001   17.008    P1OUT   8   ;zajistí správné čtení portu P1
002   16.008    P1IN    8   ; ACC := obsah P1
003   18.008    P2OUT   8   ; P2 := obsah ACC
```

?12 Proč se u instrukce RET neudává adresa, ze které má program dále pokračovat ?

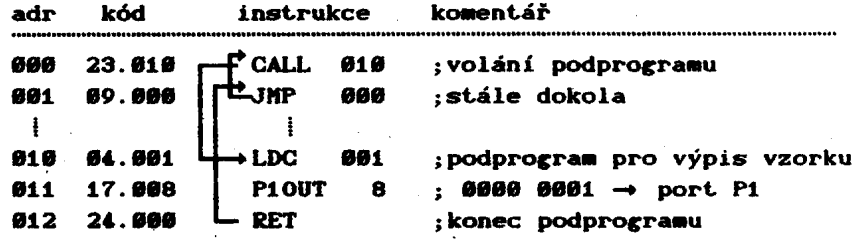
Protože návrat z podprogramu vede vždy na auresu následující za poslední instrukcí CALL. Tato návratová adresa se automaticky ukládá do zásobníku právě proto, aby mohla být použita při následující instrukci RET.
U ostatních řídících instrukcí se cílová adresa, ze které má program pokračovat, vždy uvádí.

?13 Jak lze spustit program z konkrétní adresy, řekněme 123? Lze program spustit od adresy 300 ?

Z přípustné (existující) adresy, např. 123 lze spustit program tak, že nejprve příkazem (PC) nastavíme hodnotu čítače adres na 123 a potom spustíme program příkazem (RUN) .

Z adresy 300 program spustit nelze. Adresový prostor mikropočítače PETR je v rozsahu 0 až 127 (při rozšíření až 255). Adresa 300 tedy v mikropočítači PETR neexistuje.

?.14 Může dojít k chybě E 006 (prázdný zásobník) při zpracování tohoto programu ?



Při spuštění programu od adresy 0 nebo 1 nemůže. Může k ní však dojít, spustíte-li nebo krokujete-li program od adres 010, 011, 012 (tj. instrukce RET by se měla provést dříve než instrukce CALL).

PŘÍLOHA 4:

Tabulky 8048

- seznam instrukcí mikroprocesoru 8048/8049 seřazený vzestupně podle vnitřního kódu
- abecedně seřazený seznam instrukcí mikroprocesoru 8048/8049
- funkční přehled instrukcí mikroprocesoru 8048/8049

Tabulka instrukcí PETR

Seznam instrukcí mikroprocesoru 8048/8049 seřazený vzestupně podle vnitřního kódu

kód	symb.
00	NOP
01	
02	OUTL BUS,A
03	ADD A,#d
04	JMP adr
05	EN I
06	
07	DEC A
08	INS A,BUS
09	IN A,P1
0A	IN A,P2
0B	
0C	MOVD A,P4
0D	MOVD A,P5
0E	MOVD A,P6
0F	MOVD A,P7
10	INC @R0
11	INC @R1
12	JB0 adr
13	ADDC A,#d
14	CALL adr
15	DIS I
16	JTF adr
17	INC A
18	INC R0
19	INC R1
1A	INC R2
1B	INC R3
1C	INC R4
1D	INC R5
1E	INC R6
1F	INC R7
20	XCH A,@R0
21	XCH A,@R1
22	
23	MOV A,#d
24	JMP adr
25	EN TCNTI
26	JNT0 adr
27	CLR A
28	XCH A,R0
29	XCH A,R1
2A	XCH A,R2

kód	symb.
2B	XCH A,R3
2C	XCH A,R4
2D	XCH A,R5
2E	XCH A,R6
2F	XCH A,R7
30	XCHD A,@R0
31	XCHD A,@R1
32	JB1 adr
33	
34	CALL adr
35	DIS TCNTI
36	JT0 adr
37	CPL A
38	
39	OUTL P1,A
3A	OUTL P2,A
3B	
3C	MOVD P4,A
3D	MOVD P5,A
3E	MOVD P6,A
3F	MOVD P7,A
40	ORL A,@R0
41	ORL A,@R1
42	MOV A,T
43	ORL A,#d
44	JMP adr
45	STRT CNT
46	JNT1 adr
47	SWAP A
48	ORL A,R0
49	ORL A,R1
4A	ORL A,R2
4B	ORL A,R3
4C	ORL A,R4
4D	ORL A,R5
4E	ORL A,R6
4F	ORL A,R7
50	ANL A,@R0
51	ANL A,@R1
52	JB2 adr
53	ANL A,#d
54	CALL adr
55	STRT T

kód	symb.
56	JT1 adr
57	DA A
58	ANL A,R0
59	ANL A,R1
5A	ANL A,R2
5B	ANL A,R3
5C	ANL A,R4
5D	ANL A,R5
5E	ANL A,R6
5F	ANL A,R7
60	ADD A,@R0
61	ADD A,@R1
62	MOV T,A
63	
64	JMP adr
65	STOP TCNT
66	
67	RRC A
68	ADD A,R0
69	ADD A,R1
6A	ADD A,R2
6B	ADD A,R3
6C	ADD A,R4
6D	ADD A,R5
6E	ADD A,R6
6F	ADD A,R7
70	ADDC A,@R0
71	ADDC A,@R1
72	JB3 adr
73	
74	CALL adr
75	ENT0 CLK
76	JF1 adr
77	RR A
78	ADDC A,R0
79	ADDC A,R1
7A	ADDC A,R2
7B	ADDC A,R3
7C	ADDC A,R4
7D	ADDC A,R5
7E	ADDC A,R6
7F	ADDC A,R7
80	MOVX A,@R0

kód	symb.
81	MOVX A,@R1
82	
83	RET
84	JMP adr
85	CLR F0
86	JNI adr
87	
88	ORL BUS,#d
89	ORL P1,#d
8A	ORL P2,#d
8B	
8C	ORLD P4,A
8D	ORLD P5,A
8E	ORLD P6,A
8F	ORLD P7,A
90	MOVX @R0,A
91	MOVX @R1,A
92	JB4 adr
93	RETR
94	CALL adr
95	CPL F0
96	JNZ adr
97	CLR C
98	ANL BUS,#d
99	ANL P1,#d
9A	ANL P2,#d
9B	
9C	ANLD P4,A
9D	ANLD P5,A
9E	ANLD P6,A
9F	ANLD P7,A
A0	MOV @R0,A
A1	MOV @R1,A
A2	
A3	MOVP A,@A
A4	JMP adr
A5	CLR F1
A6	
A7	CPL C
A8	MOV R0,A
A9	MOV R1,A
AA	MOV R2,A
AB	MOV R3,A

kód	symb.
AC	MOV R4,A
AD	MOV R5,A
AE	MOV R6,A
AF	MOV R7,A
B0	MOV @R0,#d
B1	MOV @R1,#d
B2	JB5 adr
B3	JMPP A
B4	CALL adr
B5	CPL F1
B6	JF0 adr
B7	
B8	MOV R0,#d
B9	MOV R1,#d
BA	MOV R2,#d
BB	MOV R3,#d
BC	MOV R4,#d
BD	MOV R5,#d
BE	MOV R6,#d
BF	MOV R7,#d
C0	
C1	
C2	
C3	
C4	JMP adr
C5	SEL RB0
C6	JZ adr
C7	MOV A,PSW
C8	DEC R0
C9	DEC R1
CA	DEC R2
CB	DEC R3
CC	DEC R4
CD	DEC R5
CE	DEC R6
CF	DEC R7
D0	XRL A,@R0
D1	XRL A,@R1
D2	JB6 adr
D3	XRL A,#d
D4	CALL adr
D5	SEL RB1
D6	

kód	symb.
D7	MOV PSW,A
D8	XRL A,R0
D9	XRL A,R1
DA	XRL A,R2
DB	XRL A,R3
DC	XRL A,R4
DD	XRL A,R5
DE	XRL A,R6
DF	XRL A,R7
E0	
E1	
E2	
E3	MOVP3 A,@A
E4	JMP adr
E5	SEL MB0
E6	JNC adr
E7	RL A
E8	DJNZ R0,adr
E9	DJNZ R1,adr
EA	DJNZ R2,adr
EB	DJNZ R3,adr
EC	DJNZ R4,adr
ED	DJNZ R5,adr
EE	DJNZ R6,adr
EF	DJNZ R7,adr
F0	MOV A,@R0
F1	MOV A,@R1
F2	JB7 adr
F3	
F4	CALL adr
F5	SEL MB1
F6	JC adr
F7	RLC A
F8	MOV A,R0
F9	MOV A,R1
FA	MOV A,R2
FB	MOV A,R3
FC	MOV A,R4
FD	MOV A,R5
FE	MOV A,R6
FF	MOV A,R7

Abecedně seřazený seznam instrukcí mikroprocesoru 8048/8049

	symb.		kód	B/C
*	ADD	A,R0	68	1/1
*		R1	69	1/1
*		R2	6A	1/1
*		R3	6B	1/1
*		R4	6C	1/1
*		R5	6D	1/1
*		R6	6E	1/1
*		R7	6F	1/1
*	ADD	A,@R0	60	1/1
*		@R1	61	1/1
*	ADD	A,#d	03	2/2
*	ADDC	A,R0	78	1/1
*		R1	79	1/1
*		R2	7A	1/1
*		R3	7B	1/1
*		R4	7C	1/1
*		R5	7D	1/1
*		R6	7E	1/1
*		R7	7F	1/1
*	ADDC	A,@R0	70	1/1
*		@R1	71	1/1
*	ADDC	A,#d	13	2/2
	ANL	A,R0	58	1/1
		R1	59	1/1
		R2	5A	1/1
		R3	5B	1/1
		R4	5C	1/1
		R5	5D	1/1
		R6	5E	1/1
		R7	5F	1/1
	ANL	A,@R0	50	1/1
	ANL	A,@R1	51	1/1
	ANL	A,#d	53	2/2
	ANL	BUS,#d	98	2/2
		P1,#d	99	2/2
		P2,#d	9A	2/2
	ANLD	P4,A	9C	1/2
		P5,A	9D	1/2
		P6,A	9E	1/2
		P7,A	9F	1/2
	CALL	0a	14	2/2
		1a	34	2/2
		2a	54	2/2
		3a	74	2/2
		4a	94	2/2
		5a	B4	2/2
		6a	D4	2/2
		7a	F4	2/2
	CLR	A	27	1/1
*		C	97	1/1
		F0	85	1/1
		F1	A5	1/1
	CPL	A	37	1/1
*		C	A7	1/1
		F0	95	1/1
		F1	B5	1/1
*	DA	A	57	1/1
	DEC	A	07	1/1

	symb.		kód	B/C
	DEC	R0	C8	1/1
		R1	C9	1/1
		R2	CA	1/1
		R3	CB	1/1
		R4	CC	1/1
		R5	CD	1/1
		R6	CE	1/1
		R7	CF	1/1
	DIS	I	15	1/1
		TCNTI	35	1/1
	DJNZ	R0,a	E8	2/2
		R1,a	E9	2/2
		R2,a	EA	2/2
		R3,a	EB	2/2
		R4,a	EC	2/2
		R5,a	ED	2/2
		R6,a	EE	2/2
		R7,a	EF	2/2
	EN	I	05	1/1
		TCNTI	25	1/1
	ENTO	CLK	75	1/1
	IN	A,P1	09	1/2
		A,P2	0A	1/2
	INC	A	17	1/1
	INC	R0	18	1/1
		R1	19	1/1
		R2	1A	1/1
		R3	1B	1/1
		R4	1C	1/1
		R5	1D	1/1
		R6	1E	1/1
		R7	1F	1/1
	INC	@R0	10	1/1
		@R1	11	1/1
	INS	A,BUS	08	1/2
	JB0	a	12	2/2
	JB1	a	32	2/2
	JB2	a	52	2/2
	JB3	a	72	2/2
	JB4	a	92	2/2
	JB5	a	B2	2/2
	JB6	a	D2	2/2
	JB7	a	F2	2/2
	JC	a	F6	2/2
	JF0	a	B6	2/2
	JF1	a	76	2/2
	JMP	0a	04	2/2
		1a	24	2/2
		2a	44	2/2
		3a	64	2/2
		4a	84	2/2
		5a	A4	2/2
		6a	C4	2/2
		7a	E4	2/2
	JMPP	@A	B3	1/2
	JNC	a	E6	2/2
	JNI	a	86	2/2

	symb.		kód	B/C
	JNT0	a	26	2/2
	JNT1	a	46	2/2
	JNZ	a	96	2/2
	JTF	a	16	2/2
	JT0	a	36	2/2
	JT1	a	56	2/2
	JZ	a	C6	2/2
	MOV	A,#d	23	2/2
	MOV	A,PSW	C7	1/1
	MOV	A,R0	F8	1/1
		R1	F9	1/1
		R2	FA	1/1
		R3	FB	1/1
		R4	FC	1/1
		R5	FD	1/1
		R6	FE	1/1
		R7	FF	1/1
	MOV	A,@R0	F0	1/1
		@R1	F1	1/1
	MOV	A,T	42	1/1
*	MOV	PSW,A	D7	1/1
	MOV	R0,A	A8	1/1
		R1,A	A9	1/1
		R2,A	AA	1/1
		R3,A	AB	1/1
		R4,A	AC	1/1
		R5,A	AD	1/1
		R6,A	AE	1/1
		R7,A	AF	1/1
	MOV	R0,#d	B8	2/2
		R1,#d	B9	2/2
		R2,#d	BA	2/2
		R3,#d	BB	2/2
		R4,#d	BC	2/2
		R5,#d	BD	2/2
		R6,#d	BE	2/2
		R7,#d	BF	2/2
	MOV	@R0,A	A0	1/1
		@R1,A	A1	1/1
	MOV	@R0,#d	B0	2/2
		@R1,#d	B1	2/2
	MOV	T,A	62	1/1
	MOVD	A,P4	0C	1/2
		A,P5	0D	1/2
		A,P6	0E	1/2
		A,P7	0F	1/2
	MOVD	P4,A	3C	1/2
		P5,A	3D	1/2
		P6,A	3E	1/2
		P7,A	3F	1/2
	MOVP	A,@A	A3	1/2
	MOVP3	A,@A	E3	1/2
	MOVX	A,@R0	80	1/2
		A,@R1	81	1/2
	MOVX	@R0,A	90	1/2
		@R1,A	91	1/2
	NOP		00	1/1

	symb.		kód	B/C
	ORL	A,R0	48	1/1
		A,R1	49	1/1
		A,R2	4A	1/1
		A,R3	4B	1/1
		A,R4	4C	1/1
		A,R5	4D	1/1
		A,R6	4E	1/1
		A,R7	4F	1/1
	ORL	A,@R0	40	1/1
		A,@R1	41	1/1
	ORL	A,#d	43	2/2
	ORL	BUS,#d	88	2/2
	ORL	P1,#d	89	2/2
		P2,#d	8A	2/2
	ORLD	P4,A	8C	1/2
		P5,A	8D	1/2
		P6,A	8E	1/2
		P7,A	8F	1/2
	OUTL	BUS,A	02	1/2
	OUTL	P1,A	39	1/2
		P2,A	3A	1/2
	RET		83	1/2
*	RETR		93	1/2
	RL	A	E7	1/1
*	RLC	A	F7	1/1
	RR	A	77	1/1
*	RRC	A	67	1/1
	SEL	MB0	E5	1/1
		MB1	F5	1/1
	SEL	RB0	C5	1/1
		RB1	D5	1/1
	STOP	TCNT	65	1/1
	STRT	CNT	45	1/1
	STRT	T	55	1/1
	SWAP	A	47	1/1
	XCH	A,R0	28	1/1
		A,R1	29	1/1
		A,R2	2A	1/1
		A,R3	2B	1/1
		A,R4	2C	1/1
		A,R5	2D	1/1
		A,R6	2E	1/1
		A,R7	2F	1/1
	XCH	A,@R0	20	1/1
		A,@R1	21	1/1
	XCHD	A,@R0	30	1/1
		A,@R1	31	1/1
	XRL	A,R0	D8	1/1
		A,R1	D9	1/1
		A,R2	DA	1/1
		A,R3	DB	1/1
		A,R4	DC	1/1
		A,R5	DD	1/1
		A,R6	DE	1/1
		A,R7	DF	1/1
	XRL	A,@R0	D0	1/1
		A,@R1	D1	1/1
	XRL	A,#d	D3	2/2

Pozn.: * - ovlivňuje CY, B/C - počet bytů/počet cyklů,
a - adresa, d - data

Funkční přehled instrukcí mikroprocesoru 8048/8049

	symb.		význam
Accumulator	ADD	A,R	Add register to A
	ADD	A,@R	Add data memory to A
	ADD	A,#d	Add immediate to A
	ADDC	A,R	Add register with CY
	ADDC	A,@R	Add data memory with CY
	ADDC	A,#d	Add immediate with CY
	ANL	A,R	And register to A
	ANL	A,@R	And data memory to A
	ANL	A,#d	And immediate to A
	ORL	A,R	Or register to A
	ORL	A,@R	Or data memory to A
	ORL	A,#d	Or immediate to A
	XRL	A,R	Excl. Or register to A
	XRL	A,@R	Excl.Or data memory to A
	XRL	A,#d	Excl. Or immediate to A
	INC	A	Increment A
	DEC	A	Decrement A
	CLR	A	Clear A
	CPL	A	Complement A
	DA	A	Decimal Adjust A
	SWAP	A	Swap nibbles of A
	RL	A	Rotate A left
	RLC	A	Rotate A left through CY
	RR	A	Rotate A right
	RRC	A	Rot. A right through CY
Input/Output	IN	A,P	Input port to A
	OUTL	P,A	Output A to port
	ANL	P,#d	And immediate to port
	ORL	P,#d	Or immediate to port
	INS	A,BUS	Input BUS to A
	OUTL	BUS,A	Output A to BUS
	ANL	BUS,#d	And immediate to BUS
	ORL	BUS,#d	Or immediate to BUS
	MOVD	A,P	Input Expander port to A
	MOVD	P,A	Out. A to Expander port
	ANLD	P,A	And A to Expander port
	ORLD	P,A	Or A to Expander port
Registers	INC	R	Increment register
	INC	@R	Increment data memory
	DEC	R	Decrement register
Branch	JMP	adr	Jump unconditional
	JMPP	@A	Jump indirect
	DJNZ	R,adr	Decrement reg. and skip
	JC	adr	Jump on CY=1
	JNC	adr	Jump on CY=0
	JZ	adr	Jump on A Zero
	JNZ	adr	Jump on A not Zero
	JT0	adr	Jump on T0=1
	JNT0	adr	Jump on T0=0
	JT1	adr	Jump on T1=1
	JNT1	adr	Jump on T1=0
	JF0	adr	Jump on F0=1
	JF1	adr	Jump on F1=1
	JTF	adr	Jump on timer flag
	JNI	adr	Jump on INT=0
	JBb	adr,	Jump on Accumulator Bit

	symb.		význam
Subroutine	CALL	adr	Jump to subroutine
	RET		Return
	RETR		Return and restore status
Flags	CLR	C	Clear CY
	CPL	C	Complement CY
	CLR	F0	Clear Flag 0
	CPL	F0	Complement Flag 0
	CLR	F1	Clear Flag 1
	CPL	F1	Complement Flag 1
Data Moves	MOV	A,R	Move register to A
	MOV	A,@R	Move data memory to A
	MOV	A,#d	Move immediate to A
	MOV	R,A	Move A to register
	MOV	@R,A	Move A to data memory
	MOV	R,#d	Move immediate to reg.
	MOV	@R,#d	Move immed. to data mem.
	MOV	A,PSW	Move PSW to A
	MOV	PSW,A	Move A to PSW
	XCH	A,R	Exchange A and register
	XCH	A,@R	Exch. A and data memory
	XCHD	A,@R	Exch.nibble of A and reg.
	MOVX	A,@R	Move ext. data mem. to A
	MOVX	@R,A	Move A to ext. data mem.
	MOVP	A,@A	Move to A from curr.page
	MOVP3	A,@A	Move to A from Page 3
Timer/Counter	STRT	T	Start Timer
	MOV	A,T	Read Timer/Counter
	MOV	T,A	Load Timer/Counter
	STRT	CNT	Start Counter
	STOP	TCNT	Stop Timer/Counter
	EN	TCNTI	Enable Timer/Counter Int.
	DIS	TCNTI	Disable Timer/Counter Int.
Control	EN	I	Enable ext. interrupt
	DIS	I	Disable ext. interrupt
	SEL	RB0	Select register bank 0
	SEL	RB1	Select register bank 1
	SEL	MB0	Select memory bank 0
	SEL	MB1	Select memory bank 1
	ENT0	CLK	Enable Clock output on T0
	NOP		No Operation

Pozn.:

R = R0-R7
@R = R0-R1
P = P1-P2 [P4-P7]
#d = data (8 bitů)
adr = adresa (8 nebo 12 bitů)
b = 0-7

SEZNAM INSTRUKCÍ STAVEBNICE "PETR"

kód	název	význam	p o p i s
01 xxx	HALT	zastavení programu	na displeji se vypíše "H" jako příznak a "*aaa*" jako adresa zastavení HALT
02 *ddd*	DISP	výpis na display	*ddd*=8 ... zobrazí ACC jako 3 cifry, *ddd*=1 až 6 ... zobrazí ACC do "*ddd*"-tého displeje *ddd*=255 ... všechny displeje se vymažou
03 *ddd*	NOP	prodleva	prodleva *ddd* * 1,28 milisekund
04 *kkk*	LDC	ACC:=*kkk*	naplnění střadače konstantou *kkk*
05 *aaa*	LDA	ACC:=MEM(*aaa*)	naplnění střadače obsahem paměti na adrese *aaa*
06 *aaa*	STA	MEM(*aaa*):=ACC	zápis obsahu střadače do paměti na adresu *aaa*
07 *aaa*	ADD	ACC:=ACC+MEM(*aaa*)	součet obsahu střadače s obsahem paměti na adr. *aaa*
08 *aaa*	SUB	ACC:=ACC-MEM(*aaa*)	rozdíl obsahu střadače a obsahu paměti na adr. *aaa*
09 *aaa*	JMP	skok v programu	skok na adresu *aaa*
10 *aaa*	AEQ	test na rovnost	F:=1 pro ACC:=MEM(*aaa*), jinak F:=0
11 *aaa*	JF	podmíněný skok	skok na adresu *aaa* pro F=1
12 *aaa*	AGT	porovnání	F:=1 pro ACC>MEM(*aaa*), jinak F:=0
13 *aaa*	ALT	porovnání	F:=1 pro ACC<MEM(*aaa*), jinak F:=0
14 xxx	CPL	ACC:=NOT ACC	inverze obsahu střadače, ACC:=255-ACC
15 *aaa*	AND	ACC:=ACC and MEM(*aaa*)	logický součin obsahu střadače a obsahu paměti na adr. *aaa*
16 *dd*	P1IN	ACC:=P1	*d*=8 ... čtení celého portu P1, *d*=0 až 7 ... čtení jen *d*-tého bitu
17 *dd*	P1OUT	P1:=ACC [2] pro *d*=0	*d*=8 ... zápis celého portu P1, *d*=0 až 7 ... zápis jen *d*-tého bitu
18 *dd*	P2OUT	P2:=ACC [2] pro *d*=0	*d*=8 ... zápis celého portu P2, *d*=0 až 7 ... zápis jen *d*-tého bitu
19 *aaa*	LDAI	ACC:=@MEM(*aaa*)	naplnění střadače podle směrníku z adresy *aaa* paměti
20 *aaa*	STAI	@MEM(*aaa*):=ACC	zápis obsahu střadače do paměti dle směrníku z adr. *aaa*
21 *aaa*	JMPI	nepřímý skok	skok dle směrníku v paměti na adrese *aaa*
22 *aaa*	OR	ACC:=ACC or MEM(*aaa*)	logický součet obsahu střadače a obsahu paměti na adr. *aaa*
23 *aaa*	CALL	volání podprogramu	skok do podprogramu na adresu *aaa*, přitom se uloží návratová adresa do zásobníku
24 xxx	RET	návrat z podprogramu	návrat z podprogramu podle adresy v zásobníku
25 *aaa*	JT	podmíněný skok	podmíněný skok na adr. *aaa* pokud je linka T0=1
26 xxx	KEY	čtení klávesnice	ACC:=KEY přečte kód tlačítka není-li stisknuta žádná klávesa, vrací ACC:=128
27 *00*	INT	ovládání přerušení	*h*=1 ... povolení, *h*=0 ... zákaz vnějšího přerušení

Seznam chyb:

1 Pokus zapsat do paměti hodnotu, kterou v ní nelze zobrazit (číslo větší než 255) nebo chyba při převodu BCD → binární číslo.

2 Adresa je mimo rozsah osazené paměti.

3 Neznámá instrukce – pokus interpretovat data jako instrukci (program nejspíše "zabloudil").

4 Operand mimo povolený rozsah (0..1, 0..8 apod.).

5 Přeplnění zásobníku návratových adres (při CALL).

6 Prázdný zásobník návratových adres (při RET).

7 Chyba čtení z magnetofonu – nesouhlasí kontrolní součet.

Vydáno pro potřebu uživatelů mikroelektronických systémů TESLA

Řada : Operační systém MS DOS
Název : Mikropočítačová stavebnice Petr
Autor : Kolektiv autorů
Vydavatel : TESLA ELTOS, státní podnik
Institut mikroelektronických aplikací
Třída Vítězného února 17, Praha 2
ve Vydavatelství a nakladatelství Novinář
ve spolupráci s čs. redakcí VN MON Praha
Vydání : první, 1988
Náklad : 1 000 výtisků

Vydala Tesla Eltos, státní podnik,
Institut mikroelektronických aplikací ve Vydavatelství a nakladatelství Novinář Praha ve spolupráci s československou redakcí VN MON v roce 1988