DŮM TECHNIKY ČSVTS PRAHA

MIKROPROCESOR Z 80 A JEHO APLIKACE

JAROSLAV TOMÁŠ HYAN

PRAHA 1984

OBSAH

ÚVOD

Ještě dnes není dořešen spor odborníků firem Intel a Texas Instruments o tom, kdo vlastně vynalezl mikropočítač. Každopádně však lze pohlížet na Dr. Teda Hoffa jako na otce mikroprocesoru; tento muž - v době kdy firma Intel měla stejné problémy jako většina jiných výrobců polovodičových obvodů, tj. vyhovět potřebě trhu po specializovaných obvodech při nízkých výrobních nákladech - přišel na myšlenku zjednodušit centrální jednotku řídicího počítače PDP 8 (Digital Equipment) v její logické struktuře a integrovat výsledek na jediný čip. Tím chtěl dosáhnout redukce hardware, neboť pak by bylo možné potřebné funkce řídit programem uloženým v pevné paměti ROM, což se mu skutečně podařilo.

Parametry tohoto prvního procesoru vyplynuly ze zakázky od firmy Busicom (na obvody nasaditelné v programovatelném kalkulátoru), a sice čtyřbitová šířka slova pro zpracování čísel v kódu BCD, binární instrukce BCD a adresování až 16 desítkových míst. Procesor v konečné formě obsahoval cca 2300 tranzistorů a nesl název 4004; nicméně nebyla jím realizována představa objednatele. Kalkulátor byl osazen 12 jinými čipy o průměrné hustotě 2000 tranzistorů na čip.

V roce 1970 přichází k firmě Intel Federico Faggin - pozdější zakladatel a prezident fy Zilog - kde společně s Masatoshi Shimou počítač dokončuje. Busicom pracuje nyní již jen se čtyřmi čipy, které jsou známy jako "set" MCS-4. Nicméně jejich výroba je stále ještě příliš drahá. Proto - se souhlasem fy Busicom - je sada nabídnuta i jiným zájemcům (inzerátem z 15.Nov.1971). Odezva je překvapující: do konce února 1972 je Intelem prodáno na 85 000 sad tohoto typu.

Avšak již v době prací na MCS-4 probíhá paralelně realizace projektu prvního osmibitového mikroprocesoru, typu 8008, a sice na objednávku fy Computer Terminals Corp. (nynější Datapoint) na LSI obvody pro inteligentní terminál 2200. Protože tento terminál je později realizován s obvody logiky TTL,

není jím procesor 8008 využit. Přesto však je v dubnu 1972 veřejnosti představen jako první osmibitový mikroprocesor na kanálu P s 45 instrukcemi a průměrnou rychlostí provádění instrukce 35 mikrosekund, a to v osmnáctivývodovém pouzdře DIL. Provozuschopný systém s 8008 vyžaduje však dalších 20 integrovaných obvodů TTL, ROM, RAM a v/v obvodů při možném adresování jen do 16 KB.

Pro mnohá nasazení jsou však procesory 4004 či 8008 příliš pomalé. Proto - zásluhou F. Faggina - přináší fa Intel na trh v dubnu 1974 jako standardní procesor typ 8080, který ještě dnes nachází své uplatnění. Je vyrobený technologií N-MOS, má prováděcí čas průměrně 2 μs, proti 8008 má o cca 30 instrukcí navíc a adresuje 64 KB. Jako nevýhodu třeba uvést potřebu tří napájecích napětí, samostatného generátoru taktu a systémového řadiče (tři čipy tedy tvoří vlastní CPJ), jen jednoduché adresovací způsoby a nepohodlné přerušování. Není proto divem, že během let vznikají a uplatňují se další typy mikroprocesorů, například v roce 1976 typ 8085. Ten svým jedním čipem představuje CPJ, má jen jedno napájecí napětí a navíc sériový vstup a výstup. Současně - výsledkem práce M. Shimy a zásluhou F. Faggina u nově založené firmy Zilog - vzniká mikroprocesor Z80, který ve své instrukční a přerušovací struktuře vykazuje podstatná zlepšení proti rodině 8080. Nelze ovšem ani opomenout typ 6502 (vylepšená verze typu 6800 Motorola p. Paddlem, s mnoha adresovacími způsoby), který v roce 1978 byl nejvíce prodávaným mikroprocesorem.

U různých výrobců polovodičů postupem času vznikají různé typy mikroprocesorů, a dokonce i jednočipové mikropočítače. V současné době lze čítat něco kolem stovky odlišných typů mikroprocesorů /26/, přičemž osmibitové typy převažují, i když dnes se do středu pozornosti dostávají šestnáctibitové mikroprocesory (Intel-8086, Zilog-Z8000, Motorola-68000, National Semiconductor-16000). Ne bez perspektivy jsou i nízkopříkonové verze osmibitových standardů v technologickém provedení C-MOS, jako jsou např. 80C85, 6301, NSC 800 atd.

Nicméně sledujeme-li produkci komerčních výrobků osobních a domácích mikropočítačů, zjistíme, že ze světově uznávaných standardů v drtivé většině jsou nasazovány mikroprocesory typu Z80 a Sy 6502. Vzhledem k tomu, že v NDR se vyrábí analogon typu Z80 (U880D), jakož i hlavní podpůrné obvody PIO, SIO a CTC (U855D, U856D a U857D), je nasnadě, že budou dosažitelné i u nás k případným aplikacím.

Následující obsah je tedy zaměřen na seznámení s vlastnostmi mikroprocesoru Z80 a jeho podpůrných obvodů, instrukčního souboru, koncepcí mikropočítačů jakož i příkladů aplikací; to vše pro potřeby potencionálního uživatele.

I. MIKROPROCESOR Z80 A JEHO ARCHITEKTURA

Blokové schéma na obr. 1 znázorňuje v hrubých rysech vnitřní architekturu mikroprocesoru Z80, tvořícího - na rozdíl od μP 8080 - v jednom čipu tzv. centrální (základní) procesorovou jednotku. (U mikroprocesoru 8080 je CPJ - centrální procesorová jednotka - totiž tvořena třemi čipy, tj. vlastním μP 8080, dále generátorem taktu 8224 a posléze systémovým kontrolorem 8228.) Avšak nejen z tohoto hlediska, ale i m.j. počtem registrů představuje mikroprocesor Z80 vyšší vývojový stupeň při zachování softwareové kompatibility směrem nahoru.

I.1. Registry

CPJ - centrální procesorová jednotka - komentovaného mikroprocesoru obsahuje 208 bitů paměti RAM, jež jsou přístupny programátorovi. Na obr. 2 je zachyceno, jak tato statická paměť je uspořádána do osmnácti osmibitových registrů a čtyři šestnáctibitové registry. Z obr. 2 je patrno, že 16 osmibitových registrů se dělí ve dvě v podstatě shodné skupiny víceúčelových registrů, které mohou být individuálně využívány buď samostatně jako osmibitové, či párovaně jako šestnáctibitové. Zbývající registry jsou určeny pro speciální používání; jsou to:

a) programový čítač PC (program counter). V tomto registru je uložena šestnáctibitová adresa běžné instrukce přečtené z operační paměti. Programový čítač je automaticky inkrementován o jedničku (tzn. v něm obsažená adresa je zvětšena o "+1") po vyslání jeho obsahu na adresovou sběrnici. Je-li v zpracovávaném programu předepsán skok, pak při jeho dekódování je adresa tohoto skoku automaticky zapsána do PC při současném přepsání inkrementované. Při iniciaci (reset) ukládá se do PC adresa 0000H jako počáteční
b) ukazatel zásobníku SP (stack pointer). Ukazatel zásobníku obsahuje šestnáctibitovou adresu vrcholu zásobníku umístěného kdekoli v operační paměti RAM. Tato zásobníková část

paměti je organizována způsobem LIFO (last in - first out). Data mohou být zapsána do zásobníkové (= zápisníkové = sklípkové) paměti z registrů CPJ či převzata ze zásobníku do registrů CPJ provedením příkazů PUSH či POP. Data vyňatá ze zásobníku jsou vždy ta poslední data, která byla předtím do zásobníku uložena. Zásobník tedy dovoluje jednoduchou implementaci mnohoúrovňového přerušení, neomezený počet vyvolaných podprogramů jakož i zjednodušení mnohých typů manipulací s daty

c) dva indexové registry (IX a IY). Dva nezávislé indexové registry obsahují šestnáctibitovou základní adresu, která je použita při tzv. indexovaném adresování. Při tomto způsobu je indexový registr použit jako základna k označení oblasti v operační paměti, do níž mají být data ukládána či z níž mají být snímána. V indexové instrukci je obsažen přídavný byte (slabika, široká 8 bitů), specifikující tzv. displacement základny. Displacement je vyjádřen jako dvojkový komplement celého čísla se znaménkem; uvedeným způsobem lze značně zjednodušit sestavu programu, zvláště obsahuje-li tabulky dat

d) registr adresy přerušení I (interrupt page address register). Mikroprocesor může pracovat způsobem, v němž nepřímé volání po kterékoliv paměťové buňce může být dosaženo jako odezva na přerušení. Registr I je použit pro tento účel pro uložení horních osmi bitů nepřímé adresy, zatímco zařízení (jež vzneslo požadavek na přerušení) poskytuje dolních osm bitů. Tak je umožněno dynamické umísťování přerušovacích rutin kdekoli v operační paměti při minimálním přístupovém čase k rutině

e) osvěžovací registr paměti R (memory refresh register). Mikroprocesor obsahuje osvěžovací čítač dovolující použít dynamické paměti stejným způsobem jako statické. Tento sedmibitový registr je automaticky inkrementován (tj. k jeho obsahu je přičtena +1) po každém zachycení instrukce (instruction fetch). Obsah osvěžovacího registru je vysílán na dolní část adresové sběrnice s řídicím signálem RFSH,

zatímco CPJ dekoduje a provádí zachycenou instrukci. Tento způsob osvěžování obsahu dynamických pamětí nezpomaluje činnost CPJ. Programátor může použít registr R k testovacím účelům; běžně se však programově nevyužívá

f) střadač A (accumulator) a příznakový registr F (flag register). CPJ obsahuje dva nezávislé osmibitové střadače (A a A´) a dva přidružené osmibitové příznakové registry (F a F´). Střadač obsahuje výsledky aritmetických či logických operací, zatímco příznakový registr F indikuje specifické podmínky osmi či šestnáctibitových operací, jako např. zda výsledek určité operace je či není roven nule. Programátor má možnost volby, s kterým párem registrů chce pracovat (A+F či A´+ F´), dále pak jejich záměnu - jednou instrukcí - takže může používat snadno kterýkoliv pár

Víceúčelové registry B, C, D, E, H a L

CPJ obsahuje dvě skupiny rovnocenných registrů; každá skupina obsahuje šest osmibitových registrů, jež mohou být individuálně použity jako osmibitové, či v párech BC, DE či HL jako šestnáctibitové. Programátor může použít kteroukoliv skupinu registrů; obvykle však používá první skupinu, zatímco druhou vyhrazuje pro nasazení v systémech, kde se požaduje rychlá odezva na přerušení.

Jediným příkazem změny lze přejít na registry druhé skupiny. Tato možnost velmi redukuje obslužný čas přerušovací rutiny vypuštěním použití zásobníku pro uchování a přečtení obsahu registrů, nutného u každého podprogramu. Rovněž lze využívat jedné skupiny registrů jako neobsáhlé paměti RAM, a to u průmyslově nasazených systémů řízených jednoúčelovým programem z pevné paměti ROM.

I.2. Instrukční registr a řízení CPJ

V blokovém schématu mikroprocesoru Z80 na obr. 1 nacházíme kromě již zmíněných speciálních a víceúčelových registrů

jako samostatnou část též tzv. instrukční registr, na nějž logicky navazuje dekodér instrukcí a řídicí jednotka CPJ. Úkolem instrukčního registru je uložení instrukce zachycené (přečtené) z operační paměti a její předání do dekodéru, kde je rozšifrována = dekódována.

Řídicí jednotka provádí uvedené funkce a generuje příslušné řídicí signály, potřebné pro přenos dat z či do registrů, z paměti či periférií do mikroprocesoru a opačně, pro obsluhu jednotlivých vstupně-výstupních linek, pro ovládání ALJ - aritmeticko-logické jednotky apod.

I.3. Aritmeticko-logická jednotka ALJ

Osmibitové aritmetické a logické instrukce jsou prováděny v aritmeticko-logické jednotce mikroprocesoru. ALJ vnitřně komunikuje se všemi registry prostřednictvím vnitřní datové sběrnice. Dále ALJ je schopna provádět následující typy funkcí: sčítání, odečítání, logický součet a součin, logické exklusivní nebo, srovnávání, levostrannou či pravostrannou rotaci či posuv (aritmetický a logický), inkrementaci, dekrementaci (snižování o -1), nahazování, mazání (=nulování) a testování bitu.

I.4. Z80 - tvar pouzdra, označení a význam jeho vývodů

Mikroprocesor Z80 je zapouzdřen do standardního čtyřicetivývodového pouzdra DIL (dual in line package), obdobně jako u nás již známý uP MHB 8080. Na obr. 3a je naznačeno systémové přiřazení funkcí jednotlivým vývodům, na obr. 3b pak označení jednotlivých vývodů vlastního pouzdra.

Dále uvádím význam jednotlivých označení:

A0 až A15 - adresová sběrnice o šířce šestnácti bitů, třístavová, aktivní v jedničce. Sběrnice adresuje paměť až do 64 Kbytů, nebo - dolními osmi bity - 256 vstupních či výstupních bran. Během osvěžovacího intervalu objevuje se na dolních sedmi bitech platná osvěžovací adresa dynamických pamětí

D0 až D7 - datová sběrnice o šířce osmi bitů, třístavová, aktivní v jedničce. Umožňuje výměnu dat mezi CPJ a pamětí či vstupně-výstupními zařízeními

$\overline{M1}$ - strojní cyklus jedna (machine cycle one) - výstup, aktivní v nule, indikující počátek právě probíhajícího instrukčního cyklu, tj. zachycení instrukce (instruction fetch). Během provádění dvouslabikového operačního kódu je $\overline{M1}$ generován při zachycení každé operační slabiky. Tyto operační dvouslabikové kódy vždy začínají s CBH, DDH, EDH nebo FDH. $\overline{M1}$ se objeví též na počátku signálu $\overline{IORQ}$, čímž indikuje cyklus potvrzení přerušení

$\overline{MREQ}$ - požadavek paměti (memory request) - třístavový výstup, aktivní v nule, indikující, že na adresové sběrnici se nachází platná adresa pro čtení či zápis do paměti

$\overline{IORQ}$ - požadavek V/V (input/output request) - třístavový výstup, aktivní v nule, indikující, že dolní polovina adresové sběrnice obsahuje platnou v/v adresu pro vstupně-výstupní operaci čtení či zápisu. Signál $\overline{IORQ}$ je taktéž generován, spolu se signálem M1, jestliže je potvrzen požadavek přerušení, a to za účelem indikace, že přerušovací vektor může být umístěn na datovou sběrnici. Operace potvrzení přerušení se objeví v intervalu $\overline{M1}$, kdy nejsou přípustné v/v operace

$\overline{RD}$ - čtení z paměti (memory read) - třístavový výstup, aktivní v nule, indikující, že CPJ chce číst data z paměti nebo vstupně-výstupních zařízení

$\overline{WR}$ - zápis do paměti (memory write) - třístavový výstup, aktivní v nule, indikující, že na datové sběrnici mikroprocesoru jsou platná data, jež mají být uložena do adresované paměti nebo vstupně-výstupního zařízení

$\overline{\text{RFSH}}$ - osvěžení (refresh) - výstup, aktivní v nule, indikující, že dolních sedm bitů adresové sběrnice obsahuje osvěžovací adresu pro dynamické paměti. Bit A7 obsahuje úroveň logické nuly, A8 až A15 je při signálu $\overline{\text{RFSH}}$ nositelem obsahu registru I

$\overline{\text{HALT}}$ - stav HALT - výstup, aktivní v nule, indikující, že CPJ provedla programovou instrukci HALT a vyčkává na nemaskovatelné či maskovatelné přerušení. Za stavu HALT provádí CPJ instrukce NOP za účelem udržení kontinuity osvěžovacích signálů $\overline{\text{RFSH}}$

$\overline{\text{WAIT}}$ - stav vyčkávání - vstup, aktivní v nule. Oznamuje procesoru, že adresovaná paměť nebo v/v zařízení není připraveno pro přenos dat. CPJ zařazuje stav vyčkávání tak dlouho, pokud na vstupu $\overline{\text{WAIT}}$ je úroveň logické nuly. Tehdy však není generován osvěžovací signál $\overline{\text{RFSH}}$. Pomocí signálu $\overline{\text{WAIT}}$ je umožněna synchronizace pomalejších pamětí či v/v zařízení; $\overline{\text{WAIT}}$ lze využít též pro krokování

$\overline{\text{INT}}$ - požadavek přerušení (interrupt request) - vstup, aktivní v nule. Signál $\overline{\text{INT}}$, s úrovní logické nuly, je generován v/v zařízením. Signál je akceptován procesorem na konci instrukčního cyklu, pokud byl vnitřní klopný IFF programově řízený, uvolněn, a pokud není aktivní signál $\overline{\text{BUSRQ}}$. Je-li požadavek přerušení akceptován, vyšle CPJ - během intervalu M1 - potvrzující signál $\overline{\text{IORQ}}$, a to na začátku příštího instrukčního cyklu. Procesor reaguje na požadavek přerušení jedním ze tří možných způsobů, které budou popsány později

$\overline{\text{NMI}}$ - nemaskovatelné přerušení (non maskable interrupt) - vstup, aktivní v nule, reagující na sestupnou hranu. Má vyšší prioritu než $\overline{\text{INT}}$ a je vždy akceptován na konci běžné instrukce, a to bez ohledu na stav vnitřního klopného obvodu IFF. $\overline{\text{NMI}}$ způsobí automatický skok CPJ na adresu 0066H, nesmí se však vyskytovat signál $\overline{\text{BUSRQ}}$

$\overline{\text{RESET}}$ - výchozí stav - vstup, aktivní v nule, iniciující CPJ. Iniciace spočívá ve vynulování programového čítače PC, rovněž tak i registru I a R, dále v uzavření vnitřního klopného obvodu IFF a volbě přerušovacího způsobu "0" (viz dále). Během iniciace přechází adresová a datová sběrnice do vysokoimpedančního stavu a rovněž ostatní řídicí signály jsou inaktivní včetně osvěžovacího (proto při použití dynamických pamětí musí mít nulovací signál RESET definované maximální trvání tak, aby nedošlo k výpadku $\overline{\text{RFSH}}$ z min. periody 2 ms)

$\overline{\text{BUSRQ}}$ - požadavek sběrnice (bus request) - vstup, aktivní v nule. Přes tento vstup je signálem s úrovní logické nuly požadován přístup na sběrnice CPJ. Pak je možné externě komunikovat s paměťmi či v/v zařízeními bez účasti procesoru; všechny třístavové výstupy CPJ pak jsou ve vysokoimpedančním stavu

$\overline{\text{BUSAK}}$ - potvrzení sběrnice (bus acknowledge) - výstup, aktivní v nule, indikující žadateli, že sběrnice jsou ve vysokoimpedančním stavu a tudíž volné k použití externě. Při BUSAK je osvěžovací signál potlačen

$\emptyset$ - jednofázový systémový takt procesoru. Pro CPJ typu Z80 činí max 2 MHz, pro Z80A max 4 MHz a pro Z80B 6 MHz. Vstup musí být opatřen rezistorem 330R připojeným na napájecí napětí +5 V; tím je zajištěno buzení vstupu z běžných hradel série TTL

I.5. Časování CPJ

Centrální procesorová jednotka provádí jednotlivé instrukce programu tak, že krokuje vždy definovanou sestavou několika základních úkonů. Tyto obsahují např. zápis do paměti či čtení z paměti, zápis do v/v zařízení či čtení z v/v zařízení a potvrzení přerušení. Všechny instrukce jsou tedy toliko různé sledy základních operací, kde každá z nich vyžaduje tří až šest taktů ke kompletaci, popřípadě více k synchronizaci CPJ s pomalejším zařízením, jako třeba paměťmi či periferiemi.

Hodinový takt mikroprocesoru je dán použitým kmitočtem generátoru taktu - T_i; základní operace je představována strojním cyklem - M_i. Závislost mezi hodinovými takty (T state) a strojními cykly (machine cycles), tvořícími dohromady jeden instrukční cyklus, přináší obr. 4. První strojní cyklus M1 každé instrukce představuje zachycení operačního kódu instrukce (jež má být provedena) - OP code fetch a sestává ze 4 až 6 taktů - pokud není prodloužen signálem WAIT. Následující strojní cykly M2, M3,... způsobí přenos dat mezi CPJ a pamětí či v/v zařízením a sestávají ze tří až pěti taktů, které taktéž mohou být prodlouženy.

Všechny časovací průběhy práce mikroprocesoru mohou být rozloženy do poměrně jednoduchých diagramů - viz dále uvedená vyobrazení, kde jsou zakresleny průběhy základních operací bez i s vyčkávacími stavy WAIT.

Na obr. 5 je zachyceno časování pro strojní cyklus M1, při němž - jako již bylo předesláno - je sejmut operační kód instrukce (tzn. vlastní instrukce určená k provedení). Přitom je obsah programového čítače umístěn na adresovou sběrnici, a to při samém začátku cyklu M1. O polovinu periody taktu později stává se aktivním signál $\overline{\text{MREQ}}$. A protože ve stejném okamžiku je již adresa na sběrnici stabilizována, lze použít sestupné hrany pulsu $\overline{\text{MREQ}}$ přímo jako uvolňovacích hodin čipů dynamických pamětí. Rovněž signál $\overline{\text{RD}}$ se stává aktivní, čímž indikuje připravenost paměťových dat k předání na datovou sběrnici. CPJ přebírá data z paměti na datovou sběrnici s nástupnou hranou hodinového (taktovacího) pulsu T3 a tatáž hrana pulsu je použita k inaktivaci signálů $\overline{\text{MREQ}}$ a $\overline{\text{RD}}$. Stav T3 a T4 cyklu M1 je použit k osvěžování dynamických pamětí. Současně v tomto čase CPJ dekóduje zachycenou instrukci; během T3 a T4 tedy A0 až A6 obsahuje osvěžovací adresu dynamických pamětí, danou stavem vnitřního čítače CPJ, při současné aktivaci signálu $\overline{\text{RFSH}}$. Tehdy je opět aktivní signál $\overline{\text{MREQ}}$ pro společné použití s $\overline{\text{RFSH}}$; samotného $\overline{\text{RFSH}}$ nelze použít, neboť stabilita osvěžovací adresy je výrobcem zaručena pouze během trvání pulsu $\overline{\text{MREQ}}$.

Na obr. 6 je znázorněno, jak se cyklus M1 prodlouží, jestliže je aktivován signál WAIT. Během taktu T2 (a každého následující Tw) prověřuje CPJ vstup $\overline{\text{WAIT}}$, a to v okamžiku sestupné hrany taktu Ø. Jestliže je tento vstup v tomto okamžiku aktivní, tj. s úrovní logické nuly, pak CPJ zavede (další) vyčkávací stav Tw pro další takt cyklu. Použití této techniky dovoluje vyrovnat se i s delšími přístupovými časy některých pamětí, a to právě zmíněným prodloužením strojního cyklu (a tím též i instrukčního cyklu).

Čtení z paměti a zápis

Obr. 7 znázorňuje časovací diagram při cyklech čtení a zápisu do paměti. Oba cykly sestávají ze tří taktů, pokud ovšem nejsou zařazeny vyčkávací stavy Tw. Signály $\overline{\text{MREQ}}$ a $\overline{\text{RD}}$ mají stejný průběh jako u strojního cyklu M1; při zápisu je signál $\overline{\text{MREQ}}$ opět aktivní po ustálení adresy, takže může být použit jako uvolňovací pro čipy dynamických pamětí. Signál $\overline{\text{WR}}$ je též aktivní po ustálení dat na sběrnici, takže může být použit přímo jako $R/\overline{W}$ puls ke skutečně každému typu polovodičových pamětí. Mimoto signál $\overline{\text{WR}}$ se stává inaktivní (neúčinný) v polovině periody taktu T3, tedy dostatečně dlouho před tím, než se může měnit adresa a data; nemůže tedy dojít k překrývání obsahů sběrnic, a tím ke konfliktům.

Na obr. 8 je zachycen časovací diagram stejných cyklů, tj. cyklů čtení a zápisu, avšak prodloužených o vyžádané vyčkávací stavy Tw. Operace prodloužení je identická s popsanou u obr. 6. K obr. 8 však třeba podotknout, že ačkoliv jak cyklus čtení, tak i cyklus zápisu jsou znázorněny samostatně, vždy se vyskytují pohromadě s jinými cykly.

Čtení z v/v zařízení a zápis

Obr. 9 přináší časovací diagram vstupně-výstupních cyklů, tj. cyklů pro čtení z v/v zařízení a pro zápis z v/v zařízení. Vyčkávací stav Tw^+, nacházející se za T2, je procesorem automaticky vkládán. To proto, že doba mezi okamžikem aktivace signálu $\overline{\text{IORQ}}$ a okamžikem, kdy procesor prověřuje úroveň vstu-

pu $\overline{WAIT}$, je velmi krátká, a tak bez tohoto vyčkávacího stavu by nebyl dostatek času pro dekódování adresy v/v zařízení, a rovněž tak i k aktivaci vstupu $\overline{WAIT}$, pokud je požadován běžný vyčkávací stav Tw. Jinými slovy, bez tohoto - vnitřně zavedeného taktu Tw^+ - by nebylo možné využít plné rychlosti mikroprocesoru. Během operace čtení z v/v zařízení je použit signál $\overline{RD}$ k uvolnění adresované brány (portu) na datovou sběrnici, stejně jako v případě čtení z paměti. Pro operaci zápisu do v/v zařízení je použit signál $\overline{WR}$ jako aktivující v/v bránu, a sice jeho nástupní hranou.

Obr. 10 zobrazuje průběh prodloužených cyklů komunikace se vstupně-výstupními zařízeními o vyčkávací stavy Tw. (Tw^+ vložen procesorem, takt Tw zařazen na základě vnějšího požadavku.)

Požadavek uvolnění sběrnic a jeho potvrzení

Následující diagram na obr. 11 zachycuje průběh časování cyklu při požadavku uvolnění sběrnic a jeho potvrzení (bus request/acknowledge cycle). Signál $\overline{BUSRQ}$ je prověřován procecorem s nástupní hranou posledního taktu každého strojního cyklu M. Je-li $\overline{BUSRQ}$ aktivní, pak procesor uvede svou datovou a adresovou sběrnici jakož i další třístavové řídicí výstupy do vysokoimpedančního stavu, a sice s nástupní hranou následujícího hodinového pulsu - taktu Tx. Od tohoto okamžiku může používat kterékoliv externí zařízení sběrnice mikroprocesoru pro přenos dat mezi v/v a pamětí. Čas odezvy na požadavek uvolnění sběrnic tedy se rovná maximálně době jednoho strojního cyklu; použití sběrnic - při defacto odpojeném procesoru - není časově omezeno. Ukončení požadavku prověřuje procesor obdobným způsobem, tzn. že zjišťuje, zda je signál $\overline{BUSRQ}$ ještě aktivní. Není-li již tomu tak, pak po uplynutí následujícího taktu jsou sběrnice jakož i řídicí linky opět připojeny k procesoru.

Při použití popsaného přímého přístupu k paměti (při odpojeném procesoru) je nutno zajistit externím řadičem osvěžo-

vání pamětí - pokud v daném systému jsou instalovány dynamické paměti. V komentovaném cyklu nemůže dojít k přerušení činnosti CPJ ani signály $\overline{NMI}$ či $\overline{INT}$.

Žádost o přerušení

Na obr. 12 je zakreslen průběh cyklu žádosti o přerušení a jeho potvrzení. Statický vstup $\overline{INT}$ je prověřován ke konci kterékoliv instrukce s nástupní hranou posledního hodinového pulsu - stavu (taktu) T. Pokud je přerušovací klopný obvod (vnitřní - IFF) ve stavu "1" a současně není-li signál $\overline{BUSRQ}$ aktivní, je generován speciální strojní cyklus $\overline{M1}$. Přitom je k signálu $\overline{M1}$ přidán signál $\overline{IORQ}$ místo obvyklého $\overline{MREQ}$, a to k indikaci perifernímu zařízení (jež požadavek na přerušení vyvolalo), že může vložit na datovou sběrnici osmibitový přerušovací vektor. Signál $\overline{IORQ}$ tedy potvrzuje přijetí žádosti o přerušení. Ve speciálním strojním cyklu $\overline{M1}$ jsou zařazeny procesorem dva vyčkávací stavy Tw^+ umožňující - v dostatečném čase - přečtení přerušovacího vektoru.

Procesor Z80 reaguje na tři druhy přerušení. Při způsobu "0" (mod "0") pracuje stejně jako mikroprocesor 8080. Při způsobu "1" provádí CPJ vždy restart na adresu 38H, což má stejný účinek jako provedení instrukce RST 7. Způsob "2" představuje nejúčinnější přerušovací prostředek: sedmibitový vektor ve spojení s obsahem registru I umožňuje provést přerušení na libovolné místo v paměti.

Nejvýznamnějších osm bitů je sejmuto z registru I, nejnižších sedm bitů z datové sběrnice, přičemž bit 0 (první) je uvažován jako nula. Tak je pevně určena adresa, která může začínat jen na sudých buňkách paměti. Tam se nalézá tabulka adres, a CPJ si tak vybírá další dva bity z paměti - tvořící novou adresu - jež je interpretována jako startovací pro přerušovací rutinu.

Obr. 13 přináší průběh cyklu žádosti o přerušení, prodloužený o další (z vnějšku vyžádaný) vyčkávací stav Tw.

Předposlední ukázka průběhu časování mikroprocesoru Z80 zachycuje cyklus nemaskovatelného přerušení (NMI - non maskable interrupt), obr. 14. Impuls na vstupu $\overline{NMI}$ nahodí vnitřní klopný obvod NMI, jehož stav je testován procesorem ke konci každé instrukce. Obvod NMI je prověřován ve stejném okamžiku jako přerušovací vstup $\overline{INT}$; proti němu má však prioritu a nemůže být vyřazen z funkce programově. Jeho obvyklou funkcí je zajistit okamžitou odezvu na důležité signály, jako např. na signál oznamující výpadek sítě. Odezva procesoru je obdobná jako při cyklu čtení, avšak s tím rozdílem, že obsah datové sběrnice je ignorován. Procesor automaticky ukládá adresu programového čítače do zásobníkové paměti a skáče na adresu 0066H; ošetřovací rutina nemaskovatelného přerušení pak musí pochopitelně začínat na této adrese.

Výstup ze stavu HALT

Kdykoliv je procesorem vykonávána programová instrukce HALT, pak CPJ provádí tzv. prázdné instrukce NOP do té doby, než je přijat požadavek přerušení. (Instrukce NOP umožňují aktivaci osvěžovacího signálu $\overline{RFSH}$, zatímco procesor je "zastaven"). Oba přerušovací vstupy $\overline{INT}$ a $\overline{NMI}$ jsou prověřovány procesorem při nástupní hraně hodinového impulsu v každém taktu T4. Je-li jeden z nich aktivní a uvolňovací klopný obvod přerušení IFF je nahozen, pak je stav HALT ukončen při příští nástupní hraně hodinového impulsu Ti. Následující cyklus je cyklem přerušení - odpovídající typu potvrzeného přerušení. (Byly-li zachyceny v současný okamžik požadavky na oba typy přerušení, pak NMI má přednost.) Každý cyklus prováděný za stavu HALT je typově shodný se strojním cyklem $\overline{M1}$, ovšem s tím rozdílem, že data přečtená z paměti jsou ignorována a procesoru je vnitřně vnucena instrukce NOP. Za programového stavu HALT je výstup $\overline{HALT}$ aktivní v nule.

I.6. Adresovací způsoby

Jednou z výhod mikroprocesoru Z80-CPU je větší počet adresovacích způsobů, který jej řadí mezi skutečně univerzální

a výkonné procesory. Adresovací způsoby jsou následující:

1. bezprostřední adresování (immediate addressing)
2. bezprostřední rozšířené adresování (imm.extended addr.)
3. adresování modifikované stránky nula (modified page-zero addressing)
4. relativní adresování
5. rozšířené adresování (extended addr.)
6. indexové adresování (indexed addr.)
7. registrové adresování (register addr.)
8. zahrnující registrové adresování (implied register addr.)
9. nepřímé registrové adresování (indirect register addr.) a
10. adresování bitů (bit addressing)

A nyní k jednotlivým způsobům:

Bezprostřední adresování

Při tomto adresovacím způsobu obsahuje slabika, následující operační kód v paměti, aktuální operand, tedy

OP kód	} jeden nebo dvě slabiky
operand	

d7 d0

Příklad: operační kód pro instrukci ADD A, n je v šestnáctkovém vyjádření C6. Uvažujme, že střadač před provedením operace obsahuje A7 a dále, že operační kód C6 se nachází v paměťovém místě 0600H, a tudíž operand n v místě 0601H s obsahem třeba 07H. Po provedení instrukce ADD A zvětší se obsah střadače A o obsah slabiky bezprostředně následující za instrukcí, tedy A = A7 + 07 = AE.

Bezprostřední rozšířené adresování

Při tomto adresovacím způsobu je operand následující za operačním kódem dvouslabičný, přičemž bezprostředně se nacházející slabika (první) za OP kódem obsahuje nižší řád operandu, druhá pak vyšší řád operandu. Jedná se tedy o rozšíření na šestnáctibitový operand.

Příklad: operační kód pro instrukci LD HL, nn je v šestnáctkovém vyjádření 21, nachází se na místě 0400H v paměti. Do registrového páru HL se má vložit šestnáctibitové slovo, třeba 3D6FH, které - ve dvou po sobě následujících slabikách - tvoří operand nn takto:

0400H	21	(OP kód)
0401H	6F	(nižší slabika)
0402H	3D	(vyšší slabika)

Adresování modifikované stránky nula

Mikroprocesor Z80 má speciální jednobytové (jednoslabikové) instrukce volání (call) do kteréhokoliv z určených osmi míst v nulté stránce paměti (tj. v bloku prvních 256 adres paměti počínaje 00H). Příkladem tohoto adresovacího způsobu je instrukce restartu RST p, která - v závislosti na operandu p, realizuje předání obsahu programového čítače do zásobníkové paměti (kvůli návratu) a jeho naplnění jednou z osmi adres nulté stránky. Adresy jsou následující: 0000H, 0008H, 0010H, 0018H, 0020H, 0028H, 0030H a 0038H.

Význam instrukce používající uvedeného adresovacího způsobu spočívá v tom, že dovoluje jedinou slabikou určit šestnáctibitovou adresu, na níž se nachází volaná obslužná rutina.

Příklad: Operační kód pro instrukci RST,08 je v šestnáctkovém vyjádření CF. Je-li třeba na adrese 1C00, pak po jejím provedení je:

1. uložena tato adresa do ukazatele zásobníku (stack pointer), a to horní byte 1C do buňky "SP-1", dolní byte 00 do "SP-2"
2. do programového čítače vložena dvoubytová adresa 0008H obslužné rutiny a pak provedena
3. po ukončení obslužné rutiny následuje návrat do hlavního programu, a to přečtením "odložené" adresy ze zásobníku (tj. 1C00), připočítání k ní jedničky (tj. 1C01) a na takto získané následné adrese pak již mikroprocesor pokračuje.

Relativní adresování

Relativní adresování používá jeden byte dat, následující za OP-kódem, k určení doplňku (displacement) D vymezujícího adresu skoku. Displacement je vždy representován jako dvojkový komplement se znaménkem, takže jeho binární hodnota se pohybuje mezi IOOO 0000 a OIII IIII (tj. 80H až 7FH, či desítkově -128_{10} až $+127_{10}$). Adresa skoku se pak rovná adrese instrukce zvětšené o upravený doplněk, tj. A-126 až A+129 - podle hodnoty doplňku, obecně tedy $A_{sk} = A_{inst} + 2 + displ.$

Příklad: podle programu

```
1C00   18   ...JR,e
1C01   06   ...e=displ.= +6
1C02   XX
  .     .
  .     .
  .     .
1C08   C6   ...ADD A,A1
1C09   A1   ...konstanta A1
1C0A   76   ...stop
```

musí μP provést nepodmíněný skok na adresu 1C00 + 02 + 06 = = 1C08; čehož důkazem po provedení je vykonání již citované instrukce C6 ukládající do vynulovaného střadače informaci A1H.

Vzhledem k tomu, že instrukce skoku je dvoubytová, musí programový čítač před jeho provedením 2x inkrementovat (jednou 1C00+1, podruhé 1C01+1); odtud je tedy odvozena výše uvedená a až do tohoto okamžiku nevysvětlená dvojka v sumě, tvořící konečnou adresu daného skoku.

Rozšířené adresování

Instrukce používající způsobu rozšířeného adresování (jedno či dvouslabičné) využívají dvou osmibitových slabik nn k vytvoření šestnáctibitové adresy, popřípadě šestnáctibitového operandu.

Typické schéma instrukce s rozšířeným adresováním má tvar:

OP kód	jedna nebo dvě slabiky
adresa nebo operand nižšího řádu	n1
adresa nebo operand vyššího řádu	n2

Příklad: instrukce uložení obsahu paměťových míst nn do střadače A (= LD A,(nn)) je typickou ukázkou rozšířeného adresování. Její zápis v programu by mohl vypadat třeba takto:

```
1C0A   3A   ...operační kód pro LD A, (nn) zapsaný
                na adrese 1C0AH
1C0B   20   ..., kde 20 představuje n1
1C0C   1E   ..., kde 1E představuje n2,
```

takže obsah paměťové buňky, který má být vložen do střadače, se nachází na adrese 1E20H; n1 a n2 jsou ukazatelem místa v paměti.

Indexové adresování

Tento způsob adresování používá dva šestnáctibitové registry IX a IY a hodnoty doplňku (displ.) - následujícího za OP kódem - k vyjádření skutečné adresy místa v paměti.

Schéma instrukce:

OP kód OP kód	vždy dvouslabičný určující jeden z obou registrů
displ.	dvojkový doplněk

Typickou ukázkou instrukce s indexovým adresováním je uložení do střadače obsahu paměťového místa, určeného obsahem indexového registru a doplňku displ.

Indexové adresování velmi zjednodušuje programování používáním tabulek dat, kdy indexový registr (IX nebo IY či oba) označuje počátek každé tabulky. Adresování je relokační - stejně jako již komentované relativní adresování.

Příklad: instrukce LD (IX+d), A znamená uložení obsahu střadače A do paměťové buňky označené obsahem IX a doplňku; v rozepsání pak:

```
1C30   DD   ...OP kód pro LD IX+d, A
1C31   77   ...OP kód pro LD IX+d, A
1C32   05   ...doplněk = 05H
```

Obsahuje-li střadač třeba 3FH a registr IX 2000H, pak po provedení daného úseku programu bude hodnota 3FH uložena do paměťové buňky 2000 + 05 = 2005H.

Registrové adresování

Mnohé operační kódy mikroprocesoru Z80 obsahují informační bity, jež určují, který z registrů CPJ má být použit pro vykonání instrukce. Typickou ukázkou registrového adresování je instrukce přenosu dat mezi různými registry procesoru - LD r, r´- kde za r či r´ mohou být dosazeny registry A, B, C, D, E, H nebo L. Jednoslabičný operační kód má obecně tvar

, kde X = 0 nebo I

Příklad: tříbitový kód registru D je 0I0, a registru E je 0II. Operační kód pro uložení obsahu registru E do registru D bude mít tedy tvar: 0 I 0 I 0 0 I I , což v šestnáctkovém vyjádření je: 53H.

Zahrnující registrové adresování

V zahrnujícím adresovém způsobu používá se speciálních instrukcí, jež automaticky zahrnují jeden nebo více registrů CPJ mezi nositele operandu. Patří sem třeba aritmetické instrukce, u nichž je vždy střadač A (registr CPJ) uchovatelem výsledku. Jiným příkladem tohoto typu speciální instrukce je třeba LD R, A; to je instrukce, která ukládá obsah střadače do osvěžovacího registru paměti.

Nepřímé registrové adresování

Tento způsob adresování specifikuje šestnáctibitový registrový pár procesoru k použití jako ukazatele kteréhokoliv místa v paměti; používá se k přenosu dat mezi procesorem a paměťovým místem, určeným obsahem zvoleného registrového páru. Typickou ukázkou je instrukce uložení obsahu střadače A do paměťové buňky, určené obsahem registrového páru DE - LD (DE), A. (Označení kteréhokoliv registru kulatými závorkami v mnemotechnickém tvaru znamená, že se vždy jedná o paměťové místo, a že obsah výrazu v kulatých závorkách je vždy jen ukazatelem!) Nepřímé registrové adresování dovoluje jednoduchý přístup k jakékoliv buňce paměti. Instrukce přesunu bloku a hledání mikroprocesoru Z80 jsou extenzí uvedeného adresovacího způsobu, kde je přidána automatická inkrementace, dekrementace či porovnávání.

Adresování bitů

Mikroprocesor Z80 má mj. větší počet instrukcí, dovolujících nahazovat, nulovat či testovat kterýkoliv bit v kterémkoliv registru CPJ či paměťovém místu.
Operační kód má obecně tvar:

X X X X X X X X	...OP kód
0 I X X X X X X	...operand

kde b = tříbitový kód bitu, tj. 000 až III a
r = tříbitový kód registru.

Pokud se manipulace s bity (s bitem) týká některé paměťové buňky, pak se používá indexového či nepřímého registrového adresování k jejímu označení.

Některé instrukce - např. aritmetické či ukládací - obsahují víc než jeden operand. V takových případech může být použito i dvou způsobů adresování - např. pro "load" bezprostředního způsobu pro určení zdroje a indexového či nepřímého registrového pro určení místa.

I.7. Operační kódy

Instrukční soubor mikroprocesoru Z80 zahrnuje 158 instrukcí, vyjádřených dále uvedenými operačními kódy, v nichž je již pojato 78 instrukcí mikroprocesoru 8080A. Instrukční soubor se dělí do skupin:

a) příkazy pro uložení (load) a výměnu. Tyto příkazy způsobují přesun dat mezi registry nebo mezi registry a pamětí. Zdroj a místo určení je specifikováno příkazem. Instrukce výměny umožňují výměnu obsahu dvou registrů.

b) příkazy pro blokový přenos a vyhledávání. Z80 používá též příkaz pro přenos bloku paměti do jiných míst operační paměti, dále pak jediný příkaz pro přezkoušení vyhledaného bloku paměti

c) aritmetické a logické příkazy. Tyto příkazy operují s daty ve střadači, registrech či v určených místech paměti. Výsledky jsou uloženy ve střadači s případným nastavením příznakových bitů. Aritmetické operace obsahují šestnáctibitové sčítání a odečítání mezi registrovými páry

d) příkazy rotace a posunu. Data mohou být posouvána či rotována ve střadači nebo v paměti

e) příkazy pro manipulaci s bity. Jednotlivé bity mohou být nahazovány (set = "1"), nulovány (reset = 0) či testovány co do logické úrovně, a to ve střadači či paměti. Výsledek testování je indikován úrovní příslušného bitu v příznakovém registru F

f) příkazy skoků (jump), volání (call) a návratu (return). Příkazem skoku je realizováno větvení programu na místo specifikované obsahem programového čítače, přičemž tento obsah je určován bezprostředním, rozšířeným či registrově nepřímým adresovacím způsobem. Volání je speciální forma skoku, u něhož adresa, následující příkaz volání, je uložena do zásobníkové paměti (stack), a to ještě před provedením příkazu. Návrat je pak reversním postupem volání. Tato kategorie obsahuje dále speciální příkazy restartu

g) příkazy vstupu a výstupu. Tyto příkazy umožňují přenos dat mezi registry a pamětí k externím zařízením v/v

h) řídicí příkazy. Tyto příkazy způsobují zastavení CPJ nebo vykonávání tzv. prázdné instrukce NOP. Schopnost uvolnit či "uzavřít" přerušovací vstupy je další možností řídicích příkazů, rovněž tak zvolení přerušovacího způsobu 0, 1 nebo 2.

Formáty dat a příkazů

Paměť mikroprocesoru Z80 je organizována po osmibitových slovech, zvaných slabiky (bytes), kde každé slabice přináleží určité paměťové místo, označené odpovídající adresou. Mikroprocesor Z80 umožňuje přímé adresování 65 536 slabik, tj. 64 Kbytů paměti (2^{16}), a to šestnáctibitovou adresovou sběrnicí; daný paměťový prostor však nemusí být vždy plně využitelný, tzn. nemusí být vždy osazen všemi paměťovými obvody RAM a ROM - viz dále.

Data jsou ukládána ve formátu:
D7 D6 D5 D4 D3 D2 D1 D0 , kde D0 je nejméně významný bit (least significant bit = LSB) a kde D7 je bitem nejvýznamnějším (most significant bit = MSB). Osmibitová slabika může mít tři významy: tvoří buď instrukci (příkaz), nebo adresu, nebo data, popř. jejich část, přičemž správný význam vyplývá z polohy (umístění) jedné každé slabiky v daném programu.

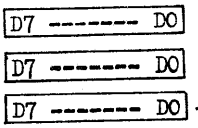
Vlastní příkazy mikroprocesoru Z80 jsou jednoslabičné - dle vzoru

D7 ------- D0	OP kód

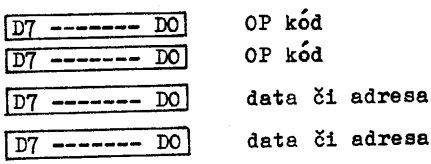
nebo dvouslabičné

D7 ------- D0	OP kód
D7 ------- D0	data či adresa

či tříslabičné



nebo dokonce čtyřslabičné



OP kódy ukládání a výměny

Tabulka č. 1 zobrazuje operační kódy všech osmibitových instrukcí ukládání mikroprocesoru Z80. Rovněž tabulka zobrazuje adresovací způsoby každého příkazu. Zdroj dat je uveden v nejvyšší vodorovné řadě, zatímco místo určení (příkazu) je specifikováno levým krajním sloupcem tabulky. Například příkaz pro uložení obsahu registru B do registru C má operační kód 50H a jedná se o již komentovaný registrový způsob adresování.

Mnemonický výraz v assembleru pro tuto úplnou skupinu příkazů je "LD" (od "load"), následovaný místem určení a zdrojem, tedy:

LD	místo určení	zdroj

Z tabulky č. 1 dále vyplývá, že jsou přípustné jisté kombinace adresovacích způsobů, např. pro zdroj registrové a pro místo určení nepřímé registrové - viz LD (HL), C s operačním kódem 71H. (Význam tohoto kódu, vyjádřeného slovně, je tento: ulož do paměťového místa, indikovaného obsahem registrového páru HL, obsah registru C.) Většina operačních kódů je jednoslabičná, nicméně nalezneme dvouslabičné příkazy ukládání pro víceúčelové registry A až L, využívající způsobu bezpro-

středního adresování, např. LD E, 31H (ulož do registru E konstantu 31H) s OP kódem 1E a operandem 31.

Všechny ukládací příkazy, používající indexové adresování pro zdroj či umístění, jsou tříslabičné, kde třetí slabiku tvoří doplněk d (displ.). Rovněž oba příkazy rozšířeného adresování (OP kódy 3Ann a 32nn) jsou též tříslabičné.

Ukládací příkazy do paměťových buněk, používajících indexového adresování pro umístění a bezprostředního adresování pro zdroj, jsou čtyřslabičné - např. LD (IX+d), n (ulož operand n do paměťového místa určeného obsahem indexového registru IX a dvojkového doplňku d - OP kód DD 36 d n).

Políčka, podbarvená tangírou v tab. č. 1, obsahují operační kódy společné pro mikroprocesor 8080A - předchůdce Z80.

Tabulka č. 2 zobrazuje operační kódy šestnáctibitových operací ukládání; je podobná předcházející tabulce, ovšem rozšířené adresování se zde týká všech registrových párů. Instrukce nepřímého registrového adresování, jež specifikují ukazatel zásobníku, jsou významné instrukce "PUSH" a "POP". Liší se od ostatních šestnáctibitových instrukcí ukládání tím, že ukazatel zásobníku je automaticky dekrementován či inkrementován, jakmile daná slabika je vložena (do) či vyňata ze zásobníku. Např. instrukce: PUSH BC s operačním kódem C5H po provedení generuje následující sekvenci příkazů:

```
dekrementuj SP   (= ukazatel zásobníku)
LD (SP), B
dekrementuj SP
LD (SP), C
```

přičemž obsah registrového páru BC je uložen v buňkách zásobníku takto:

C na buňce specifikované obsahem SP,
B na buňce specifikované obsahem SP+1.

Instrukce POP ... je přesnou reversí příkazu PUSH. Jak PUSH, tak i POP používají šestnáctibitový operand a slabika vyššího řádu je vždy uložena první a vyňata poslední.

Příkazy, používající rozšířeného bezprostředního adresování pro zdroj, vyžadují za OP kódem dvě slabiky dat, např.: LD DE, 1234H (ulož do reg. páru DE konstantu 1234H), kde slabika nižšího řádu operandu následuje za OP kódem a za ní teprve slabika vyššího řádu, tedy takto: .ll 34 12., kde ll je OP kód uvedeného mnemonického příkazu.

Tabulka č. 3 obsahuje operační kódy šestnáctibitových instrukcí výměny, implementovaných do mikroprocesoru Z80. Operační kód 08H dovoluje programátorovi přepnout z příznakového a střadačového registru AF na AF', OP kód D9H pak přejít na duplikační sadu šesti víceúčelových registrů B', C', D', E', H',a L'. Vzhledem k tomu, že tyto OP kódy jsou jednoslabičné, vyžádá si přechod z jedné sady na druhou minimum času, což lze výhodně použít pro rychlé odezvy na požadavek o přerušení.

OP kódy blokového přesunu a vyhledávání

V tabulce č. 4 jsou obsaženy operační kódy čtyř výkonných instrukcí blokového přenosu. Každá z nich pracuje se třemi registrovými páry, a sice s HL (ukazuje místo zdroje dat), s DE (ukazuje místo určení dat) a s BC (operuje jako čítač slabik).

Po iniciaci těchto registrových párů může programátor použít kteroukoliv z uvedených čtyř instrukcí, tj. "LDI, LDIR, LDD nebo LDDR", jimž odpovídají postupně operační dvouslabičné kódy: EDA0H, ED80H, EDA8H, ED88H. Instrukce "LDI" (= load and increment) přesune jednu slabiku z místa určeného obsahem registrového páru HL do místa určeného obsahem registrového páru DE, přičemž registrové páry po provedení přesunu jsou automaticky inkrementovány, takže ukazují na následující buňky paměti; současně je dekrementován čítač sla-

bik BC. Instrukce "LDIR" je rozšířením předcházející (= load, increment and repoat); při jejím provedení tedy dochází k postupnému přesunu slabik až do vynulování čítače slabik - tedy skutečně k přesunu bloku dat jedinou instrukcí.

Instrukce "LDD a LDDR" mají obdobný účinek; registrové páry po přesunu jsou však dekrementovány. HL tedy obsahuje vždy vyšší adresu než DE. V tabulce č. 5 jsou vyznačeny operační kódy čtyř instrukcí vyhledávání. První ("CPI" - compare and increment) srovnává data ve střadači s obsahem paměťového místa určeného obsahem registrového páru HL. Výsledek srovnání je uložen v jednom příznakovém bitu, registrový pár HL je inkrementován, registrový pár BC, sloužící jako čítač slabik, je dekrementován.

Instrukce "CPIR" (= compare, increment and repeat) opakuje srovnávání slabik, pokud není dosaženo shody nebo vynulování páru BC. Tak je možné touto jedinou instrukcí prohledat vytčenou paměťovou oblast pro jakýkoliv osmibitový znak.

Instrukce "CPD" a "CPDR" pracují obdobně, avšak po každém srovnání je pár HL dekrementován. Vyhledávání v daném případě startuje na nejvyšší adrese paměťového bloku.

OP kódy aritmetických a logických příkazů

Tabulka č. 6 zahrnuje operační kódy všech osmibitových aritmetických příkazů, jež mohou být provedeny střadačem; zahrnuty jsou též OP kódy přírůstku ("INC") a úbytku ("DEC").

Zvolená aritmetická operace je provedena s daty nacházejícími se ve střadači A a určenými registrem, výsledek operace je uložen ve střadači, s výhradou příkazu "CP", jenž nechává po provedení obsah střadače nedotčen. Příkaz "ADD" vykoná binární součet dat obsažených ve střadači i v určeném zdroji, příkaz "SUB" pak odečtení (rozdíl). Je-li použito příkazů "ADC" či "SBC", je pak při provedení přičítán či odečítán přenosový bit (carry flag).

Další operační kódy, uvedené v tabulce č. 6, vztahují se na tři logické příkazy, a sice na příkazy logického součinu ("AND"), logického součtu ("OR") a exklusivního součtu ("XOR").

V tabulce č. 7 je znázorněno pět operačních kódů pracujících se střadačem či přenosovým bitem. První z nich (instrukce "DAA") umožňuje podmíněné přizpůsobení střadače pro aritmetické operace s BCD (binary coded decimal) čísly. Pro tyto operace se používá příznakový bit N, který je nahozen (= 1), byla-li poslední operace odečítání. "CPL" vytváří komplement střadače, instrukce "NEG" pak dvojkový doplněk obsahu střadače. "CCF" generuje doplněk příznakového bitu přenosu a "SCF" nahazuje přenosový bit (C = 1).

Tabulka č. 8 zobrazuje operační kódy šestnáctibitových aritmetických instrukcí, prováděných mezi registrovými páry. Je zde obsaženo pět skupin příkazů pro sečítání, sečítání a odečítání s přenosem - ovlivňující všechny příznakové bity - jakož i pro inkrementování a dekrementování. Instrukce "ADC" a "SBC" značně zjednodušují šestnáctibitové aritmetické operace včetně výpočtu adresy.

OP kódy příkazů rotace a posunu

Význačnou schopností mikroprocesoru Z80 je možnost rotovat či posouvat data ve střadači, v kterémkoliv víceúčelovém registru nebo paměťové buňce. Všechny OP kódy rotace a posunu jsou zachyceny v tabulce č. 9. Směry rotace či posunu instrukcí "RLC" až "SRL" jsou vyznačeny vpravo, vedle tabulky č. 9. Instrukce "RLD" a "RRD" dovolují rotaci BCD čísla obsaženého ve střadači s dvounibbleovým číslem v paměti, jehož adresa je indikována registrovým párem HL; tyto instrukce dovolují realizovat účinnou aritmetiku s BCD čísly.

OP kódy příkazů pro manipulaci s bity

Jak již bylo řečeno, mikroprocesor Z80 má možnost nahazovat, nulovat či testovat kterýkoliv bit ve střadači, v kte-

rémkoliv víceúčelovém registru či paměťové buňce, a to jedinou instrukcí. V tabulce č. 10 je zahrnuto 240 operačních kódů dvou či čtyřslabičných instrukcí vyhrazených pro uvedený účel. Registrovým adresováním je specifikován střadač či kterýkoliv víceúčelový registr, na němž má být žádaná operace provedena. Nepřímým registrovým či indexovým adresováním lze operovat s vnějšími paměťovými místy. Operace testování bitů nahrazuje příznakový bit Z, jestliže testovaný bit je nulový, viz stať popisující příznakové bity registru F.

OP kódy příkazů pro skoky, volání a návrat

V tabulce č. 11 jsou zaznamenány všechny operační kódy příkazů skoků, volání a návratů, jež jsou implementovány do CPJ Z80. Skokem se realizuje větvení programu, při němž je programový čítač naplněn šestnáctibitovou adresou specifikovanou jedním ze tří adresovacích způsobů (bezprostřední rozšířené, relativní a nepřímé registrové). Skupina skoků je charakterizována jistými odlišnými podmínkami, po jejichž naplnění dochází k větvení. Nejsou-li splněny, program pouze pokračuje na následující instrukci v dané sekvenci. Všechny podmínky jsou závislé na datech v příznakovém registru F - viz dále. Bezprostřední rozšířené adresování se používá ke skokům do kteréhokoliv místa paměti, přičemž operační kód je následován dvěma slabikami vyjadřujícími adresu místa větvení - instrukce je tedy tří byteová. Při relativním adresování je instrukce skoku pouze dvouslabičná, kde druhá slabika je tvořena dvojkovým komplementem (displacement) k existujícímu stavu PC, a to v rozsahu +129 až -126. (Doplněk se počítá od adresy OP kódu příkazu.)

Při nepřímém registrovém adresování nacházíme tři typy skoků, jež se vyznačují - pro určení místa větvení - vložením obsahu registrového páru HL či obsahu jednoho z indexových registrů IX nebo IY přímo do programového čítače. To umožňuje, že se skoky mohou stát funkcí předcházejících kalkulací.

Volání je speciální forma skoku, kdy adresa následující po příkazu volání je uložena do zásobníkové paměti ještě před provedením skoku. Instrukce návratu je opakem volání, kdy data nacházející se na vrcholu zásobníku (stack) jsou vložena (popped) přímo do programového čítače k vytvoření adresy skoku na pokračování programové sekvence. Instrukce volání ("call") a návratu ("ret") realizují jednoduše přechod na podprogramy.

V instrukčním souboru mikroprocesoru Z80 nacházíme ještě dvě speciální instrukce, a sice 'návrat z přerušení' ("RETI") a 'návrat z nemaskovatelného přerušení' ("RETN") s operačními kódy ED4D a ED45, jež jsou nepodmíněné, obdobně jako 'návrat' reprezentovaný kódem C9. Obě instrukce se používají s výhodou v zapojeních s periferními obvody či zařízeními s danou prioritou přerušení, kdy umožňují rychlý návrat (v minimálním čase) z přerušovací rutiny.

K usnadnění řízení programových smyček lze použít instrukce "DJNZ e". Tato dvouslabičná instrukce používající relativního adresování dekrementuje registr B a skok se uskuteční, jestliže obsah registru B je různý od nuly. Displacement e je opět vyjádřen jako znaménkový dvojkový doplněk v rozsahu +129 až -126.

V tabulce č. 12 jsou zaznamenány operační kódy příkazů restartu; instrukce jsou jednoslabičné a je jich osm. Jejich význam spočívá v tom, že jimi mohou být jednoduše volány často používané rutiny (začínající pochopitelně na té či oné adrese) při minimalizaci použití paměti.

OP kódy příkazů vstupu a výstupu

Mikroprocesor Z80 - proti typu 8080 - vyznačuje se větším počtem příkazů vstupu a výstupu. Adresování vstupních či výstupních zařízení může být buď bezprostřední, nebo nepřímé registrové, s využitím registru C. Operační kódy vstupních a výstupních příkazů jsou zachyceny v tabulkách č. 13 a č. 14,

z nichž je patrno, že při nepřímém registrovém adresování se pohyb dat může dít mezi v/v zařízením a kterýmkoliv víceúčelovým registrem. Dále se setkáváme s 2x čtyřmi příkazy pro blokový přesun dat. Tyto příkazy jsou podobné příkazům blokového přenosu v paměti (viz tab. č. 4), ovšem s tím, že používají registrový pár HL jako ukazatel zdroje (v paměti - pro výstup) nebo umístění (pro vstup), zatímco registr B je použit jako čítač slabik. V registru C je uložena adresa portu, pro nějž je vstupní či výstupní příkaz určen. A protože registr B má šíři osm bitů, může se příkaz blokového přenosu v/v vztahovat na max. délku bloku 256 bytů.

Při příkazech "IN A, n" a "OUT n, A" objeví se adresa v/v zařízení v dolní polovině adresové sběrnice (A0 až A7), zatímco v horní je indikován obsah střadače A. Při všech v/v příkazech nepřímého registrového způsobu, včetně blokových přenosů, obsah registru C je přenesen do dolní poloviny adresové sběrnice (tzn. opět adresu v/v zařízení), zatímco obsah registru B je přenesen do horní poloviny adresové sběrnice.

OP kódy řídicích příkazů

Poslední tabulka č. 15 zachycuje sedm operačních kódů řídicích příkazů. První čtyři ("NOP", "HLT", "DI" a "EI") jsou stejné jako u mikroprocesoru MHB 8080, zbývající tři se týkají již jen µP Z80 a určují jeden ze tří přerušovacích způsobů ("IM0", "IM1", "IM2") - viz stať "Žádost o přerušení" v kapitole I.5, dále pak I.9.

I.8. Příznakové bity (flags)

Každý z dvou příznakových registrů (F, F´) obsahuje šest bitů (S, Z, H, P/V, N a C), jejichž hodnota je jedničková nebo nulová při různých operacích CPJ. Čtveřice z těchto bitů je testovatelná; jinými slovy řečeno, tyto bity jsou používány jako podmínky pro skokové, volací a návratové příkazy - např. příkaz skoku může být realizován, jestliže určitý bit v příznakovém registru je jedničkový.

Tvar příznakového registru, jenž je dostupný programátorovi, je následující:

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	H	X	P/V	N	C

kde X vyjadřuje blíže neurčené bity, jimž není třeba věnovat pozornost.

Testovatelné bity jsou tyto:

1. bit přenosu C (carry) - Úroveň tohoto bitu je odvozena od bitu nejvyššího řádu střadače. Přenosový bit C je např. nahozen (tj. nabyde hodnoty logické jedničky) během vykonání příkazu součtu, při němž je generován přenos nejvyšším bitem střadače. Tento bit je taktéž nahozen, jestliže je generován požadavek výpůjčky (borrow) při odčítání. Rovněž tak příkazy rotace a posunu ovlivňují hodnotu tohoto bitu

2. bit nuly Z (zero) - Tento bit je nahozen (= 1), jestliže výsledkem operace je nula ve střadači - tzn. nulový obsah střadače. V opačném případě je Z nulový

3. bit znaménka S (sign) - Tento bit je určen pro použití se znaménkovými čísly. Je nahozen (= 1), jestliže výsledek operace je záporné číslo. A protože nejvýznačnější sedmý bit (D7) reprezentuje znaménko čísla (záporné číslo má vždy D7 = 1), indikuje znaménkový bit S stav sedmého bitu střadače

4. bit parity/přeplnění P/V (parity/overflow) - Tento dvouúčelový bit indikuje paritu výsledku ve střadači při provádění logických příkazů (např. "AND A,B") a reprezentuje přeplnění při provádění aritmetických operací s dvojkovými znaménkovými doplňky. Bit přeplnění indikuje, že dvojkové doplňkové číslo ve střadači je chybné, protože buď převyšuje možné maximum (+127), nebo je menší než možné minimum (-128) zobrazitelné v dané doplňkové notaci.

Při logických příkazech ("AND", "OR", "XOR") je paritní bit nahozen (= 1), jestliže výsledek má sudou paritu, a je nulován, jestliže výsledek má paritu lichou.

Pozn.: funkce bitu P/V je mírně odchylná u mikroprocesoru 8080, viz literární pramen /70/.

Zbývající dva bity příznakového registru jsou netestovatelné, jsou určeny pro aritmetiku s BCD čísly. Jsou to:

5. bit polovičního přenosu H (half carry) - Tento příznakový bit má hodnotu 1, jestliže operace odčítání či přičítání (s BCD čísly) generuje přenos do či výpůjčku z D4 střadače. (Jak již sám název vypovídá, je hodnota bitu H odvozena z výsledku operace vykonané na dolních čtyřech bitech - tedy polovině slabiky.) Při použití příkazu "DAA" používá se tento příznakový bit ke korekci výsledku předcházejícího desítkového sčítání či odčítání
6. bit příznaku sčítání/odčítání N (add-subtract flag) - Protože algoritmus korekce s BCD čísly je odchylný pro sčítání či odečítání, specifikuje bit N posledně provedený typ příkazu, čímž je zajištěna správnost operace "DAA". N = 1, jestliže posledně provedeným příkazem bylo odčítání.

V tabulce č. 16 je zachyceno, jak zmíněných šest bitů je ovlivňováno různými příkazy, prováděnými CPJ. Příkazy, které nejsou uvedeny v tabulce, neovlivňují příznakové bity.

Pro ujasnění je třeba se zmínit o některých speciálních případech: tak instrukce blokového vyhledání nahazuje příznak Z, jestliže poslední operace srovnávání indikuje shodu mezi zdrojovými a střadačovými daty. Rovněž paritní bit je nahozen, jestliže obsah čítače slabik (registr BC) není roven nule. Stejný průběh změn úrovně paritního bitu je u příkazů blokového přenosu.

Speciální případ představují též blokové vstupní a výstupní příkazy. Zde je bit Z používán k indikaci stavu registru B, jenž je použit jako čítač slabik. Je-li blokový v/v přenos

kompletní, je příznakový bit Z vynulován (tj. B = 0), zatímco v případě příkazu blokového přenosu je paritní bit vynulován při kompletní (ukončené) operaci. Konečně je-li obsah osvěžovacího nebo I registru uložen do střadače, klopný obvod uvolňující přerušení vloží svůj stav do paritního příznaku, takže konečný stav CPJ může být v kterémkoliv čase zabezpečen.

I.9. Přerušení (interrupt)

Přerušení je jistý druh procesu, v němž mikropočítač přeruší provádění hlavního programu a započne vykonávat jiný program (podprogram, subrutinu), umístěný někde jinde v paměti. Ačkoliv je zde jistá podobnost se skokem či voláním, přerušení je odezvou na vnější stimul.

Existuje celá řada důvodů, pro něž je požadována schopnost přerušení. Jedním z nich je třeba možnost bezprostředně reagovat na vnější podmínku alarmu, např. signál z detektoru kouře, překročení max přípustné teploty apod. Jiným důvodem je, že vstupní data se objevují v relativně dlouhých intervalech, kdy by bylo neekonomické, aby počítač na ně čekal a v mezičase nepracoval - např. při snímání výsledku z analogově/číslicového převodníku, jehož konverzní čas obnáší 20 ms. (V mezičase, před ukončením konverze, může mikropočítač provést stovky až tisíce operací.) Jiným takovým příkladem je snímání a vydávání dat z periferních zařízení, jako je třeba tiskárna, dálnopis, terminál atd. Elektromechanická zařízení, jako je třeba tiskárna, jsou notoricky "pomalá", i když vytisknou 500 znaků/min, kdy tedy vytisknutí jednoho znaku si vyžádá doby 125 ms. Naprosti tomu Z80-CPU je schopna vydat do vstupního vyrovnávacího registru tiskárny jeden znak ve třech mikrosekundách - z toho vyplývá, že Z80 může vyslat 42 000 znaků v čase, v němž však tiskárna vytiskne pouze jediný! Cesty k zvládnutí této situace spočívají vesměs v tom, že periferní zařízení vysílá signál do CPJ indikující, že je připraven převzít další znak. Tento signál současně oznamuje požadavek na přerušení.

Jakmile je přijat požadavek na přerušení (ať již z jakéhokoliv důvodu - alarmu, připravenosti atd.), mikropočítač přeruší hlavní program (po dokončení právě probíhajícího příkazu) a skočí na příslušnou o b s l u ž n o u rutinu; po jejím vykonání vrací se zpět na přerušené místo v hlavním programu. Tak je - pomocí přerušení - efektně hospodařeno s provozním časem mikroprocesoru.

Z80-CPU má dva přerušovací vstupy, nemaskovatelný NMI, který nemůže být programátorem vyřazen z funkce, a maskovatelný INT, jehož funkci lze programově potlačit. Vstup INT se tedy používá k příjmu požadavku na přerušení; lze jej však během programových period - v nichž jsou zařazeny časovací smyčky, jež se nesmějí přerušit - programově vyřadit, popřípadě opět uvolnit. Děje se tak příkazy "EI" (enable interrupt) a "DI" (disable interrupt), jež nahazují nebo nulují uvolňovací klopný obvod přerušení IFF. Je-li IFF vynulován, nemůže být mikroprocesorem přijat požadavek na přerušení (přes vstup INT). Ve skutečnosti obsahuje Z80-CPU klopné obvody dva, a sice IFF_1 a IFF_2. Stavem IFF_1 je uvolněno či vyřazeno přerušení, zatímco IFF_2 dočasně paměťově uchovává výstupní stav IFF_1.

Při iniciaci Z80-CPU jsou oba klopné obvody vynulovány, takže přerušení je vyřazeno. Mohou být uvolněny instrukcí "EI" programátorem v kterémkoliv čase. Je-li prováděna instrukce EI, nemůže být přijat požadavek na přerušení, to lze teprve tehdy, až je provedena i instrukce následující po "EI". Toto "pozdržení" je nutné pro případy, kdy by následující instrukce mohla být "RET" (return - návrat), u níž je přerušení přípustno až po jejím úplném ukončení.

Je-li přijat požadavek na přerušení, jsou oba klopné obvody automaticky vynulovány, čímž zamezují dalšímu příjmu požadavku přerušení. Pokud si programátor přeje uvolnění dalšího přerušení, musí zařadit znovu instrukci "EI" (řeší se obvykle smyčkou).

Účelem obvodu IFF_2 je uchování stavu IFF_1, když se objeví nemaskovatelné přerušení. Je-li přijato nemaskovatelné přerušení, je IFF_1 vynulován, čímž je zamezeno další přerušení, pokud ovšem není znovu programátorem uvolněn (= nahozen). Při instrukcích "LD A,I" (load register A with register I) či "LD A,R" je kopírován stav IFF_2 do příznakového bitu parity, kde může být uchován či testován.

Druhý způsob uchování stavu IFF_1 poskytuje instrukce "RETN" (return from non-maskable interrupt). Protože tato instrukce indikuje, že návrat z nemaskovatelného přerušení byl ukončen, obsah IFF_2 je nyní kopírován do IFF_1, čímž je jeho obsah automaticky obnoven.

Stavy obou klopných obvodů přerušení jsou v tabulce č. 17, a to pro různé, výše citované instrukce.

Příjem požadavku na přerušení

Nemaskovatelné přerušení je akceptováno mikroprocesorem v kterýkoliv okamžik. Objeví-li se tedy požadavek na přerušení, CPJ ignoruje následnou instrukci, kterou vyzvedla, a místo ní provede restart na adrese 0066H. Chová se jako při restartu, avšak skáče - jak uvedeno - na odchylnou adresu, než jsou adresy restartu, viz tab. č. 12.

Při maskovatelném přerušení může CPJ reagovat jedním ze tří volitelných způsobů:

- způsob 0 (mode 0), jenž je identický se způsobem odezvy na požadavek přerušení u mikroprocesoru 8080A, a jenž je automaticky zaveden vždy po iniciaci (reset). Lze ovšem tento či jiný způsob zavést i programově, v daném případě tedy příkazem "IM 0". Při tomto způsobu zařízení vyvolávající přerušení umístí na datovou sběrnici instrukci, kterou CPJ v zápětí provede. Obvykle se jedná o jednobyteovou instrukci restartu. Ovšem může být též použito tříslabičné instrukce skoku na kterékoliv místo v pracovní paměti. Počet hodinových taktů, potřebných pro provedení dané instruk-

ce je automaticky zvětšen o dva, které přidává CPJ za účelem časového vyrovnání pro externí řízení priority v přerušovacím řetězci

- způsob 1 (mode 1). Je-li tento způsob zvolen programátorem instrukcí "IM1", je odezvou mikroprocesoru skok na adresu 0038H. (Odezva je identická s průběhem nemaskovatelného přerušení, ovšem s tím, že místo adresy 0066H se skok děje na zmíněnou adresu 0038H. Dalším rozdílem je opětné přidání dvou taktů k dokompletování exekutivy instrukce.)

- způsob 2 (mode 2). Je nejvýkonnějším způsobem odezvy na požadavek přerušení. Dovoluje nepřímé volání po jakémkoliv místě v paměti, na němž se nachází počátek obslužné rutiny. Zatímco v modu 0 je možných jen osm míst přerušení (= úrovní), v modu 2 jich je dokonce 128. Při tomto způsobu programátor určí tabulku šestnáctibitových adres startu obslužných rutin přerušení. Tato tabulka může být umístěna kdekoliv v paměti. Pro příjem požadavku přerušení musí být vytvořen šestnáctibitový ukazatel k dosažení potřebné obslužné rutiny, počínající na té či oné adrese v tabulce. Horních osm bitů ukazatele je vytvořeno obsahem registru I. Pro ten účel musí být registr I předem naplněn požadovanou hodnotou programátorem, a to příkazem "LDI, A" (po resetu je registr I vynulován). Dolních osm bitů ukazatele je naplněno vektorem vnějšího zařízení. Ve skutečnosti je však z vnějšku požadováno pouze sedm bitů, protože první bit musí být vždy nula. Adresy ukazatele tak začínají vždy na sudých místech v tabulce, což je nutné proto, že jimi označené paměťové buňky obsahují pouze dolní byte startovacích adres, přičemž horní byte sousedí. V tabulce je tak max. 128 startovacích adres rutin

Na obr. 16 je znázorněna typická sekvence přerušení v modu 2. Hlavní program je umístěn na stránce 64 (viz příloha) - tzn. s počátkem na adrese 4000H, vektorová tabulka je umístěna od adresy 8000H a obslužná rutina přerušení má počátek na adrese 6050H. Registr I obsahuje tedy 80H a za-

řízení vyvolávající přerušení je naprogramováno na vydání 04H na datovou sběrnici při přerušení. Klopný obvod IFF_1 musí být nahozen a úroveň na $\overline{BUSRQ}$ musí být jedničková. Pak je tato sekvence:

1. vnější zařízení vyšle signál $\overline{INT}$ do CPJ
2. po potvrzení příjmu požadavku (interrupt acknowledge) vyšle periférie data 04H na datovou sběrnici. Ty splynou s 80H z I registru do adresy 8004H. Na této adrese v paměti se nachází dolní byte šestnáctibitové adresy určující start obslužné rutiny, horní byte pak na adrese 8005H
3. programový čítač je inkrementován a jeho obsah je uložen do paměťového zásobníku (stack)
4. do programového čítače je uložena adresa, nalezená na místě 8004H; je to adresa obslužné rutiny 6050H. Řízení programu tedy skáče na tuto adresu
5. po provedení posledního příkazu "RETI" obslužné rutiny jsou data programového čítače PC, uchráněná v zásobníku, vložena zpětně do PC (4004H)
6. hlavní program pokračuje nyní od instrukce na adrese 4004H

II. Podpůrné obvody série Z80

K mikroprocesoru Z80 existují - obdobně jako i u série Intel 80 a u jiných mikroprocesorů /26/, /18/ - tzv. podpůrné obvody, které pomáhají spolu s CPJ vytvářet vlastní mikropočítačový systém. Jsou to obvody: Z80-PIO (parallel input/output port controller)
Z80-SIO (serial input/output controller)
Z80-CTC (counter timer circuit)
Z80-DMA (direct memory access circuit)
Z80-COMBO

II.1. Z80-PIO

Je paralelní vstupně-výstupní řadič obsahující dvě osmibitové brány, jejichž způsob práce je uživatelem programovatelný. Obvod připouští kompletní "handshaking", takže může být použit i pro synchronní přenos.

Z80-PIO může být naprogramován k provozu ve čtyřech odchylných způsobech: vstupním, výstupním obousměrným (jen pro port A) a bitovým:

a) výstupní způsob (byte output mode) - též zvaný způsob - Ø, je používán k zajištění zápisu dat z CPJ po datové sběrnici do periférie. Je-li zvolen tento způsob, pak operace zápisu dat((data write) vyžaduje generování handshakingového signálu "ready", jenž oznamuje periférii, že data jsou připravena a platná. Signál "ready" s aktivní úrovní "1" a data zůstávají k dispozici, pokud není zpět přijmut strobovací signál z periférie;

b) vstupní způsob (byte input mode) - též zvaný způsob - 1, dovoluje vybranné bráně - portu - chovat se pouze jako vstupní. Když je mikroprocesorem prováděna operace čtení, vyšle PIO signál "ready" do periférie. Ten sdělí periférii, že Z80-CPJ je ve stavu schopném přijmout vstupní data. Nato periférie odpoví vysláním strobovacího signálu, který způsobí přenos dat do vstupního registru PIO;

c) obousměrný způsob (byte bidirectional mode) - též zvaný způsob - 2, využívá osmibitovou bránu A pro obousměrný přenos dat, čtyř bitových vedení pak pro handshaking;

d) bitový způsob (bit control mode) - též zvaný způsob - 3, používá se pro stavové a řídicí aplikace. Při něm se nevyužívají signály pro hanshaking. Způsob definuje, které linky datové sběrnice budou vstupní a které výstupní. Jakmile je zvolen způsob -3, pak následující slovo přivedené do PIO definuje zmíněné podmínky; bližší v /37/, /38/.

Z80-PIO se nachází ve čtyřicetivývodovém pouzdře DIL s roztečí mezi oběma řadami 15 mm; na obr. 17 je zakreslena vnitřní struktura tohoto podpůrného obvodu, jakož i označení jeho vývodů. Struktura sestává ze sběrnicového rozhraní, vnitřní logiky řízení (ICL), v/v logiky portu A a portu B a řídicí logiky přerušení.

Rozhraní sběrnice dovoluje přímé připojení PIO k CPJ bez oddělovacích zesilovačů sběrnic (ty se používají u rozsáhlej-

ších systémů). Oba vstupně-výstupní porty dovolují přímé připojení periferií. Vstupně-výstupní logika portů se skládá z šesti registrů pro styk s potvrzením ("handshaking"), jež jsou rozkresleny na dalším obr. Nacházíme zde 8 bitový datový vstupní registr, 8 bitový datový výstupní registr, 2 bitový řídicí registr zvoleného způsobu, 8 bitový maskovací registr, 8 bitový v/v výběrový registr a 2 bitový maskovací řídicí registr.

Význam jednotlivých vývodů pouzdra je tento:

D_0 až D_7 - vývody pro připojední k datově sběrnici mikropočítače. Jsou obousměrné a třístavové; přes ně se děje transfer dat jakož i přenos příkazů z CPJ do PIO

B/A SEL - vstup, aktivní v jedničce určuje pro přenos port B, při Ø pak port A. (Často se pro výběr používá adresová linka A_0.)

C/D SEL - vstup, aktivní v jedničce, určuje typ dat přenášených mezi CPJ a PIO. Při úrovni log "1" je osmibitová informace na datové sběrnici interpretována jako příkaz pro vybraný port; úroveň log "0" na tomto vstupu sděluje obvodu, že informace na sběrnici jsou data určená k přenosu. (Pro výběr se často používá adresové linky A_1.)

$\overline{CE}$ - vstup, aktivní v nule, provádí (při úrovni log "0") uvolnění čipu k činnosti

Ø - systémový takt (vstup) pro vnitřní synchronizaci jistých signálů; takt je jednofázový

$\overline{M1}$ - vstup, aktivní v nule, pro příjem signálu strojního cyklu jedna z mikroprocesoru; synchronizuje vnitřní operace PIO, zejména přerušení

$\overline{IORQ}$ - vstup, aktivní v nule, pro příjem stejnojmenného signálu z CPJ, synchronizující vnitřní operace PIO

$\overline{RD}$ - vstup, aktivní v nule, detekující cyklus čtení CPJ. Používá se se signály B/A SEL, C/D SEL, $\overline{CE}$ a $\overline{IORQ}$ pro přenos dat z PIO do CPJ

IEI - přerušovací vstup, aktivní v jedničce, indikuje, že žádné další zařízení vyšší priority není obsluhováno přerušovací rutinou

IEO - přerušovací výstup, aktivní v jedničce (interrupt enable out). Je to další signál požadovaný k formování schéma priorit přerušení. Má úroveň "1" pouze při IEI = = "1" a neobsluhuje-li CPJ požadavek přerušení z tohoto PIO. Signál IEO tak blokuje požadavek přerušení od zařízení s nižší prioritou, pokud zařízení s vyšší prioritou je právě obsluhováno

$\overline{INT}$ - výstup, aktivní v nule, indikující, že PIO požaduje přerušení od CPJ

A_0 až A_7 - osmibitová obousměrná třístavová sběrnice portu A

$\overline{A\ STB}$ - vstup, aktivní v nule, synchronizující port A signálem z vnějšího zařízení

A RDY - výstup, aktivní v jedničce, sdělující, že registr A je připraven (ready)

B_0 až B_7 - osmibitová obousměrná třístavová sběrnice portu B. Sběrnice je používána k přenosu datové, stavové či řídicí informace mezi portem B a periferním zařízením. Jednotlivé vývody portu B jsou schopny dodávat proud 1,5 mA při 1,5 V k řízení tranzistorů typu Darlington

$\overline{B\ STB}$ - vstup, aktivní v nule, dovolující periférii synchronizovat port B, tedy jako $\overline{A\ STB}$. Je-li však port A naprogramován v modu -2 (obousměrném), pak přes tento vstup jsou strobována data z periférie do vstupního registru portu A

B RDY - výstup, aktivní v jedničce, sdělující, že registr B je připraven. Je-li však port A v modu -2, pak výstup sděluje (úrovní log. "1"), že vstupní registr portu A je prázdný a připravený přijmout data z periférie

Nastavení PIO

1. Reset

Z80-PIO provádí "reset" automaticky po připojení napájecího napětí. Tehdy jsou vynulovány oba maskovací registry portů, sběrnice portů jsou vysokoimpedanční, signály kvitování (handshake "ready") jsou neaktivní, způsob -1 je volen auto-

maticky. Dále jsou vynulovány oba výstupní registry portů, oba klopné obvody uvolňující přerušení; nenuluje se registr adresového vektoru.

PIO může být resetován signálem $\overline{M1}$ při současné nepřítomnosti signálu $\overline{RD}$ nebo $\overline{IORQ}$, a to ihned (v tom okamžiku), kdy přestane být aktivní. (PIO nemá samostatný vývod "res" v důsledku limitace počtu vývodů pouzdra DIL.) PIO setrvává v iniciačním stavu (reset), pokud neobdrží řídicí slovo z CPJ.

2. Vektor přerušení

PIO byl navržen k spolupráci s CPJ používající odezvy na přerušení dle způsobu "2" (viz v části I.5. stať Žádost o přerušení). Tento způsob vyžaduje vydání přerušovacího vektoru interupčním zařízením. Vektor je CPJ použit k formování adresy přerušovací rutiny daného portu, je umístěn na datovou sběrnici během cyklu potvrzení přerušení, a to zařízením s nejvyšší prioritou, dožadujícího se obsloužení. Žádaný vektor přerušení je do PIO nahrán zápisem ř í d i c í h o slova do daného portu v tomto formátu:

	D7	D6	D5	D4	D3	D2	D1	D0
řídicí slovo	V7	V6	V5	V4	V3	V2	V1	0

(nulový bit na vedení D0 indikuje, že řídicí slovo je vektor přerušení)
V tomto případě je D0 použit jako tzv. příznakový bit; je-li nulové hodnoty, způsobí uložení informace V1 až V7 do vektorového registru.

3. Volba operačního způsobu

Jak již bylo řečeno, port A může být provozován čtyřmi způsoby. Všimněme si, že označení způsobů bylo částečně voleno mnemonicky: způsob -0 = Out, -1 = In, -2 = dvousměrný. Port B však nemůže pracovat způsobem -2, jinak je shodný s A, proti A však má nižší prioritu. Způsob operace musí být stabilizován zapsáním ř í d i c í h o slova do PIO v následujícím formátu:

	D7	D6	D5	D4	D3	D2	D1	D0
Řídicí slovo	M1	M0	X	X	1	1	1	1
	označují způsob				označují řídicí slovo operačního způsobu			

X = nepoužitý bit

Bity D7 a D6 tedy formují dvojkový kód určující jednoznačně zvolený způsob operace, a to:

D7	D6	
0	0	-0 = výstupní
0	1	-1 = vstupní
1	0	-2 = obousměrný
1	1	-3 = bitový

Bity D5 a D4 jsou přitom ignorovány, naproti tomu D0 až D3 musí mít tvar 1111.

Je-li volen bitový způsob (-3), pak následující řídicí slovo vyslané do PIO musí definovat vstupní a výstupní linky sběrnice portu. Formát tohoto řídicího slova má tvar:

	D7	D6	D5	D4	D3	D2	D1	D0
Řídicí slovo	107	106	105	104	103	102	101	100

Má-li kterýkoliv bit úroveň log "1", pak je korespondující vedení v dalším použito jako vstupní; obdobně - při úrovni log "0" stává se dané vedení vedením výstupním.

Během způsobu -3 je strobovací signál ignorován a linka "ready" má úroveň log "0". Data mohou být zapisována do portu či čtena z portu centrální procesorovou jednotkou (CPJ) v kterýkoliv okamžik.

4. Řídicí slovo přerušení

Řídicí slovo přerušení pro každý port má tvar:

	D7	D6	D5	D4	D3	D2	D1	D0
řídicí slovo	E.I	and or	H L	masks foll.	0	1	1	1
		používá se pouze v -3			označuje řídicí slovo přerušení			

Je-li D7 - 1, pak je nastaven klopný obvod přerušení a port může generovat přerušení. Je-li D7 = 0, nelze přerušení generovat. Objeví-li se požadavek přerušení za stavu D7 = 0, je vnitřně zaznamenán v PIO a přenesen do CPJ po uvolnění (při D7 = 0). Bity D6, D5 a D4 jsou používány hlavně v způsobu -3, avšak nahození D4 (D4 = 1) v řídicím slově během kteréhokoliv způsobu operace způsobí vynulování nevyřízeného požadavku na přerušení. Bit D6 (and/or) definuje logické operace, jež mohou být provedeny při monitorování portu. Je-li D6 = 1, je specifikována funkce AND, při D6 = 0 pak funkce OR. Např. je-li specifikována funkce AND, musí všechny bity nabýt specifikovaný stav dřív, než bude generováno přerušení, zatímco funkce OR generuje přerušení, jestliže některý specifikovaný bit přechází do aktivního stavu. Bit D5 definuje aktivní polaritu linky datové sběrnice portu, jež má být monitorována. Je-li tedy D5 = 1, pak datové linky portu jsou monitorovány pro stav o úrovni log "1", při D5 = 0 pak pro stav o nízké úrovni (log "0").

Jestliže bit D4 = 1, pak následující ř í d i c í slovo, vyslané do PIO, definuje maskování:

	D7	D6	D5	D4	D3	D2	D1	D0
řídicí slovo	Mb7	Mb6	Mb5	Mb4	Mb3	Mb2	Mb1	Mb0

Pouze ty linky portu, jejichž maskovací bit je nulový, budou monitorovány pro generování přerušení.

Uvolňovací klopný obvod přerušení portu může být nulován (reset) nebo nahozen (set log "1"), aniž by bylo nutné modifikovat zbytek řídicího slova přerušení, a sice následujícím příkazem ve tvaru osmibitového slova:

	D7	D6	D5	D4	D3	D2	D1	D0
příkaz:	int. enable	X	X		0	0	1	1

kde D7 se rovná nule či jedničce podle přání programátora.

Shrneme-li výše uvedené, pak na popisovaném podpůrném obvodu - proti obdobnému typu 8255 konkurenčního výrobce - jsou zvláště význačné při práci v modu -3 schopnosti určení:
- které z osmi vývodů portu mají sloužit jako vstupní či výstupní,
- které ze vstupních vedení mají vyvolávat přerušení a které nikoliv,
- zda k vyvolání přerušení poslouží nízká (L = log "0") či vysoká (H = log "1") úroveň signálu,
- zda vyvolání přerušení naprogramovanými vstupními vedeními má být mezi sebou provázané logickým součtem (OR) nebo logickým součinem (AND).

Všechny tyto stavy jsou realizovatelné libovolně často někde v uživatelském programu vydáním jednoho či více ř í d i c í c h slov paralelnímu obvodu, tj. jeho naprogramováním.

II.2. Z80-SIO

Tento podpůrný obvod má v principu stejnou funkci jako v předcházejícím popsaný paralelní stykový (= podpůrný) obvod, a sice umožňovat datovou komunikaci mezi mikropočítačem a vnějším okolím. Ta se však v tomto případě děje s é r i o v ě ; odtud i jeho název: sériový vstupně-výstupní obvod. Tím je vhodný zejména pro připojení obrazovkových terminálů, tiskáren se sériovými vstupy, jednotek pružných disků a všeobecně sériových datových kanálů.

Obvod obsahuje dva úplné obousměrné kanály, tzn. čtyři sériové vstupně-výstupní porty, přičemž oba kanály mohou pracovat jak synchronně, tak i v asynchronním režimu. V posledně jmenovaném režimu mohou být přenášena slova o délce 5, 6, 7 nebo 8 bitů. Ke konci každého slova je automaticky přidáván 1,

lal/2 nebo 2 stop-bity. SIO - obdobně jako i jiný UART - generuje paritní bit (sudý, lichý, žádný), provádí detekci parity, rozpoznává chybu rámce, přeplnění a přerušení. Jsou přípustny hodinové kmitočty rovné 1x, 16x, 32x nebo 64x datové rychlosti.

Mimo již zmíněný synchronní a asynchronní provoz je též možný bitově orientovaný protokol, jako je IBM BiSync, HDLC-high data link communication, SDLC-synchroneous data link communication aj. Z80-SIO může generovat CRC kódy v každém synchronním způsobu a je programován CPJ v tradičním asynchronním formátu.

Struktura obvodu

Blokové schéma na obr. 19 zachycuje vnitřní strukturu obvodu, sestávající ze sběrnicového rozhraní, logiky vnitřního řízení, logiky řízení přerušení, řízení modemu a dvou plně duplexních kanálů A a B. Logika řízení přerušení určuje, který kanál (a tím i které periferní zařízení) má vyšší prioritu; v dané struktuře má vyšší prioritu vždy kanál A před B. Každý z obou kanálů je realizován několika osmibitovými registry včetně vyrovnávacích registrů přijímače a vysílače a dvěma šestnáctibitovými registry pro generování a ověřování bitů CRC (cyclic redundancy check), dle CRC-16 či -CCITT.

Z80-SIO je vyroben technologií N-MOS, je umístěn ve čtyřicetivývodovém pouzdru DIL s roztečí 15 mm, k napájení vyžaduje jen jedno napětí +5 V, k provozu vystačí s jednofázovým hodinovým kmitočtem. Na obrázkuje zachyceno funkční rozdělení jeho vývodů (vstupů a výstupů), rovněž tak i jejich číselné označení.

Význam jednotlivých vývodů obvodu Z80-SIO

D0 až D7 - vývody pro připojení k datové sběrnici mikropočítače. Jsou obousměrné a třístavové; přes ně se děje přenos dat z a do mikropočítače, dále pak i řídicích slov

$\overline{B/A}$ - vstup pro volbu kanálu (při úrovni log "1" je vybrán kanál B)

$\overline{C/D}$ - vstup pro volbu řízení (log "1") nebo přenos dat (log "0")

$\overline{CE}$ - vstup, aktivní v nule, uvolňující obvod (chip enable)

$\overline{M1}$ - vstup, aktivní v nule, pro příjem signálu strojního cyklu jedna

$\overline{IORQ}$ - vstup, aktivní v nule, pro příjem stejnojmenného signálu z CPJ

$\overline{RD}$ - vstup, aktivní v nule, detekující cyklus čtení CPJ

Ø - vstup systémového taktu

$\overline{RESET}$ - nulovací vstup, aktivní v nule, uzavírá oba přijímače a vysílače a uvádí řízení modemu do úrovně log "1". Při resetu jsou interupty uzavřeny, po resetu musí být zapsána řídicí slova, aby mohla data být přijímána či vysílána

IEI - vstup, aktivní v jedničce, pro přerušení (interrupt enable in)

IEO - přerušovací vstup, aktivní v jedničce. IEI a IEO formují řetězcové propojení vnějších zařízení pro prioritní řízení přerušení (daisy-chaining)

$\overline{INT}$ - výstup, aktivní v nule, indikující, že SIO požaduje přerušení od CPJ

$\overline{WAIT/READY\ A}$, $\overline{WAIT/READY\ B}$ - dva vstupy, každý pro jeden kanál. Mohou být naprogramovány buď ke spolupráci s řadičem DMA jako linky "ready", nebo k synchronizaci rychlosti dat mezi CPJ a SIO

$\overline{CTSA}$, $\overline{CTSB}$ - dva vstupy, aktivní v nule, každý pro jeden kanál. Vstupy jsou navázány na Schmittovy klopné obvody, což umožňuje přivedení i pomalých signálů (s malou strmostí nástupní hrany pulsu). Mohou být naprogramovány jako AUTO ENABLES, kdy působí ve funkci uvolňování přenosu, nebo mohou být naprogramovány pro řídicí účely

DCDA, DCDB - dva vstupy, aktivní v nule, každý pro jeden kanál (data carrier detect), působí jako uvolňovací pro příjem - tedy obdobně jako CTS-, které v modu AUTO ENABLE uvolňují vysílání

RxDA, RxDB - dva vstupy, aktivní v jedničce, určené pro příjem dat

TxDA, TxDB - dva výstupy, aktivní v jedničce, pro vysílání dat

$\overline{\text{RxCA}}$, $\overline{\text{RxCB}}$ - dva hodinové vstupy přijímače, aktivní v nule, navázané na Schmittovy klopné obvody

$\overline{\text{TxCA}}$, $\overline{\text{TxCB}}$ - dva hodinové vstupy vysílače, aktivní v nule, navázané na Schmittovy klopné obvody

Pozn.: vzhledem k danému počtu vývodů použitého pouzdra jsou k dispozici pouze dva vývody pro tři signály: $\overline{\text{TxCB}}$, $\overline{\text{RxCB}}$ a $\overline{\text{DTRB}}$. Ty hodinové jsou obvykle propojeny spolu, tedy: $\overline{\text{RxTxCB}}$, $\overline{\text{DTRB}}$ je pak samostatný výstup. Pokud však by byla požadována odchylná hodinová rychlost či fáze přijímače a vysílače, musí být hodinové signály samostatné. Proto se shledáváme s dvěmi versemi Z80-SIO, a sice SIO/0 s hodinovými vstupy spojenými a SIO/1 s oddělenými. Označení a číslování předmětných vývodů u obou verzí vypadá takto:

SIO/0	27	- $\overline{\text{RxTxCB}}$
	26	- TxDB
	25	- $\overline{\text{DTRB}}$
SIO/1	27	- $\overline{\text{RxCB}}$
	26	- $\overline{\text{TxCB}}$
	25	- TxDB

Použití, jakož i výskyt SIO/0 je však častější.

$\overline{\text{RTSA}}$, $\overline{\text{RTSB}}$ - dva výstupy, aktivní v nule, každý pro jeden kanál (request to send), indikují stav bitu RTS v synchronním i asynchronním provozu

$\overline{\text{DTRA}}$, $\overline{\text{DTRB}}$ - dva výstupy, aktivní v nule (data terminal ready). Výstupy indikují stav bitu DTR

$\overline{\text{SYNCA}}$, $\overline{\text{SYNCB}}$ - dva vývody, použitelné jako vstupy či výstupy, aktivní v nule, pro účely synchronizace. Je-li volen způsob vnější synchronizace, znakový rámec započíná při následující vzestupné hraně signály $\overline{\text{RxC}}$. Je-li volena vnitřní znaková synchronizace, působí vývody jako výstupy aktivní

během části hodinového cyklu, v němž je rozeznán synchronizační znak. Synchronizační podmínka není trvale zaznamenávána, takže výstup je aktivní vždy v těch okamžicích, kdy je rozeznána synchronizační stopa, bez ohledu na provázanost znaků. V asynchronním způsobu jsou předmětné vývody pouze vstupy pro HUNT/SYNC bity ve stavovém registru 0 a mohou být užity pro jakoukoli požadovanou vstupní funkci. Je-li uvolněn interrupt vnějšího stavu, nemůže být SYNC ponechán jako "plovoucí", neboť by tak mohlo být způsobeno nepravé přerušení. Pokud jsou vývody použity pro vnější synchronizaci, jsou aktivní s klesající hranou hodinového pulsu RxC.

Řídicí registry SIO a jejich programování

Způsob práce SIO je určen obsahy řídicích registrů. Ty musí být ovšem naprogramovány ještě před započetím činnosti SIO, ovšem některé povely (řídicí slova) a způsoby mohou být měněny během činnosti SIO. Stavový registr může být člen v kterémkoliv okamžiku. Obvod SIO může být naprogramován pro asynchronní přenos dat (nejčastěji používaný způsob), popřípadě pro synchronní v různých formátech (monosync, bisync, external sync detect) se čtyřmi způsoby přerušení (1 - bez přerušení, 2 - s přerušením na první znak, 3 - s přerušením na každý znak s generováním vektoru, 4 - s přerušením na každý znak bez generování vektoru), dále v modu SDLC.

Nyní je již zřejmé, že Z80-SIO je vícefunkční stykový obvod navržený výrobcem k uspokojení různých požadavků na sériovou komunikaci, i když jeho hlavní role spočívá v sériově-paralelní a paralelněsériové konverzi dat.

Používá se série povelů (řídicích slov) k tzv. počáteční iniciaci, tj. určení způsobu činnosti SIO, počtu bitů ve znaku, počtu stop-bitů, synchronizačních znaků, přenosové rychlosti apod.

Každý z obou kanálů obsahuje řídicí registry, jež musí být naplněny softwareově; volba kanálu (B/$\overline{A}$) a stavu řízení (C/$\overline{D}$) se děje obvykle prostřednictvím vedení adresové sběrnice. Z přiložených úrovní na citovaných vstupech vyplývá pak funkce, viz níže:

C/$\overline{D}$	B/$\overline{A}$	funkce
1	0	pro řídicí slovo kanálu A
1	1	pro řídicí slovo kanálu B
0	0	pro přenos dat kanálem A
0	1	pro přenos dat kanálem B

Z80-SIO obsahuje osm registrů zápisu (write), jejichž naplněním je dána funkční schopnost každého kanálu. Všechny tyto registry, s výjimkou prvního (write register 0) vyžadují dvoubytovou informaci k správnému naprogramování. První slabika (byte) svými třemi nejméně významnými bity (D0 až D2) určuje programovaný registr zápisu, zatímco druhá slabika je již aktuální ř í d i c í slovo, které je vloženo do daného registru.

Registr zápisu 0 je zvláštním případem: RESET - ať již jako vnitřní povel či vnější vstup - vykonává jeho iniciaci. Všechny základní povely (CMD0 až CMD2) a kontrolní znaky CRC0 a CRC1 mohou být dosaženy jedním bytem. Na dále uvedené tabulce jsou zakresleny formáty registrů zápisu včetně vyznačení významů jejich jednotlivých bitů.

Mimo tyto registry zápisu obsahuje ještě Z80-SIO tři čtecí registry (read registers), z jejichž obsahu může být přečten status každého kanálu. Stavová informace (= status) obsahuje podmínky chyb, přerušovací vektor a standardní komunikační stykové signály protokolu. K přečtení obsahu vybraného čtecího registru musí být softwareem nejprve zapsán do SIO byte obsahující označení registru (D0 až D2) stejným způsobem jako u registru zápisu. Pak - pomocí operace READ - může být přečten stav adresovaného čtecího registru centrál-

ní procesorovou jednotkou. Významné je, že programátor - po označení registru - má volnost rozhodnutí, zda bude předmětný registr testovat (číst) či provádět jeho iniciaci. V další tabulce jsou uvedeny formáty čtecích registrů včetně vyznačení významu jejich jednotlivých bitů. Bližší podrobnosti programování registrů nalezne zájemce v literatuře /37/ a /53/.

II.3. Z80-CTC

V některých případech použití mikropočítačů pro řízení je nutné dodržení určitých časových požadavků. Tak ku příkladu při styku s vnějším zařízením bez kvitování musí po uplynutí lhůty následovat přerušení vyčkávací smyčky, aby tak nebyl mikropočítač blokován na neurčitý čas. Rovněž při řízení procesů musí být vstupní kanály dotazovány nebo jsou přes ně vysílány informace v určitých časových úsecích. Vyhovění časovým nárokům sice může být realisováno mikroprocesorem, a sice odpracováním speciálních časovacích programů. Během takto vyplněného času však není CPJ k dispozici pro vlastní uživatelský program, a mimoto je dodržení exaktních časových podmínek těžko uskutečnitelné, neboť musí být respektován průběhový čas uživatelského programu. Ten navíc nemusí být konstantní, neboť mnohdy je závislý na vstupních datech - a tím má i variabilní délku.

Z těchto důvodů vyplynula nutnost vyvinutí speciálních časovacích obvodů (např. u fy Intel typ 8253) - /54/, jež mohou být naprogramovány řídicím slovem z procesoru. Jejich činnost pak je dále již nezávislá na CPJ. Zpravidla načítají hodinové impulsy či jejich n-tiny, a to až k dosažení předem dané konečné hodnoty, čímž uplyne definovaný časový interval. Dosažení konečné hodnoty je indikováno zpravidla přerušením. Mnohdy jsou takovéto časovače programovatelné též jako čítače; místo hodinového taktu jsou na jejich vstupy přiváděny impulsy z vnějšího okolí (odpovídající sledovaným událostem), jež jsou postupně načítány. Aktuální stav čítače může být kdykoli - prostřednictvím příkazů v/v - předán procesoru.

Časovač-čítač slouží tedy ke koordinovanému průběhu a kontrole řízeného procesu, k realizaci vazeb programových přerušení a jako programovatelné časové normály mikropočítačů (např. jako generátor přenosové rychlosti pro SIO apod.). Jedním z dalších podpůrných obvodů ze základní čtveřice série Z80 je Z80-CTC = čítačový a časovací programovatelný obvod.

Z80-CTC je integrovaný obvod LSI, umístěný v dvacetiosmivývodovém pouzdře DIL s roztečí 15 mm, vyrobený technologií N-MOS, vyžadující k svému napájení jedno napětí +5 V a jednofázový systémový takt Ø, jehož všechny vstupy a výstupy jsou TTL kompatibilní, (obr. 21).

Struktura obvodu

Blokové schéma Z80-CTC je na obr. 22. Skládá se ze sběrnicového rozhraní - přímo připojitelného k datové sběrnici mikroprocesoru Z80 - logiky vnitřního řízení, logiky řízení přerušení (tedy obdobně jako předcházející stykové obvody Z80-PIO a Z80-SIO) a sady čtyř kanálů vzájemně nezávislých čítačů/časovačů, identifikovatelných číselným označením 0 až 3.

CTC má schopnost generovat přerušovací vektor pro každý separátní kanál. Citované čtyři kanály mohou být propojeny do prioritního řetězce, přičemž kanál 0 má vždy nejvyšší prioritu pro přerušení.

Každý kanál se skládá ze dvou registrů, dvou čítačů a řídicí logiky; registry jsou: osmibitový registr časové konstanty a osmibitový registr řízení kanálu, čítače jsou: osmibitový sestupný čítač a osmibitový dělič (prescaler).

Registr řízení kanálu

Jak plyne z výše uvedeného, CTC obsahuje čtyři tyto registry. Volba kteréhokoliv z nich pro zápis řídicího slova se děje prostřednictvím vstupů CS_0 a CS_1 takto:

kanál 0 - $CS_0 = 0$, $CS_1 = 0$, kanál 1 - $CS_0 = 1$, $CS_1 = 0$,
kanál 2 - $CS_0 = 0$, $CS_1 = 1$, kanál 3 - $CS_0 = 1$, $CS_1 = 1$

Dělič (prescaler) je osmibitový registr, použitelný jen při časování (timer mode), který může být naprogramován mikroprocesorem přes registr řízení kanálu. Programuje se dělení vstupního hodinového taktu Ø, a to šestnácti nebo dvěstěpadesáti šesti. Výstup z děliče je pak veden na vstup sestupného čítače (down counter), který - jak při iniciaci, tak vždy po dosažení nulového obsahu - je automaticky naplněn obsahem registru časové konstanty. Vždy, když čítač po odečítání dosáhne nulového obsahu, nabyde jeho výstup ZC (zero count) úrovně log "1".

Registr časové konstanty (time constant register) - TC je osmibitový registr, používaný jak při čítání (counter mode), tak i při časování. Programován je mikroprocesorem hned po kanálovém řídicím slovu, a sice časovou konstantou 1 až 256. Tento registr uloží naprogramovanou hodnotu do sestupného čítače v okamžiku iniciace nebo při jeho nulovém obsahu. Je-li vložena do registru TC nová konstanta, zatímco čítač odečítá, nová hodnota je platná až po ukončení čítání (časování).

Sestupný čítač je osmibitový registr použitelný v obou zmíněných modech. Je naplněn při iniciaci, dále pak přes TC při dosažení nuly. Sestupný čítač je dekrementován (tj. jeho obsah je zmenšen o jedničku) při každé hraně vnějšího hodinového taktu, ať již systémového taktu Ø (při odečítání), nebo děleného taktu z děliče (při časování). V kterémkoli okamžiku - provedením instrukce I/O READ - při adresaci příslušného kanálu CTC může mikroprocesor zjistit stav čítače.

Kanály CTC mohou být naprogramovány ke generování požadavku na přerušení při každém dosažení nulového obsahu čítače. V kanálech 0, 1 a 2 při dosažení podmínky vynulování objeví se signalizační impuls na odpovídajícím vývodu ZC. Avšak vzhledem k omezenému počtu vývodů není tomu tak u kanálu 3.

Logika řízení přerušení zaručuje činnost CTC v souladu se systémem prioritního přerušování mikroprocesoru Z80, kde priorita jakékoliv periférie je determinována jejím fyzickým umístěním v konfiguraci. Signály IEI a IEO zajišťují obsluhu v soustavě-řetězci periférií.

Účelem generovaného požadavku na přerušení u CTC je - jako i u jiných periférií - přimět CPJ k vykonání obslužné rutiny přerušení. Dle systému prioritního přerušování Z80, zařízení či kanály s nižší prioritou nemohou se domáhat přerušení, pokud zařízení s vyšší prioritou je právě obsluhováno. Naproti tomu pro periférie či kanály s vyšší prioritou je přerušení obsluhy možné.

Jsou-li kanály CTC naprogramovány ke generování požadavku na přerušení, pak - má-li dojít k realizaci - musí CPJ pracovat v modu 2, viz stať 1.5 - Žádost o přerušení. Po přijetí požadavku (žádosti) vyšle mikroprocesor potvrzení, načež logika řízení přerušení CTC určí kanál s nejvyšší prioritou dožadující se přerušení. Tehdy se stane vstup IEI časovače CTC aktivní (tj. nabyde úrovně H = log "1"), čímž indikuje, že žádné další zařízení vyšší priority v řetězci není obsluhováno; na datovou sběrnici je umístěn osmibitový přerušovací vektor. Jeho pět bitů vyššího řádu bylo však do CTC zapsáno již při iniciaci - další dva bity jsou uloženy přerušovací logikou jako binární kód korespondující s kanálem dožadujícím se přerušení, a konečně poslední, nejméně významný bit D0 je nulový, viz schéma:

D7	D6	D5	D4	D3	D2	D1	D0	
V7	V6	V5	V4	V3	X	X	0	
kde X = 0 nebo 1, tedy					0	0	-	kanál 0
					0	1	-	kanál 1
					1	0	-	kanál 2
					1	1	-	kanál 3

Přerušovací vektor se použije k nastavení ukazatele na místo v paměti, kde je v tabulce uložena adresa obslužné rutiny přerušení. Vektor representuje nižší byt, zatímco CPJ přečte obsah z registru I jako vyšší byte šestnáctibitového ukazatele. Takto označená adresa místa v paměti obsahuje nižší byte, následující adresa (vyšší) obsahuje vyšší byte adresy, na níž již leží první instrukce, tzn. započíná vlastní obslužná rutina přerušení. Takto tedy v modu 2 jediný osmibitový vektor, převzatý z CTC, způsobí nepřímé volání do jakéhokoliv místa v paměti.

V systému Z80 je platná konvence, že všechny adresy přerušovacích rutin mají nižší byte v sudé buňce paměti, vyšší byte pak v následující liché buňce. Nejméně významný bit je tedy vždy sudý a nulový.

Instrukce RETI je používána na konci každé přerušovací rutiny k nulování výstupu IEO za účelem správného řízení větvených přerušení různých priorit. CTC monitoruje (sleduje) datovou sběrnici a dekóduje citovanou instrukci, jakmile se objeví. Tak je časovač CTC informován o ukončení přerušovací rutiny mikroprocesoru, aniž by byla nutná další komunikace s CPJ.

Význam jednotlivých vývodů Z80-CTC

D0 až D7 - třístavová obousměrná datová sběrnice; D0 je nejméně významné datové vedení

CS0, CS1 - dva vstupy, aktivní v jedničce, pro výběr kanálu 0 až 3

$\overline{CE}$ - vstup, aktivní v nule, uvolňující obvod. Při uvolnění může CTC přijmout řídicí slovo, vektor přerušení či hodnotu časové konstanty během cyklu "I/O write" z CPJ, nebo vyslat obsah sestupného čítače do CPJ během cyklu "I/O read". V mnoha aplikacích je signál pro $\overline{CE}$ odvozován z osmi méně významných bitů adresové sběrnice pro kteroukoliv z v/v adres bran odpovídajících citovaným čtyřem kanálům CTC

Ø - vstup systémového taktu (jednofázový hodinový takt je použit k synchronisaci určitých vnitřních signálů)

$\overline{M1}$ - vstup, aktivní v nule, pro příjem signálu strojního cyklu jedna z CPJ. Je-li $\overline{M1}$ aktivní a též $\overline{RD}$ je aktivní, pak CPJ přináší instrukci z paměti. Je-li však $\overline{M1}$ aktivní a též $\overline{IORQ}$, CPJ potvrzuje přerušení s vybídnutím CTC k uložení přerušovacího vektoru na datovou sběrnici Z80, pakliže používá prioritního řetězce a jeden z jeho kanálů se domáhá přerušení

$\overline{IORQ}$ - vstup, aktivní v nule, pro příjem stejnojmenného signálu z CPJ. Signál $\overline{IORQ}$ se používá v konjunkci s $\overline{CE}$ a $\overline{RD}$ k přenosu dat a řídicích slov kanálů mezi CPJ a CTC. V průběhu zápisového cyklu (write cycle) musí být aktivní $\overline{IORQ}$ a $\overline{CE}$, $\overline{RD}$ pak neaktivní. CTC totiž nedostává specifický zápisový signál, ale generuje si jej vnitřně inverzí z platného $\overline{RD}$. V čtecím cyklu CTC musí být aktivní signály $\overline{IORQ}$, $\overline{CE}$ a $\overline{RD}$ k přenosu obsahu sestupného čítače na datovou sběrnici

$\overline{RD}$ - vstup, aktivní v nule, indikující cyklus čtení CPJ

IEI - vstup, aktivní v jedničce, pro přerušení. Pomáhá vytvořit systémový široký prioritní řetězec, který stanoví priority, jestliže více než jedno zařízení v systému má schopnost požadovat přerušení

IEO - výstup, aktivní v jedničce

$\overline{INT}$ - výstup, aktivní v nule, indikující požadavek přerušení; u CTC tedy v okamžiku dosažení nulového obsahu sestupného čítače (jednoho ze čtyř)

$\overline{RESET}$ - nulovací vstup, aktivní v nule. Zastavuje čítání ve všech kanálech a nuluje uvolňovací bity přerušení ve všech řídicích registrech. Uvádí výstupy ZC/TO do neaktivních stavů a výstupy datové sběrnice do vysokoimpedančního stavu

CLK/T_i - vstup, jehož aktivitu volí uživatel - low/high. Čtyři vstupy odpovídají čtyřem kanálům CTC. V čítacím modu aktivní hrana na tomto vstupu dekrementuje sestupný čítač; uživatel může zvolit čelní či týlovou hranu pulsu jako aktivující

ZC/TO_i - výstup, aktivní v jedničce (zero count-timeout). Tři výstupy, odpovídající kanálům 0, 1 a 2. Jak v čítacím, tak i v časovacím modu je indikován nulový stav sestupného čítače jedničkovým pulsem na daném kanálovém výstupu ZC/TO_i, kde i = 0, 1 nebo 2

Programování časovače Z80-CTC

Před započetím jakékoliv operace CTC kanálů musí být mikroprocesorem vloženo do CTC řídicí slovo kanálu a datové slovo časové konstanty. Pakliže v řídicím slově kanálu je sedmým bitem připuštěno přerušení (D7 = 1), musí být vložen ještě do CTC přerušovací vektor, který je vyhovující pro všechny čtyři kanály.

1. Uložení řídicího slova kanálu do registru

Pomocí CS0 a CS1 je volen požadovaný kanál - obvykle přes adresové linky A0 a A1 a řídicí slovo je zapsáno v cyklu I/O write. Charakteristické pro řídicí slovo je, že jeho nejméně významný bit D0 je vždy jedničkový, viz schéma:

D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	X	X	X	1

kde X = 0 nebo 1

Ostatních sedm bitů svou zvolenou úrovní určují operační způsoby a podmínky, což bude dále ukázáno:

D7 = 1 : požaduje se generování žádosti o přerušení vždy při nulovém stavu sest. čítače jak v čítacím, tak i v časovacím způsobu

D7 = 0 : přerušení není dovoleno

D6 = 1 : požaduje se čítací způsob (dělič - prescaler - není použit)

D6 = 0 : požaduje se časovací způsob. Dělič je buzen hodinovým kmitočtem systému Ø, výstup děliče budí sestupný čítač. Na jeho výstupu ZC/TO je sled pulsů o perio-

dě dané součinem t_c.P.TC, kde t_c je perioda hodinového taktu Ø, P je dělicí činitel 16 nebo 256 a TC je časová konstanta 1 až 256

D5 = 1 : určuje dělicí faktor - 256 (definuje se jen pro časování)

D5 = 0 : určuje dělicí faktor - 16 (definuje se jen pro časování)

D4 = 1 : pozitivní hrana pulsu určuje start časování, nebo pozitivní hrana pulsu dekrementuje sestupný čítač

D4 = 0 : negativní hrana pulsu startuje operaci časování, nebo negativní hrana pulsu dekrementuje sestupný čítač

D3 = 1 : jako u D5 definuje se též jen pro časování. Externí spouštěcí puls je platný pro start operace časování po nástupné hraně T2 strojního cyklu následujícího po uložení časové konstanty. Dělič je dekrementován o dva hodinové takty později po týlu impulsu - jinak o tři takty. Jakmile je časovač odstartován, běží volně rychlostí určenou registrem časové konstanty

D3 = 0 : definuje se jen pro časování. Časovač začíná operaci při náběžné (stoupající) hraně T2 strojního cyklu následujícího po tom, jenž uložil časovou konstantu

D2 = 1 : označuje, že následující slovo k zapsání do kanálu bude datové slovo časové konstanty

D2 = 0 : nebude následovat datové slovo čas. konstanty. Z naprogramování bitu D2 do tohoto stavu vyplývá, že řídicí slovo kanálu hodlá operativně předatovat stav kanálu. (Kanál totiž nemůže operovat bez správně naprogramovaného datového slova v registru časové konstanty; nahození bitu 2 v řídicím slově kanálu představuje jedinou cestu zápisu do registru časové konstanty)

D1 = 1 : resetování kanálu. Kanál zastavuje čítání či časování. Jsou-li nastaveny bity D2 = 1 a D1 = 1, pak kanál pokračuje v operaci po uložení časové konstanty

D1 = 0 : kanál pokračuje v běžné operaci.

2. Uložení datového slova časové konstanty do TC registru

Kanál nemůže započíst činnost v časování či čítání, jestliže ještě nebylo mikroprocesorem vloženo datové slovo časové konstanty do příslušného registru CTC. Datové slovo CT následuje po řídicím slovu kanálu za předpokladu, že bit D2 = 1 (v řídicím slově). Datové slovo časové konstanty tvoří celé číslo v rozsahu 1 až 256 (jsou-li všechny bity slova rovny nule, je tento stav interpretován jako číslo 256). Tvar datového slova je:

D7	D6	D5	D4	D3	D2	D1	D0
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0

kde TC_i = 0 nebo 1 při i = 0 až 7.

3. Uložení přerušovacího vektoru do registru

Časovač Z80-CTC byl navržen k součinnosti se Z80-CPJ při tzv. druhém způsobu přerušení (mode -2), při němž vektor přerušení působí (po vydání) jako dolní byte ukazovatele adresy obslužné rutiny přerušení.

Tvar slova s přerušovacím vektorem je následující:

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	X	X	0

, kde X = 0 nebo 1, podle zvoleného kanálu. Charakteristické zde je, že nejméně významný bit D0 je vždy nulový. (Tím je označeno pro CTC, že se jedná o přerušovací vektor, který má být uložen do příslušného registru, a nikoliv řídicí slovo, které končí jedničkovým bitem D0.)

V3 až V7, tj. horních pět bitů, zadává uživatel, rovněž tak D0 = 0. Bity D2 a D1 pro zadání jsou nulové - tak jako kdybychom programovali pouze kanál 0 CTC. Ovšem při vydání vektoru časovač CTC automaticky sám doplní úrovně D2 a D1 ve shodě s binární adresou toho kanálu, který vydal žádost o přerušení. Z uvedeného již vyplývá, proč je část slova s přerušovacím vektorem V7 až V3 stejná pro všechny kanály.

Závěrem této stati uveďme příklad zapojení jednoduchého mikropočítačového systému podle /31/, kde kromě dvou obvodů PIO se shledáváme s jedním časovačem Z80-CTC, adresovaným a uvolňovaným linkami A0, A1 a A2. Z praktického hlediska je na předmětném schématu zajímavé jednak připojení dynamické paměti RAM pomocí běžných obvodů TTL MH 7404 a UC 7402, jednak přerušovací řetězec programovatelných podpůrných obvodů pracujících spolu s CPJ v modu -2. Priorita žádostí o přerušení je dána pořadím periferního obvodu v řetězci vzniklém propojením vstupu IEI následujícího obvodu s výstupem IEO předcházejícího obvodu, přičemž u obvodu s nejvyšší prioritou je vstup IEI připojen pevně na úroveň log "1", viz obr. 23.

II.4. Z80-DMA

Z80-DMA (direct memory access) je obvod přímého přístupu do paměti. Přímý přístup do paměti je operace velmi užitečná ve výpočetní technice, neboť dovoluje přímý přenos dat z externího zařízení - periférie do paměti počítače, aniž by však k tomu bylo zapotřebí součinnosti CPJ. Tím je přenos dat značně urychlen; uvážíme-li, že při běžném způsobu přenosu jednoho každého datového slova do specifikovaného místa paměti je nutno nejprve vykonat instrukci přesunu vstupního data do střadače CPJ a jako druhou instrukci pak přesun informace ze střadače do adresované buňky paměti, pak při způsobu DMA provádění zmíněných instrukcí zcela odpadá. DMA - přímý přístup do paměti - tedy dovoluje přímé umístění dat z periférie do vyhrazené oblasti paměti; při větším počtu dat přímo přenášených je časová úspora proti běžnému způsobu značná. (Citovaný způsob se často využívá při komunikaci mezi pevnými disky typu Winchester o značné datové kapacitě, či systémovými disketami a operační velkokapacitní pamětí mikropočítačového systému, což vede k zrychlení obslužného provozu, apod.)

Z80-DMA je programovatelný stykový obvod, který zajišťuje či generuje všechny adresovací, časovací a řídicí signály k uskutečnění přenosu bloku dat mezi dvěma bránami v systému

opírajícím se o Z80-CPU. Brány mohou představovat operační systémovou paměť nebo jakékoliv vstupně-výstupní periferní zařízení systému.

Struktura obvodu

Blokové schéma na obr. 24 zobrazuje vnitřní strukturu obvodu, sestávající z více částí provázaných vnitřní datovou sběrnicí, které budou v dalším stručně popsány. Obvod DMA je vyroben technologií N-MOS a je umístěn ve čtyřicetivývodovém pouzdru DIL s roztečí 15 mm, k napájení vyžaduje jen jedno napětí +5 V. K provozu vystačí s jednofázovým hodinovým kmitočtem Ø. Další obr. 25 zachycuje funkční rozdělení vývodů pouzdra DMA, rovněž tak i jejich číselné označení.

Jednotlivé části obvodu jsou:
- sběrnicové rozhraní, umožňující přímé připojení obvodu k systémovým sběrnicím mikroprocesoru Z80
- řídicí logika a řídicí registry, nastavující třídu, způsob práce a ostatní základní řídicí parametry DMA
- adresovací, čítací a pulsní obvody, generující správné adresy portů pro operace čtení a zápisu při zajištění adresové inkrementace či dekrementace. Obvod čítání slabik kontroluje počet přenášených či hledaných bytů; pulsní obvod vydává v určitý okamžik signální impuls
- časovací obvod, dovolující uživateli specifikovat časování operace "čtení/zápis" pro oba adresované porty kanálu
- srovnávací obvod (match circuity), obsahující srovnávací a maskovací registr pro porovnání bitů slabiky
- obvod přerušení a žádosti o sběrnici, obsahující řídicí registr, logiku prioritního dekódování a registr přerušovacího vektoru, a
- stavový registr, obsahující běžný status DMA.

Dále popsané registry výše zmíněných částí DMA jsou k dispozici uživateli k naprogramování či indikování jejich činnosti; jsou to:

řídicí registry - obsahují řídicí informace pro DMA, jako např. kdy zahájit přerušení či generování pulsu, jaký způsob či třídu operace provádět apod. (jen zápis - 8 bitů)

časovací registry - obsahují časovací parametry pro obě brány (jen zápis - 8 bitů)

registr vektoru přerušení - obsahující osmibitový vektor, jenž DMA umístí na datovou sběrnici po obdržení signálu $\overline{\text{IORQ}}$ během sekvence potvrzení přerušení, pokud je to zařízení s nejvyšší prioritou dožadující se přerušení (čtení i zápis - 8 bitů)

registr délky bloku - obsahuje celkovou délku bloku dat, jež má být vyhledána nebo přenesena (jen zápis - 16 bitů)

čítač slabik - čítající počet přenášených či vyhledávaných slabik. Po příkazu "load" či "continue" je čítač slabik vynulován. Potom jej každá operace přenosu inkrementuje, až se rovná obsahu registru délky bloku. V tom okamžiku je nahozen "konec bloku" ve stavovém registru a operace je zastavena. Je možno naprogramovat též generování přerušení (jen čtení, 16 bitů)

srovnávací registr - obsahuje slabiku, pro níž je hledán protějšek v operaci vyhledávání (search) - (jen zápis - 8 bitů)

maskovací registr - obsahuje osmibitovou masku určující, které bity mají být přezkušovány pro protějšek (jen zápis - 8 bitů)

registr adresy startu/port A a port B - obsahuje adresy startu pro oba porty, zahrnuté v operacích přenosu. V operaci "jen vyhledávání" je specifikována pouze adresa jednoho portu. Pouze adresy startu paměti vyžadují udání horní a dolní slabiky, vstupně-výstupní brány jsou obecně adresovány dolními osmi bity (= dolní slabikou). V tom případě adresa obsažená v registru je pevná adresa (jen zápis - 16 bitů každá)

adresové čítače/port A a port B - jsou naplněny obsahem korespondujících registrů startovní adresy, kdykoliv je započato vyhledávání či přenos příkazy "load" či "continue". Jsou inkrementovány, dekrementovány nebo zůstávají pevné - dle naprogramování (jen čtení - 16 bitů každý)

řídicí registr pulsů - obsahuje programem určenou délku bloku v slabikách, po níž DMA generuje signální puls na vývodu $\overline{INT}$. (Protože tento puls se objevuje při aktivních signálech $\overline{BUSRQ}$ a $\overline{BUSAK}$, neinterpretuje ho CPJ jako požadavek na přerušení. Místo toho je signál používán pro komunikaci s periferními vstupně-výstupními zařízeními, (jen zápis - 16 bitů každý)

stavový registr - nepoužívá bitů D2, D6 a D7. Bity D5 (konec bloku), D4 (match), D3 (nevyřízené přerušení), D1 (aktivní signál ready) a D0 (platná adresa zápisu) - pokud jsou nastaveny (= log "1"), indikují uvedené funkce

Význam jednotlivých vývodů obvodu Z80-DMA

D0 až D7 - vývody pro připojení k datové sběrnici mikropočítače. Jsou obousměrné, třístavové. Povely z CPJ, status DMA a data z paměti či periférií jsou přenášena přes tyto vývody

A0 až A15 - adresová sběrnice systému. Všech šestnáct adresovacích linek je použito obvodem DMA k adresování operační paměti či vstupně-výstupních portů

Ø - vstup systémového taktu (hodinový kmitočet)

$\overline{M1}$ - vstup, aktivní v nule, pro příjem signálu strojního cyklu jedna (machine cycle One) z CPJ

$\overline{IORQ}$ - vstup/výstup, aktivní v nule, indikující požadavek v/v komunikace systémové sběrnice

$\overline{MREQ}$ - vstup/výstup, aktivní v nule, indikující požadavek paměťové komunikace se systémovou sběrnicí

$\overline{RD}$ - vstup/výstup, aktivní v nule, indikující čtení z a do systémové sběrnice

$\overline{WR}$ - vstup/výstup, aktivní v nule, inidkující zápis do a z systémové sběrnice

$\overline{CE/WAIT}$ - vstup, aktivní v nule, uvolňující obvod. Může být též programován do funkce "wait" v průběhu signálu $\overline{BAI}$ = log "0"

$\overline{BUSRQ}$ - vstup/výstup, aktivní v nule, vznáší požadavek řízení na adresovou, datovou a řídicí/stavovou sběrnici

$\overline{BAI}$ - vstup, aktivní v nule (bus acknowledge in), indikující, že systémové sběrnice jsou uvolněny pro řízení DMA

$\overline{BAO}$ - výstup, aktivní v nule (bus acknowledge out) formující společně s $\overline{BAI}$ řetězcové propojení pro systémové prioritní řízení

$\overline{INT}$ - výstup, aktivní v nule, indikující požadavek na přerušení

IEI - přerušovací vstup, aktivní v jedničce

IEO - přerušovací výstup, aktivní v jedničce a formující společně s IEI řetězcové propojení pro prioritní řízení přerušení

RDY - vstup s programovatelnou aktivitou v nule nebo jedničce, monitorovaný obvodem DMA za účelem určení, kdy periferní zařízení přidružené k portu DMA je připraveno k operaci čtení či zápisu.

Operační způsoby a třídy

Obvod DMA může být naprogramován v jednom z možných čtyř operačních způsobů. Jsou to:

a) po jednom byte (single byte at a time), kdy řízení je vráceno mikroprocesoru vždy po přenosu jednoho každého bytu

b) přetržitě (burst), kdy operace přenosu probíhá tak dlouho, pokud vstup RDY obvodu DMA je aktivní. Řízení se vrací mikroprocesoru, když RDY přestane být aktivní, na konci bloku či při rovnosti určitých bitů (match), pokud jsou programovány

c) spojitě (continous), kdy úplný přenos či vyhledávání bloku

dat je dokončeno před předáním řízení mikroprocesoru, a

d) transparentně, kdy operace DMA se děje během osvěžovacích period dynamických pamětí, tedy bez viditelných časových ztrát CPJ

Obvod DMA má tři třídy operací. Jsou to:

1. třída přenosu (transfer only)
2. třída vyhledávání (search only), a konečně
3. kombinovaná třída vyhledávání a přenosu

Během přenosu jsou data čtena z jednoho portu a zapisována do druhého portu, slabiku po slabice. Porty mohou být naprogramovány do funkce operační paměti či periferního vstupně-výstupního zařízení. Tak je možné, že blok dat může být přepsán (přenesen) z jedné části hlavní (= operační) paměti do jiné její části, nebo z periférie do hlavní paměti.

Během vyhledávání jsou data pouze čtena a srovnávána slabiku po slabice s obsahy dvou vnitřních registrů DMA (srovnávacího a maskovacího), z nichž první obsahuje srovnávací slabiku (match byte) a druhý zvolenou maskovací slabiku, jež dovoluje porovnání pouze určitých bitů. Jestliže je některá slabika z vyhledávaných (prohlížených) dat shodná, je nahozen vnitřní stavový DMA bit. Pokud je tato skutečnost programově zabezpečena, obvod DMA zastaví operaci vyhledávání a nebo generuje přerušení.

V poslední operační třídě se jedná o kombinované vyhledávání s přenosem. Při takovémto druhu operace je přenášen blok data podle 1., pokud se nedojde ke shodnosti slabik. Pak - jako při vyhledávání - je operace zastavena nebo/a je generováno přerušení.

Adresování obou portů obvodu DMA je buď pevné, nebo sekvenční, s inkrementací, nebo dekrementací od startovní adresy. Délka operace (počet slabik) je specifikována naprogramovaným obsahem registru délky bloku. Obvod DMA může adresovat bloky dat o délce až do 64 KB. Během přenosu jsou generovány dvě separátní adresy, a sice jedna během čtecího cyklu a jedna během zápisového cyklu.

Jakmile je obvod DMA naprogramován, může být uvolněn k činnosti. V okamžiku uvolnění (enable), kdy vstup RDY se stává aktivní, požaduje DMA sběrnici úrovní log "0" na výstupu $\overline{\text{BUSRQ}}$. Mikroprocesor to kvituje signálem $\overline{\text{BUSACK}}$ (= 0), který je obvykle přiveden na vstup $\overline{\text{BAI}}$ obvodu DMA. Jakmile DMA přijme signál $\overline{\text{BAI}}$, odstartuje svoji programovou operaci; po jejím ukončení změní DMA úroveň na výstupu $\overline{\text{BUSRQ}}$ na log "1".

Nastavení DMA

Z předchozího vyplývá, že obvod Z80-DMA může pracovat různými způsoby v odlišných operačních třídách; volba způsobu a třídy pochopitelně závisí na přání uživatele, který musí obvod DMA příslušně naprogramovat.

Pokud je obvod DMA uvolněn, řídí systémové sběrnice a přenos dat mezi oběma svými porty. Není-li uvolněn, řízení a přenos dat není vykonáván. Nastavení (= naprogramování) obvodu DMA povelovými slovy se může dít jak při uvolněném, tak i při neuvolněném stavu, v němž vytrvá až do následného uvolnění příslušným signálem $\overline{\text{CE}}$, a to po jeho naprogramování. To se uskutečňuje - jako u předcházejících podpůrných obvodů - řídicími/povelovými osmibitovými slovy (v daném případě tedy slabikami) vydanými mikroprocesorem prostřednictvím výstupních instrukcí OUT.. .

Po přiložení provozního napětí na obvod DMA či po jeho vynulování (reset) nachází se DMA vždy v neuvolněném stavu, v němž nemůže požadovat přístup na sběrnice, a zahájit tak přenos dat, ani započít přerušení. Teprve po přijetí povelových slov, obsahujících informace určené pro řídicí a další registry, může DMA započít svoji naplánovanou činnost.

V následujícím jsou uvedeny tvary šesti různých povelových slov, spolu s významy jejich některých bitů. Slova jsou rozdělena do dvou skupin, a sice skupiny 1 (slovo 1A a 1B) obsahující dvě povelová slova a skupiny 2 obsahující čtyři slova (2A až 2D).

Povelové slovo 1A

D7	D6	D5	D4	D3	D2	D1	D0
0	BLU	BLL	SAU	SAL	X	C1	C0

C1-C0 = 0 0 není povoleno, určuje slovo 1B
= 0 1 přenos
= 1 0 vyhledávání
= 1 1 vyhledávání a přenos

D2 = 1 z portu A je čteno, do portu B je zapisováno (pokud ovšem není zvoleno vyhledávání, při němž port B není nikdy adresován)

D2 = 0 z portu B je čteno, do portu A je zapisováno (pokud ovšem není zvoleno vyhledávání, při němž port A není nikdy adresován)

D3 = SAL - port A, startovací adresa (nižší) následuje
D4 = SAU - port A, startovací adresa (vyšší) následuje

D5 = BLL - následuje délka bloku (nižší)
D6 = BLU - následuje délka bloku (vyšší)

D7 = 0 - nulový osmý bit specifikuje skupinu 1

Povelové slovo 1B

D7	D6	D5	D4	D3	D2	D1	D0
0	TB	AF	I/D	P/M	A/B	0	0

D0 = D1 = 0 specifikuje povelové slovo 1B
D2 = 0 programuje port B
D2 = 1 programuje port A
D3 = 0 port adresuje hlavní paměť
D3 = 1 port adresuje v/v periférii
D4 = 0 po každé slabice je adresa dekrementována
D4 = 1 po každé slabice je adresa inkrementována
D5 = AF pevná adresa
D6 = TB následuje řídicí slovo časování (timing control byte)

Povelové slovo 2A

D7	D6	D5	D4	D3	D2	D1	D0
1	EN	EI	MB	MS	SC	0	0

D0 = D1 = 0 specifikuje povelové slovo 2A
D7 = 1 jedničkový osmý bit specifikuje skupinu 2
D2 = SC (stop on compare) zastav při rovnosti
D3 = MS následuje maskovací byte
D4 = MB následuje srovnávací byte
D5 = EI uvolnění přerušení
D6 = EN uvolnění čipu

Povelové slovo 2B

D7	D6	D5	D4	D3	D2	D1	D0
1	M1	M0	IC	BAU	BAL	0	1

D0 = 1, D1 = 0 specifikuje povelové slovo 2B
D2 = BAL následuje nižší adresa portu B
D3 = BAU následuje vyšší adresa portu B
D4 = IC následuje řídicí slovo přerušení
D6-D5 = M1-M0 = 0 0 po jednom byte
0 1 spojitě
1 0 přetržitě
1 1 transparentně

Povelové slovo 2C

D7	D6	D5	D4	D3	D2	D1	D0
1	-	AR	WM	RD	-	1	0

D0 = 0, D1 = 1 specifikuje povelové slovo 2C
D3 = 0 výstup RDY aktivní v nule
D3 = 1 výstup RDY aktivní v jedničce
D4 = 0 jen CE
D4 = 1 CE a WAIT jsou multiplexovány na tomtéž vývodu č. 16
D5 = 0 bez účinku

D5 = 1 automaticky opakuje celou operaci po dosažení konce bloku

Povelové slovo 2D

D7	D6	D5	D4	D3	D2	D1	D0
1	f4	f3	f2	f1	f0	1	1

D1 = D0 = 1 specifikuje povelové slovo 2D

hex	D6	D5	D4	D3	D2	funkce
C3	1	0	0	0	0	reset
C7	1	0	0	0	1	reset port A timing
CB	1	0	0	1	0	reset port B timing
CF	1	0	0	1	1	load
D3	1	0	1	0	0	continue
AB	0	1	0	1	0	enable int
AF	0	1	0	1	1	disable int
A3	0	1	0	0	0	reset int
87	0	0	0	0	1	enable DMA
83	0	0	0	0	0	disable DMA
BB	0	1	1	1	0	read byte follows
A7	0	1	0	0	1	reset RD
BF	0	1	1	1	1	RD status
B3	0	1	1	0	0	force ready
B7	0	1	1	0	1	enable after RETI
8B	0	0	0	1	0	reset status

Význam funkcí:

load - nuluje čítač slabik a ukládá startovací adresu pro oba porty

continue - nuluje jen čítač slabik; adresování pokračuje od právě nacházejícího se místa

enable interrupt - dovoluje výskyt přerušení

disable interrupt - zamezuje výskyt přerušení

reset interrupt - nuluje a neuvolňuje veškeré přerušovací obvody

enable DMA/disable DMA - uvolňuje/zamezuje uvolnění všech operací DMA s výjimkou přerušení

read byte follows - následující zápis do DMA obsahuje bitovou masku určující, který z registrů DMA má být čten

reset RD - následující čtení bude ze stavového registru

RD status - čtení statusu

force ready - ready je považováno za aktivní bez ohledu na stav úrovně vývodu RDY. Používá se při operacích mezi paměťmi, kdy signály RDY nejsou třeba

enable after RETI - DMA nevyžaduje sběrnici, pokud neobdržel signál RETI

reset status - nuluje stavové bity srovnávání (match) a konce bloku

reset - zamezuje přerušení a požadavku sběrnice, nuluje přerušovací obvody.

Řídicí slovo READ BYTE

D7	D6	D5	D4	D3	D2	D1	D0
-	BAU	BAL	AAU	AAL	BUC	BLC	ST

Osmý bit D7 není použit, jinak jedničková úroveň kteréhokoliv bitu D0 až D6 uvolňuje takto označený registr ke čtení mikroprocesorem.

D6 = BAU vyšší adresa portu B
D5 = BAL nižší adresa portu B
D4 = AAU vyšší adresa portu A
D3 = AAL nižší adresa portu A
D2 = BUC čítač slabik, horní byte
D1 = BLC čítač slabik, dolní byte
D0 = ST status

Řídicí slovo přerušení

D7	D6	D5	D4	D3	D2	D1	D0
-	IbR	SAI	IVF	PCF	PG	IMF	IEB

Osmý bit D7 není použit; jedničková úroveň kteréhokoliv bitu D0 až D7 volí vyznačenou funkci:

D6 = IbR přerušení před vyžádáním sběrnice
D5 = SAI status ovlivňuje přerušovací vektor
D4 = IVF následuje přerušovací vektor
D3 = PCF následuje povelové slovo PULSE COUNT
D2 = PG generování pulsu
D1 = IMF přerušení při shodě (match)
D0 = IEB přerušení po konci bloku

Řídicí slovo časování

D7	D6	D5	D4	D3	D2	D1	D0
WE	RE	-	-	ME	IE	T1	T0

Bity D4 a D5 nejsou v tomto řídícím slově používány. Kombinace T0 s T1 určuje délku cyklu, a sice následovně:

D1-D0 = T1-T0 = 0 0 čtyři takty
0 1 tři takty
1 0 dva takty
1 1 jeden takt

Úroveň logické nuly v bitech D2, D3, D6 nebo D7 způsobí, že odpovídající řídicí signál skončí o polovinu taktu před koncem cyklu.

Pozn.: celá operace - čtení či zápis v třídě přenosu nebo čtení v třídě vyhledávání - musí být alespoň 2 cykly dlouhá.

D2 = IE $\overline{\text{IORQ}}$ end (ukončení IORQ)
D3 = ME $\overline{\text{MREQ}}$ end (ukončení MREQ)
D6 = RE $\overline{\text{RD}}$ end
D7 = WE $\overline{\text{WR}}$ end

Řídicí slovo maskovacího registru (MASK BYTE)

Je to osmibitové slovo, kde nulová úroveň kteréhokoliv bitu D0 až D7 určuje srovnávání této bitové pozice řídicího slova registru se stejnou bitovou pozicí ve čtených datech.

D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	X	X	X	X

kde X = 0 nebo 1; při X = 0 zadáno srovnávání.

Řídicí slovo srovnávacího registru (MATCH BYTE)

Toto slovo vyplývá ze zadání maskovacích bitů výše popsané maskovací slabiky. Ve tvaru D0 až D7 způsobí srovnávání při operaci čtení.

Status - stavové slovo (STATUS BYTE)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	EOB	MT	IP	-	RA	WAV

Nejsou použity bity D2, D6 a D7.
D0 = WAV write address valid - platná adresa zápisu
D1 = RA ready active - RD aktivní
D3 = IP interrupt pending - očekávané přerušení
D4 = MT match - srovnávání
D5 = EOB end of block - konec bloku
D6 = D7 = D2 = X

Povelové slovo čítání pulsů (PULSE COUNT)

Je to osmibitové slovo, uložené do registru. Při dokončení každé operace je obsah registru srovnáván s nižší slabikou čítače slabik. Při srovnávání je na výstupu INT impuls (není však generováno přerušení).

Přerušovací vektor

Toto osmibitové slovo je vysláno datovou sběrnicí na mikroprocesor během kvitování přerušení z DMA, pakliže je v daném okamžiku zařízením s nejvyšší prioritou.

Jestliže je v řídicím slově přerušení bit D4 jedničkový - viz výše - a dále DMA je naprogramován na přerušení dle dané stavové podmínky (D5 = SAI = 1, viz tamtéž), pak bity D2 a D1 modifikují vektor následovně:

D2	D1	
0	0	INT on RDY
0	1	match
1	0	end of block
1	1	match, end of block

Následující příklad dle /37/ obr. č. 21 ukazuje, jak lze naprogramovat obvod DMA pro přenos dat z periférie (port A) do paměti (port B). Tabulka slabik může být uložena v paměti a přenesena do DMA výstupní instrukcí "OTIR".

port A	směr dat	port B
periférie adresa 05H	⟶	paměť startovací adresa 1050H
	délka bloku 1000H slabik	

RDY ze periférie je aktivní v jedničce,
adresa paměti je inkrementována s každým zápisem (WR).

Pozn.: přenosové funkce, tj. způsoby a třídy přenosu DMA, lze měnit podle potřeby během provozu.

II.5. Z80-COMBO

Z80-COMBO (MK 3886) je speciální podpůrný obvod série Z80, jenž svými vlastnostmi umožňuje realizovat mikroprocesorový systém s minimálním počtem součástí. Obsahuje totiž 256 bytů paměti RAM, dva časovače, sériový vstupně/výstupní port, přerušovací logiku atd. Sériový kanál umožňuje jak

synchronní, tak i asynchronní operace. Z celkového rozsahu paměti je možno jednak chránit cíleně 64 bytů proti přepsání, jednak těchto 64 bytů lze odděleně napájet ze separátního zdroje sníženým napětím (standby z chemického zdroje), a tak chránit data před zničením při eventuálním výpadku provozního napětí.

Struktura obvodu

Struktura tohoto kombinovaného obvodu (odtud i jeho název) je na obr. 26. Je patrno, že je v podstatě obdobný - z hlediska řízení - s dříve popsanými obvody. Nacházíme zde opět sběrnicové rozhraní, jež je navíc vybaveno latchy, přímo připojitelné k sběrnicím mikroprocesoru, logiku vnitřního řízení, logiku řízení přerušení, dvojici časovačů A a B, paměť RAM, sériový port a čtyři přerušovací externí kanály. Struktura byla výrobcem volena tak, aby potenciálnímu uživateli poskytla co nejvyšší flexibilitu v systémových sestavách.

Obvod může tedy vykonávat základní čtyři funkce. Jsou to: čtení a zápis do paměti, čítání a časování, sériový příjem a výdej informací, jakož i řízení přerušení. Těmito vlastnostmi je předurčen právě pro nasazení v minimálních, avšak výkonných řídicích systémech.

Obvod vyžaduje pouze jedno napájecí napětí +5 V. Avšak jak již bylo řečeno, pro ochranu části paměti lze použít další napájecí vstup V_{RR}; pokud tato ochrana není požadována, připojí se V_{RR} k napájecímu napětí +5 V.

Logika vnitřního řízení sekvenčně přijímá a dekóduje řídicí signály vyslané z CPJ, většinou se týkají paměťových operací či vstupně-výstupních operací; ani jedna z těchto operací nezávisí na systémovém taktu Ø, ale spíše na časových podmínkách a úrovních signálů na vstupech $\overline{CS}_M$, $\overline{CS}_{I-O}$, $\overline{RD}$, $\overline{WR}$, $\overline{MREQ}$ a $\overline{IORQ}$. Řídicí logika a v/v port jsou programátorovi přístupné prostřednictvím deseti vnitřních registrů, jejich nastavením je zajištěn správný provoz dat mezi CPJ a

a COMBO odpovídající dané konfiguraci. Systémový takt Ø je použit pro oba časovače. Kromě výstupů (aktivních v nule při dosažení nulového obsahu čítačů) $\overline{ZCA}$ a $\overline{ZCB}$ (zero count output) má časovač A ještě vstup externího přerušení INT O, jenž je zaveden přímo do jeho řídicího obvodu. Jím je možno volit jeden z dvou způsobů časování.

Sériový port dovoluje vstup a výstup sériových dat synchronních či asynnchronních. Port je v podstatě tvořen šestnáctibitovým registrem, do něhož může být zapisováno či z něj čteno. Přenosová rychlost je závislá na kmitočtu vnějších hodin (S. clock). Port může být využíván nejen pro sériovou komunikaci, ale může též sloužit jako interface sériových vnějších pamětí typu CCD.

Čtyři vstupní linky vnějšího přerušení dovolují uskutečnit prioritní, nezávisle vektorované, maskovatelné přerušení (spouštěné hranou). Tři z těchto vstupů jsou přivedeny přímo do řídicího obvodu externího přerušení, zatímco čtvrtý - INT O - spolu s časovačem A dovoluje realizovat čítačově programovatelné vnější přerušení. Všechny přerušovací vstupy jsou TTL kompatibilní a bufferovány Schmittovými obvody. Priorita je determinována dvojím způsobem; jednak je určena vnitřní priorita pro osm kanálů schopných generovat přerušení, jednak priorita kteréhokoliv komponentu zařízení je určena jeho fyzickým umístěním v prioritním řetězci, a to prostřednictvím vstupu IEI a výstupu IEO.

Obvod Z80-COMBO je umístěn ve čtyřicetivývodovém pouzdru DIL 15. Obr. 27 zachycuje jak označení jednotlivých vývodů pouzdra (vpravo), tak i jejich funkční rozdělení (vlevo). Význam jednotlivých vývodů je obdobný, jako u již dříve popsaných podpůrných obvodů. Za zmínku stojí pouze uvolňovací vstupy $\overline{CS}$, kdy $\overline{CS}_M$ slouží pouze pro paměť, zatímco $\overline{CS}_{I-O}$ pro celý čip. Obvod má samostatný vstup $\overline{RESET}$ sloužící počáteční iniciaci.

Příklad aplikace

Na obr. 28 je zapojení mikroprocesorového systému, tvořeného minimálním počtem prvků. Aktivní jsou celkem čtyři: oscilátor - generátor taktu 7404, Z80-CPU, Z80-COMBO a MK 34000 (ROM - 2 KB); k provozu je ovšem třeba ještě napáječ +5 V.

Systém používá nerozsáhlou paměť, a sice 2 KB programové paměti ROM a 256 bytů paměti dat RAM. Přepínání mezi oběma paměťmi obstarává adresová linka A11, přivedená - přes invertor - na uvolňovací vstupy. U ROM tady na CE3, u RAM na $\overline{CS}_M$. A protože citované vstupy jsou aktivní při vzájemně odchylných polaritách, může být přepínací signál zaveden na ně společně. K uvolňování čipu COMBO slouží adresový signál A4. Jinak je zapojení opravdu jednoduché a nevyžaduje dalšího komentáře. Předpokladem správné činnosti ovšem je naprogramování funkční činnosti COMBO; odpovídající programový úsek je obsažen samozřejmě v provozním programu pevné paměti ROM.

II.6. Oddělovací zesilovače sběrnic

Jednotlivé vývody mikroprocesoru - což se týká nejen Z80-CPU, ale většiny typů - jsou schopny proudově napájet jen jednu obvodovou zátěž TTL. Jinými slovy řečeno, k jednomu každému vývodu CPJ smí být připojen jen jeden vstup kombinačních či sekvenčních obvodů běžné řady 74/54.., vyrobených bipolární technologií. Ale i u nízkopříkonové série číslicových obvodů série 74/54LS.. je počet připojení omezen (do max. odebíraného proudu 1,8 mA), a je řádu jednotek.

Tato skutečnost musí být respektována, nemá-li dojít k proudovému, popřípadě i kapacitnímu přetížení mikroprocesoru, a tím i jeho případné destrukci. U malých mikroprocesorových systémů a školních mikropočítačů, jejichž obvodová konfigurace je nerozsáhlá (a pohybuje se kolem deseti obvodů TTL), vystačí se bez oddělovacích zesilovačů sběrnic. Tyto odělovací zesilovače svými vlastnostmi zajišťují jednak za-

nedbatelné zatížení výstupů mikroprocesoru na své vstupní straně - cca 0,25 mA, jednak výkonové napájení sběrnice na své výstupní straně - cca 50 mA. Naproti tomu u rozsáhlých systémů, nebo u systémů určených pro pozdější rozšíření, se bez oddělovacích zesilovačů (buffer) neobejdeme. Ty totiž mimo již zmíněné dostačující proudové napájení sběrnice odstraňují (tj. oddělují) i vliv kapacitní zátěže rozsáhlejších pamětí RAM, složených z většího počtu integrovaných obvodů.

Následující obrázek č. 29 zachycuje zapojení CPJ "obklopené" moderními osmibitovými oddělovacími zesilovači. V zapojení je použito dvou typů, a sice jednak typu 74LS244, jednak 74LS245. První z uvedených je osmibitový neinvertující zesilovač, který ve dvou kusech odděluje adresovou sběrnici (IO1 a IO2), třetím pak (IO4) část sběrnice řídicí. Druhý typ je taktéž osmibitový, a odděluje datovou sběrnici. Avšak protože provoz po datové sběrnici se děje jak z mikroprocesoru, tak i do něj, je tento oddělovací zesilovač obousměrný (IO3), kde směr jeho propustnosti je řízen úrovní signálu na jeho vstupu DIR. V normálním stavu je propustný směrem z mikroprocesoru na systémovou datovou sběrnici - tedy z A do B, viz obr. 32. Jenom při aktivním signálu $\overline{RD}$ nebo $\overline{M1}$, jež jsou propojeny přes člen AND (1/4 74LS08), je směr jeho propustnosti přepnut. (Člen AND vzhledem k negativní logice působí zde jako člen OR.) Pro přepínání musí být použit i signál $\overline{M1}$ - a ne jen $\overline{RD}$, neboť v cyklu kvitování přerušení, kdy má být procesorem čteno, je aktivní pouze $\overline{M1}$, a nikoliv $\overline{RD}$.

Uvolňovací vstupy G obvodů 74LS244 (vývody č. 1 a 19) jsou uzemněny, takže adresové vývody mikroprocesoru jsou trvale propojeny s adresovou sběrnicí; totéž se týká i obvodu IO4. Pokud by byl žádán přímý přístup do paměti (DMA), bylo by třeba zmíněné uvolňovací vývody obvodů IO1 až IO3 propojit - ve správné polaritě se zesíleným signálem $\overline{BUSAK}$. Tato alternativa je zakreslena na následujícím obr. 30. Na tomto

místě třeba ještě připomenout, že uvedené oddělovací zesilovače jsou třístavové; to znamená, že jejich vývody se nacházejí ve vysokoimpedančním stavu, jestliže jejich uvolňovací vstupy jsou neaktivní.

Zapojení na obr. 30 zachycuje centrální procesorovou jednotku (tj. Z80A-CPU) včetně generátoru hodinových impulsů IO1, děličky dvěma nebo čtyřmi IO2 (umožňující použití krystalu o vyšším než zadaném kmitočtu), monostabilního klopného obvodu IO3 (zajišťujícího generování nulovacího impulsu o definované délce), čtveřice invertorů IO4 a konečně oddělovací zesilovače B1 až B4.

Řídicí vstupy $\overline{\text{WAIT}}$, $\overline{\text{BUSRQ}}$, $\overline{\text{NMI}}$ a $\overline{\text{INT}}$ - obdobně jako v předcházejícím příkladu - jsou přes rezistory R5 až R8 - připojeny na +5 V. Mají tedy jedničkový signál a jsou neaktivní, mohou však být zvenčí aktivovány. Oddělovací zesilovače - z důvodu jednotnosti - jsou stejného typu: obousměrné, osmibitové, neinvertující. Proto u B2, B3 a B4 je trvale zvolen směr přenosu připojením vstupu DIR na jedničkový signál (+5 V). U B1 je směr přepínatelný, tentokráte jinou kombinací hradel (1/2 7400). $\overline{\text{BUSAK}}$ - kvitovací signál žádosti o sběrnici pro přímý přístup do paměti - je přes invertor I7 přiváděn na uvolňovací vstupy všech oddělovačů B1 až B4, tj. na jejich vývody č. 19. J-li tedy $\overline{\text{BUSAK}}$ aktivní (= 0), pak na vstupech G oddělovačů je signál log "1", který způsobí "odpojení" od sběrnic a jejich postoupení jinému uživateli.

Typy oddělovacích zesilovačů

Oddělovacích zesilovačů v jednosměrném či obousměrném provedení je celá řada. Liší se též i počtem bitových vedení (čtyř, šesti či osmibitové), proudovou zatížitelností a způsobem přenosu; bylo by nad rámec této práce všechny popisovat. Uvedeme proto jen ty nejčastěji se vyskytující z široké zahraniční produkce, dále pak i naší tuzemské provenience.

Mezi osvědčené čtyřbitové budící zesilovače patří:

8T26 - 4 invertující budiče sběrnicových vedení/přijímače - 48 mA

8T28 - 4 neinvert. budiče/přijímače, s výst. proudem á 48 mA, 17 ns

8T34 - 4 neinvert. budiče/přijímače, s výst. proudem á 50 mA, 25 ns

8T38 - 4 neinvert. budiče/přijímače, s výst. proudem á 50 mA, 25 ns

74125/74LS125 - 4 neinvertující budiče, IO_L = 16 mA, 10/16,5 ns, třístavové

74126/74LS126 - 4 neinvertující budiče, IO_L = 16 mA, 10/16,5 ns

Šestibitové budící zesilovače jsou representovány:

8T95 74LS365 74365	- 6 výkonových budičů se společným uvolněním	IO_L = 48 mA, 12 ns 16 mA, 19 ns 32 mA, 19 ns
8T96 74LS366 74366	- 6 invertujících výkon. budičů se společným uvolněním	IO_L = 48 mA, 10 ns 16 mA, 16,5 ns 32 mA, 19 ns
8T97 74LS367 74367	- 6 výkonových budičů s dvěma uvolňovacími vstupy	IO_L = 48 mA, 12 ns 16 mA, 19 ns 32 mA, 19 ns
8T98 74LS368 74368	- 6 invertujících výkon. budičů s dvěma uvolňovacími vstupy	IO_L = 48 mA, 10 ns 16 mA, 16,5 ns 32 mA, 19 ns

Nejvíce rozšířená je řada čtyř- a osmibitových budičů/zesilovačů, série 24X, uvedená na trh renovovanou firmou Texas Instruments:

74LS240 - 8 třístavových budičů, invertujících, s dvojím uvolněním, IO_L = 24 mA, 9 ns

74LS241 - 8 třístavových budičů, neinvertujících, s dvojím uvolněním opačné aktivity, IO_L = 24 mA, 12 ns

74LS242 - 4 obousměrné třístavové budiče/zesilovače, invertující, s dvojím uvolněním opačné aktivity, IO_L = 24 mA, 12 ns

74LS243 - 4 obousměrné třístavové neinvertující budiče/zesilovače, s dvojím uvolněním opačné aktivity, IO_L = 24 mA, 12 ns

74LS244 - 8 jednosměrných třístavových budičů/zesilovačů, neinvertujících, s dvojím uvolněním stejné aktivity, IO_L = 24 mA, 12 ns

74LS245 - 8 obousměrných třístavových neinvertujících budičů/zesilovačů, se společným uvolněním, IO_L = = 24 mA, 12 ns

Uvedené budiče pod stejným označením vyrábějí dnes i další výrobci. Analogon budiče 74LS245, pod označením MD74HCT245 vyrábí fa Marconi v technologii C-MOS pro provozní napětí v rozpětí 2 až 7 V. Důležité je, že ačkoliv zatížitelnost každého výstupu je citovaných 24 mA, při respektování závislosti UO_L/IO_L lze odebírat až 80 mA. Vnitřní zapojení těchto budičů/oddělovacích zesilovačů spolu s označením jejich jednotlivých vývodů je na obr. 32.

Z tuzemské produkce patří mezi budiče sběrnic následující typy:

MH3216 - 4 neinvertující třístavové budiče, obousměrné, IO_L = 50 mA, tp = 25 ns

MH3226 - 4 invertující třístavové budiče, obousměrné, IO_L = = 50 mA, tp = 25 ns

Pozn.: oba typy mají oddělené vstupy DIi a výstupy DOi jedné strany - viz schéma na obr. 33, které mohou být pro obousměrný provoz vzájemně propojeny. Připojují se k výstupu mikroprocesoru, zatímco vstupy/výstupy DBi proudově zásobují sběrnici vedoucí k paměťovým aj. blokům. Směr propustnosti je dán úrovní na vstupu DIEN, a sice pro DIEN = 0 platí DI = DB, pro DIEN = = 1 pak DB = DO.

MH3212 - 8 bitový vstupně výstupní obvod (analogon I 8212) Tento obvod obsahuje osmibitový třístavový latch s budičem, logiku pro výběr obvodu a řízení, dále pak klopný obvod pro generování a řízení přerušení. Protože tento obvod je určen pro různá nasazení (budič-oddělovací zesilovač, paměťový latch či multiplexer), věnujeme mu v dalším více pozornosti, obr. 34.

Datový latch se skládá z osmi klopných obvodů D. Výstupní signály Q kopírují vstupní D při jedničkové úrovni hodinového signálu. Uložení do latche následuje při úrovni hodinového signálu log "0". Latch se nuluje signálem $\overline{CLR}$. Hodinový signál C je však nadřazen $\overline{CLR}$. Výstupy datového latche jsou připojeny na neinvertující třístavové budiče, mající společné uvolňování signálem "en". Obvod 3212 má dále čtyři řídicí vstupy $\overline{DS1}$, DS2, MD a STB. Prvních dvou se používá pro výběr čipu (při $\overline{DS1}$ = 0 a DS2 = 1 je obvod uvolněn), kdy výstupní budiče jsou uvolněny a klopný obvod přerušení nastaven na log "1". Vstupní signál MD řídí stav budičů a určuje zdroj vstupního signálu C pro datový latch. Je-li MD = 1 (výstupní režim), jsou budiče uvolněny a zdroj hodinového signálu C pro datový latch je odvozen z logiky pro výběr obvodu. Vstupní signál STB slouží při režimu "vstup" (MD = 0) jako hodinový signál pro datový latch a pro synchronní resetování klopného obvodu SR pro indikaci požadavku přerušení. (Klopný obvod je spouštěn sestupnou hranou.)

Klopný obvod je nastaven signálem $\overline{CLR}$, aktivním v nule. Je-li klopný obvod nastaven, je ve stavu, při němž nenastává přerušení. Výstup Q klopného obvodu S-R je připojen na invertující vstup hradla NOR, jehož druhý neinvertující vstup je připojen na logiku pro výběr obvodu. Výstup $\overline{INT}$, aktivní v nule, se připojuje buď na prioritní obvod (8214), nebo na stejnojmenný vstup mikroprocesoru Z80. V lit. /36/, /54/ a /18/ nalezne zájemce příklady zapojení tohoto užitečného obvodu (z "rodiny" podpůrných obvodů µP 8080) jako jedno či obousměrného budiče sběrnice, obvodu pro přerušení, stavové-

ho latche či vstupního nebo výstupního obvodu. Pro většinu z těchto aplikací je důležité, že obvod nezatěžuje datové či adresové linky (vstupní proud jen 250 μA), zatímco jeho každý výstup může budit až deset zátěží TTL (IO_L = 15 mA). Na rozdíl od výše popsaných budičů/zesilovačů však zaujímá fyzicky více místa, neboť je umístěn v pouzdře DIL s 24 vývody, s roztečí 15 mm obou řad vývodů.

Jmenujme ještě dva perspektivní oddělovací zesilovače/budiče sběrnic naší provenience. Jsou to:

MHB 8286 - osminásobný třístavový obousměrný budič/zesilovač (obdoba typu 74LS245, avšak s nekompatibilními vývody) a

MHB 8287 - osminásobný invertující obousměrný třístavový budič/zesilovač, taktéž s jedním uvolňovacím vstupem jako u 8286, a dále

MHB 8282 - osminásobný jednosměrný třístavový budič/zesilovač, neinvertující, a jeho invertující protějšek MHB 8283. Oba posledně jmenované typy tvoří tedy analogony třístavových jednosměrných budičů LS240 a LS244, s nimiž ovšem nejsou opět vývodově kompatibilní. Navíc - vzhledem k jednomu uvolňovacímu vstupu $\overline{CS}$ a vnitřní struktuře - umožňují latchování sběrnice při nulové úrovni strobovacího signálu "strobe". Při jedničkové úrovni pak jsou propustné směrem z "A" do "B" - viz označení jejich vývodů na obr. 35.

Závěrem této stati připomeňme, že v sérii C-MOS se připravuje výroba třístavového budiče sběrnice, a sice typu MHB 4503.

III. Koncepce skladby mikropočítače

Přestože existují jednočipové mikropočítače (u nichž jedno pouzdro integrovaného obvodu široké integrace skrývá monolitický provozuschopný systém - např. i u nás vyráběný typ 8748), tak jedno nebo vícedeskové provedení "klasického" mikropočítače skládá se pochopitelně z více částí. Tu hlavní

tvoří samozřejmě mikroprocesor - v našem případě tedy Z80--CPU, dále paměti (programová a datová), generátor taktu, obvody rozhraní - styku s vnějším okolím a posléze i pomocné obvody, zprostředkující např. výběr čipu při řízení apod., viz blokové schéma mikropočítače na obr. 36.

Komplexní skladba mikropočítače vyžaduje použití generátoru taktu, který svými impulsy souvislého sledu řídí nástup a provádění jednotlivých operací. U některých mikroprocesorů bývá generátor taktu zaintegrován - u mikroprocesoru Z80-CPU se pro funkci generátoru taktu používá samostatného oscilátoru, složeného z několika běžných hradel, což bude ještě v pozdější části detailně ukázáno. Takt je jedno či vícefázový, s překrýváním nebo bez překrývání - u Z80-CPU je takt jednofázový. Časování (= taktování) musí být zcela přesné, jestliže mikropočítač má spolupracovat s okolím, např. v reálném čase, pomocí dálnopisu s předepsanou přenosovou rychlostí apod. Pak je generátor taktu stabilizován křemenným krystalem. Ve zvláštních případech může být takt (= sled impulsů) přiváděn i z vnějšku. Kmitočet taktu, nebo jak se též někde uvádí, kmitočet hodinových impulsů, zpravidla řádu MHz, není absolutním měřítkem pracovní rychlosti mikroprocesoru. Existují totiž systémy, jež při pomalejším taktu vybaví instrukci rychleji než jiné systémy s vyšším kmitočtem - to proto, že k provedení instrukce vystačí třeba s menším počtem taktů atd.

Kromě zmíněných bloků dotváří mikropočítač trojice sběrnic, které umožňují veškerou nutnou komunikaci příkazů a informací. Jsou to sběrnice adresová, mající zpravidla 16 vedení, datová (u osmibitových mikroprocesorů má osm vedení) a řídicí, jejíž počet vedení se různí podle typu mikroprocesoru a navrženého zapojení. Zpravidla se počet vedení řídicí sběrnice pohybuje mezi pěti až osmi.

Permanentní paměti

Paměť ROM, neboli tzv. pevná (permanentní) paměť, je naprogramována již přímo u výrobce, a to pomocí masky. Jejím obsahem je zpravidla obslužný provozní program, jímž se řídí cíleně veškerá činnost mikropočítače dané konfigurace. Paměť ROM však může být - ve svém rozšíření - i nositelem dalších účelových programů, např. assembleru, BASICu, různých her apod. Proto se též tento druh paměti nazývá pamětí programů.

Ve vývojových fázích či malých sériích je paměť ROM (read only memory = paměť jen pro čtení) nahrazena pamětí EPROM, taktéž permanentní pamětí, avšak programovatelnou uživatelem, a to pomocí vhodného elektronického programátoru. Programuje se buď nevratným propalováním (paměti typu PROM), nebo rozmístěním elektrického náboje (paměti typu EPROM = elektricky programovatelné ROM). Zatímco paměti ROM a PROM po naprogramování již nikdy nemohou změnit svůj obsah, je u pamětí EPROM situace odchylná. U tohoto posledně jmenovaného druhu pevné paměti je možno několikaminutovým osvětlením ultrafialovými paprsky (popř. roentgenovými) - zpravidla kolem 20ti až 30ti minut - vymazat již nevyhovující obsah a nahradit jej buď pozměněným, či zcela novým programem. K nejmodernějším druhům pevných pamětí patří tzv. EAROM (eletrically alterable ROM = elektricky pozměnitelná ROM), např. typ 2816, u níž - bez ozařování - lze opravovat či měnit její obsah i po jedné slabice.

Na obr. 37 je vyznačen tvar pouzdra a označení vývodů nejvíce používaných typů pamětí EPROM (2716, 2732, 2764) spolu s perspektivními (27128, 27256) - o vždy větší kapacitě paměťových buněk. (2464 odpovídá kapacitě 64 Kbitů, 27256 pak kapacitě 256 Kbitů, atd.) Na témže obr. 37 je uvedeno též označení vývodů některých kompatibilních pamětí RAM, a to typů 4118 - s kapacitou 1 KB, 6116 - s kapacitou 2 KB, 4864 a 5564 s kapacitou á 8 KB = 64 Kbitů. Rovněž zde nalezneme i starší typ paměti EPROM 2708, vyžadující ještě tři napájecí napětí (všechny dříve uvedené mají již jen jedno

napájecí napětí +5 V), jež zde uvádíme z hlediska úplnosti. Dále pak čtenář nalezne odchylné zapojení dvoukilobyteových pamětí 2716 fy Texas Instruments od jiných výrobců stejnokapacitní paměti EPROM, což se vztahuje i na čtyřkilobyteovou paměť TMS 2732 téhož výrobce.

Operační paměti (RAM)

Operační (pracovní) paměť je polovodičová paměť typu RAM (random acces memory - paměť s nahodilým přístupem), označovaná též RWM (read write memory - paměť pro čtení a zápis), schopná zápisu a nedestruktivního čtení v kterémkoliv okamžiku. Důležitým kritériem je doba přístupu k datům, která má u pomalejších typů vliv na pracovní rychlost celého mikropočítačového systému. Většinou se používají statické paměti RAM (mnohdy označované jako SRAM), uchovávající vloženou informaci tak dlouho, pokud je přiloženo provozní napětí - mnohdy i při snížené úrovni napájecího napětí ve vyčkávacím stavu, tzv. "stand by" - při současně sníženém odběru proudu. Používají se však též dynamické paměti, u nichž je informace uložena ve formě napěťových nábojů. Jsou někdy označovány jako DRAM; jejich výhodou je značná kapacita, nevýhodou naproti tomu nutnost periodického obnovování dat, a to v intervalech do dvou ms - tzv. refreshingem - osvěžováním. To proto, že vložený náboj vlivem svodu relativně jednoduché paměťové buňky se v čase umenšuje. Tuto skutečnost však mikroprocesor Z80-CPU respektuje, o čemž již byla zmínka v kapitole I.

Za zmínku stojí paměti RAM vyrobené technologií C-MOS, vyznačující se minimálním proudovým odběrem a širokým rozsahem napájecího napětí (+3 až +15 V). Označovány jsou někdy jako CRAM a setkáváme se s nimi nejen u přenosných mikropočítačů (kde i mikroprocesor je provedení C-MOS, viz analogon Z80, a sice typ NSC 800 /31/, či 80C85 - viz /72/ apod.), ale i u některých školních, kde pracují ve funkci simulátoru paměti ROM.

Typů pamětí RAM je nepřeberná řada; uveďme si zde jen ty nejvíce používané, popř. perspektivní typy. Jsou to:

- statické	2102	s organizací	1 Kbit x 1
	2114	s organizací	1 Kbit x 4
	4118	s organizací	1 Kbit x 8 (1 KB)
	6116	s organizací	2 Kbity x 8 (2 KB-C-MOS)
	6132	s organizací	4 Kbity x 8 (4 KB-C-MOS)
	6164	s organizací	8 Kbitů x 8 (KB-C-MOS)
- dynamické	4116	s organizací	16 Kbitů x 1
	4164	s organizací	64 Kbitů x 1

Tvary pouzder jakož i označení vývodů jsou na obr. 38, některé typy RAM z citovaných jsou zachyceny v předcházejícím obr. 37, a to pro svoji kompatibilitu rozměrů a vývodů s příslušnými typy ROM (4118 = 2758, 6116 = 2716, 6164 = = 5564 = 2764).

Paměti RAM mívají kapacitu (v základním vybavení mikropočítače) obvykle značně menší, než umožňuje adresová sběrnice, tak např. u jednodeskových školních mikropočítačů to bývá často jen 1 KB. Naproti tomu u moderních osobních mikropočítačů není překvapením operační paměť 64 KB, popřípadě 128 či 256 KB ("Husky", Anglie), apod. Pokud se týká paměti ROM, pak její velikost odpovídá danému monitoru, popř. dalším předváděcím programům; zpravidla se její kapacita pohybuje kolem 4 KB, u výkonnějších systémů s jazykem BASIC či FORT dokonce 12 KB, či více.

Interface mikropočítače

Pod pojmem interface rozumíme obvody pro styk mikropočítače s vnějším okolím. Nejsou to jen speciální stykové programovatelné obvody a oddělovací zesilovače (z nichž některé jsme již poznali v části II.) ale i výkonové členy galvanicky oddělující optoelektronické vazební členy - používané zejména při průmyslovém nasazení mikropočítače /75/ - přenosové členy (V.24, RS-232C), a v neposlední řadě i účelo-

vě sestavená styková zapojení, sestávající z více aktivních i pasívních členů. Patří sem tedy i převodníky (A-D, D-A, aj.) spolu s multiplexory přepínajícími čidla více snímaných míst či propojujícími ovládací členy (sekvenčně) na výstupní straně.

Jako interface (= stykový obvod, rozhraní) označujeme i taková obvodová zapojení, umístěná fyzicky zpravidla na samostatné desce a napájená většinou z řízené periférie, která umožňují bezproblematické připojení mikropočítače s potřebným vnějším zařízením, např. tiskárnou, souřadnicovým zapisovačem (plottrem) apod.; pod připojením se rozumí i možnost komunikace tím či onim nebo oběma směry, spolu s předáváním řídicích/synchronisačních signálů, apod. V části V.5. až V.6. se náležitostem některých důležitých rozhraní věnujeme podrobněji.

Vnitřní řízení

Z blokového schématu na obr. 36 jasně vyplývá, že mikroprocesor v čase komunikuje pomocí sběrnic s různými bloky mikropočítače. A protože jak programová paměť, tak i datová - případně i interface - jsou připojeny k společné adresové a datové sběrnici, kde první adresuje paměťové místo a po druhé se provádí přenos dat, nesmí dojít ke kolisi při přenosu dat. To se zajišťuje řízením výběru, tj. uvolňováním žádaných paměťových čipů /26/, /29/.

V tom nejjednodušším případě, kdy µP systém je třeba vybaven pouze pamětí ROM o kapacitě 1 KB a pamětí RAM, taktéž o kapacitě 1 KB - viz obr. 39 - poslouží jako řídicí vedení adresová linka A_{10}, která je přivedena na uvolňovací vstupy čipů CE obou druhů použitých pamětí. Aby však paměti byly uvolňovány vystřídaně, je daný řídicí signál (A_{10}) přiveden na uvolňovací vstup CE paměti RAM přes invertor. Z toho dále vyplývá, že paměť ROM je uvolněna při adresách 000H až 3FFH (binárně tedy 000 0000 0000 až 011 1111 1111) a paměť RAM při adresách 400H až 7FFH (binárně: 100 0000 0000 až 111

IIII IIII). Při výběru paměti RAM je jí třeba ještě přepnout na čtecí či zápisový provoz, což obstarává řídicí signál R/$\overline{W}$.

Pokud se ovšem použije více paměťových bloků, pak se pro řízení výběru používá zapojení s vhodným dekodérem, jenž uvolnění té či oné části paměti obstarává automaticky, a to částí adresové sběrnice - viz typický příklad na obr. 40. Všechny paměti jsou adresovány bity A_0 až A_9, bity A_{10} až A_{13} je adresován dekodér, který pro 16 možných vstupních stavů (0000 až IIII) generuje na svých výstupech 16 signálů, z nichž však vždy pouze jeden má aktivizující úroveň log "0" pro uvolňovací vstup CE. S měnící se částí adresy A_{10} až A_{13} se mění i poloha výstupního signálu CS_i, který tak postupně uvolňuje jednotlivé paměťové bloky. V daném případě je možno adresovat paměť o rozsahu 16 KB (A_0 až $A_{13} = 2^{14} = 16\,384$), a to v blocích o kapacitě po jednom kilobytu, přičemž výběr se týká též obvodu v/v styku (uvolňovanému signálem CS_{15}).

Nejčastěji používaným dekodérem je obvod 8205 (= 74LS138), patřícím mezi podpůrné obvody série mikroprocesoru 8080; v praxi se však setkáváme i s celou řadou vhodných kombinací členů NAND apod. Podstatné při takovýchto kombinacích je, zda je požadováno tzv. úplné dekódování či ne. (Úplným dekódováním se rozumí jednoznačný výběr v zadané adresní oblasti; příklady z obr. 39 a 40 zachycují neúplné dekódování, neboť nejsou respektovány stavy vyšších adresových vedení nad řídicími, tedy v prvním případě A_{11} až A_{15}, v druhém A_{14} až A_{15}.)

V praxi často vzniká požadavek na rozšíření stávající paměti o další kapacitu pomocí přídavné desky nebo desek s vhodnými paměťovými obvody. U takovýchto přídavných paměťových desek (obvykle stejného typu) nesmí být adresový rozsah daného paměťového bloku pevně nastavený, ale volitelný. To proto, aby jednotlivé bloky na sebe navazovaly.

Zapojení na obr. 41 se týká přídavné paměti s celkovou kapacitou 4 KB; deska je osazena staršími typy paměťových obvodů - 2112, jichž pro danou kapacitu bylo použito 32 kusy.

Pro přehlednost je na obr. však naznačena jen první a poslední dvojice. Uvolňovací vstupy CE jsou zde spínány opět dekodérem 74154, jenž je adresován bity A_8 až A_{11} adresové sběrnice. Vedení nižších bitů A_0 až A_7 adresují již jen paměťové obvody. Uvolňovací vstup G2 dekodéru je řízen součinem nejvyšších bitů A_{12} až A_{15}. A protože pro předvolbu je nutné mít tyto bity k dispozici jak v přímé, tak i v inverzní formě, zajišťují inverzi členy H1 až H4. Libovolné zvolení adresového rozsahu po 4 KB umožňují přepínače S1 až S4, jimiž se volí taková kombinace bitů, která odpovídá požadované oblasti, při níž na všech čtyřech vstupech součinového hradla 7420 jsou signály s úrovní log "1". Tak např. pro rozsah 2000H až FFFH musí být vstupy 5,4 a 2 hradla 7420 spojeny s výstupy invertorů, zatímco vstup 3 je připojen na A_{13} přímo, což odpovídá binárnímu tvaru 0010 (pro A_{15} až A_{12}). Změnou poloh přepínačů lze tedy paměťovou desku adresově umístit podle potřeby v možném rozsahu 64 KB po 4 KB tak, aby nedošlo ke kolizi překrýváním.

Přepínače se dají nahradit spínači, či jen propojkami, použije-li se místo invertorů hradel typu "výlučné nebo" - XOR. Zapojení jednoho takového hradla je na témž vyobrazení; na jeho výstupu se objeví signál s úrovní log "0" vždy jen při shodnosti úrovní obou vstupních signálů. Z toho plyne, že při sepnutém spínači S_i tomu tak bude při úrovni $A_i = 0$, při rozpojeném spínači S_i pak při úrovni $A_i = 1$. Pro tyto účely se zpravidla používá čtyřnásobné hradlo XOR typu MH 7486, 74LS266 či 74LS136.

K zapojení zbývá již jen podotknout, že v tomto případě je dekodování úplné - jednoznačné.

Na obr. 42 přinášíme zapojení jednoduché centrální procesorové jednotky (CPJ) s programovou pamětí 2 nebo 4 KB a s pamětí dat 1 KB. Všimněme si dále, že jako generátor taktu slouží dvojice Schmittových obvodů (1/2 7414), u níž se požadovaný takt (cca 2 MHz) nastavuje potenciometrovým trimrem R = 1kΩ ; není tedy stabilizován křemenným krystalem.

Řízení výběru zde obstarávají obvody IO_3 a IO_4 spolu s řídicími signály $\overline{MREQ}$ a A_{12}. Protože po resetu začíná mikroprocesor Z80 (stejně jako 8080) pracovat na adrese 0000H, musí od této adresy začínat provozní monitorovací program, nacházející se v pevné paměti. Jinými slovy řečeno, paměť ROM-IO_7 musí být adresovatelná v rozsahu 0000H až 0FFFH. To zajišťuje jedno hradlo "nebo" (1/4 7432), na jehož vstup 1 je přivedena adresa A_{12}, na vstup 2 pak signál $\overline{MREQ}$. Výstup 3 hradla pak má aktivizační signál log "0" jen tehdy, když oba zmíněné vstupy nesou též nulovou úroveň ve stejném okamžiku. Druhým hradlem "nebo" jsou uvolňovány čipy paměťových obvodů RAM - IO_5 a IO_6. Avšak protože signál z A_{12} je na jeho vstup přiveden přes invertor IO_3, je paměť RAM přístupna jen tehdy, když současně existuje požadavek po paměti ($\overline{MREQ}$ = 0) a A_{12} = = I. To odpovídá adresovému rozsahu 1000H až 13FFH. Všimněme si ještě, že čtení či zápis v daném případě je řízen pouze signálem $\overline{WR}$, přiváděným na odpovídající vstupy $\overline{W}$ paměti RAM.

Ovládání a indikace

Je nesporné, že každý mikropočítač musí mít jak ovládací prvky (zpravidla tlačítka), tak i indikační. Obě skupiny, dotvářející mikropočítač či mikroprocesorový systém, se mohou podle účelu nasazení dost podstatně lišit. Pomineme-li průmyslové a spotřební aplikace, pak klasický mikropočítač je ovládán z šestnáctkové nebo alfanumerické klávesnice; indikace chodu a programových výsledků pak je znázorňována na světloemitujících diodách, číslicových displejích, popř. obrazovkových monitorech.

Pokud není klávesnice a displej součástí mikropočítače - viz část V - kdy převažuje softwareové řešení těchto prvků nad hardwareovým (protože vyžaduje méně součástí), tvoří samostatné periférie standardně připojitelné. Popis alfanumerických klávesnic a obrazovkových monitorů nespadá tedy do rámce této práce. Naproti tomu je třeba se zmínit několika slovy o příslušném interface.

K připojení běžné paralelní klávesnice slouží zapojení uvedené na obr. 43 , které pochopitelně pracuje v součinnosti s provozním monitorem /2/ a zapojením CPJ z obr. 42. Data z klávesnice - jako bity b_0 až b_6 (b_7 nepoužit) - jsou přivedena vždy na jeden vstup hradel XOR (7486), jejichž druhé vstupy jsou pomocí spojky JV uzemněny - nebo ponechány volné. Při uzemnění nepůsobí na polaritu procházejícího signálu; leží-li však na společných vstupech hradel úroveň log "1", pak pracují hradla jako invertory. Tak je možné použít i klávesnic majících výstup v záporné logice. Vstupní port tvoří latch B2 - 74 374; na jeho vstupu musí být hodnota ASCII znaku jen v pozitivní logice. Strobovací puls z klávesnice, jenž indikuje vyslání znaku, je rovněž přiveden na dvouvstupové hradlo XOR, kde pomocí spojky JS lze upravit jeho polaritu. Přes hradlo O2 - řídicím signálem RD, IOR a adresou shodnou s nastavenou (A_1 až A_7 = B_1 až B_7) - je obsah latche B2 přenášen na datovou sběrnici mikroprocesoru.

V zapojení se setkáváme ještě s třístavovým oddělovacím obousměrným zesilovačem sběrnice B_1. Ten zde slouží k informování CPJ - osmibitovým slovem, nastaveným spínači S_0 až S_7 - v jakém požadovaném provozním způsobu a jakou přenosovou rychlostí (týkající se však sériového styku) se bude komunikovat. Pozn.: nastaveným slovem se danému monitoru /2/ vlastně říká, s jakou (volitelnou) konfigurací má pracovat.

Na obr. 44 je zaznamenán časový průběh signálů při zadávání dat z klávesnice. Bitem b_7, respektive jeho úrovní je indikováno periodicky dotazujícímu se mikroprocesoru, zda z klávesnice jsou vyslána data. Je-li tedy b_7 = 0, lze nová data teprve očekávat. Následuje-li pak strobovací signál (z klávesnice) - oznamující vyslání dat - je pak za invertorem I4 vložen jedničkový signál do klopného obvodu, čímž se na jeho inverzním výstupu N objeví úroveň log "0". Tato úroveň, jakož i vstupní data, je převzata do B2. Klesající hranou čtecího pulsu je klopný obvod opět resetován; střídající úrovní bitu b_7, ovšem pouze při součinnosti strobovacího pul-

su z klávesnice, jsou indikována platná data k převzetí. Členy V1 a V2 slouží pouze k adresování desky spolu se spínači B1 a B7.

Komunikace s mikropočítačem, jakož i vydávání abecedněčíslicových výsledků, se nejlépe sleduje na víceřádkovém obrazovkovém monitoru - displeji. Lze ovšem k tomu účelu používat i neupravovaný TV přijímač; v obou případech musí být μPC vybaven tzv. videointerfacem, v druhém navíc ještě vf modulátorem, umožňujícím přivádět modulovaný informační vf signál přímo do anténních svorek TV přijímače.

Jeden takový videointerface - s populárním zahraničním řadičem fy Thomson - typem 9364A umožňujícím nejen sériovou komunikaci, ale i přímé připojení klávesnice, popsal autor v /28/. I když většina video-řadičů zahraniční provenience je určena pro připojení k sériově pracujícímu stykovému obvodu (viz rozhraní RS 232C a Z8-SIO), lze je - při vhodném zapojení - použít i v paralelním video-interface, přímo připojitelném k datové sběrnici mikropočítače. V již citované lit. /2/ jsme našli obdobné zapojení - viz obr. 45, které je dále podrobněji popsáno; představuje typický příklad použití řadiče CRT = jednoúčelového procesoru pro generování exaktně časovaných sledů pomocných a synchronisačních pulsů, jakož i pro přenos informace k video-účelům.

Jádrem celého stykového zapojení, umístěného na jedné desce cca polovičního evropského formátu, je již citovaný řadič EF 9364A, který přebírá kompletní řízení generování znaků a jejich rozmísťování na obrazovku v šestnácti řádcích po 64 znacích včetně řízení kursoru (ukazovátka) a výmazu znaku, řádku či celého stínítka. Mimoto sestává komentovaný interface z obrazové paměti IO_1 a IO_2 (2x 2114) o kapacitě 1 KB, (což odpovídá právě max. zobrazovatelnému počtu znaků ve formátu 16 x 64), generátoru znaků IO_3, tří latchů IO_4, IO_5 a IO_6 (3x 74374), čtyřbitového čítače IO_7, posuvného registru IO_8, generátoru bodů IO_9, dekodéru IO_{10} a nezbytných pomocných obvodů IO_{11} až IO_{15}.

Data z obrazové paměti jsou přiváděna přes mezipaměť na generátor znaků, který je zde s výhodou představován dvoukilobyteovou pamětí EPROM, naprogramovanou jak pro vytváření malých a velkých písmen včetně číslic a diakritických znamének, ale i pro generování semigrafických znaků. Znaková matice má rastr 8 x 8 bodů, pro číslice a písmena se však z ní využívá pouze 5 x 7 bodů, přičemž nevyužité bodové "okolí" slouží k oddělení řádek a jednotlivých znaků mezi sebou. (Je logické, že pro vytvoření semigrafických znaků je využíván celý rastr 8 x 8 bodů; tak je možno seskupením vhodných znaků vytvářet grafické útvary či figury.) Data vycházející z IO_3 jsou registrem IO_8 transformována do sériového sledu a tvoří již vlastní videosignál (zatím bez synchronizační směsi).

Řadič IO_{16} je buzen taktem 1 MHz, který je možno odebírat prostřednictvím děliče dvěma či čtyřmi - IO_{12} jako řídicí signál od mikroprocesoru (takt $Ø_m$). Cyklické čtení z obrazové paměti obstarává řadič, který mimo zmíněný takt 1 MHz (přiváděný na vstup QI) vyžaduje ještě další řídicí frekvenci $Ø_1$, představující znakový takt, sloužící ke generování obrazu v horizontálním směru. Generátorem IO_9 je vyráběna bodová frekvence (cca 12 MHz), kterou nutno jemně nastavit (při uvádění do chodu) potenciometrovým trimrem P_1. Dělič IO_7 podělením přetváří bodovou frekvenci na frekvenci znaků, mimo to řídí ještě posuvný registr IO_8.

Data určená k zobrazení přicházejí po datové sběrnici od mikroprocesoru. Prostřednictvím řízených vstupních latchů IO_4 a IO_5 jsou předávána do obrazové paměti; jedná-li se ovšem o řídicí slova řadiče, jsou přiváděna přes IO_5 přímo na IO_{16}, což plyne z funkce dekodéru IO_{10} a jeho buzení příslušnou adresou!

Paměti IO_1 a IO_2 jsou adresovány vnitřní adresovou sběrnicí A0 až A9, vycházející ze řadiče; rovněž vedení R0 až R2 představují řádkovou adresu pro generátor znaků. Výstupem INI řadiče je synchronisován IO_9, z výstupu SYNC řadiče jsou odebírány synchronisační signály H-sync a V-sync vytvářející

ve směsi s videosignálem tzv. BAS signál, jenž se již přivádí na vstup obrazovkového monitoru. Z výstupu PT se odebírá signál ukazovátka (kursoru), který v daném případě se projevuje blikáním nejspodnější hrany rastru v místě, do nějž má být zapsán další znak.

Z výstupu W řadiče se odebírá signál pro zápis, který je přiváděn spolu s bitem b3 z IO_5 na dvouvstupové hradlo N1. Je-li za výstupem N1 úroveň log "1", pak je zápis zablokovaný. Signálem ST je řízeno přebírání řídicího slova z IO_5 do řadiče. Spínači Ji za komparátorem IO_{11} se nastavují adresy pro uvolnění desky pro přenos dat a řízení.

IV. Soubor příkazů mikroprocesoru Z80

IV.1. Soupis jednotlivých příkazů v mnemotechnickém tvaru, abecedně řazený, spolu s jejich významy

ADC HL,ss (add with carry reg. pair ss to HL) - obsah registrového páru ss je přičten s bitem přenosu C k obsahu registrového páru HL

ADC A,s (add with carry operand s to acc.) - operand s spolu s bitem přenosu C je přičten k obsahu střadače A

ADD A,n (add value n to acc.) - integer n je přičten k obsahu střadače A

ADD A,r (add reg.r to acc.) - obsah registru r je přičten k obsahu střadače A

ADD A,(HL) (add location(HL) to acc.) - obsah paměťové buňky, jejíž adresa byla specifikována obsahem registrového páru HL, je přičten k obsahu střadače A

ADD A,(IX+d) (add location (IX+d) to acc.) - obsah paměťové buňky, jejíž adresa byla specifikována obsahem registru IX a doplňku d, je přičten k obsahu střadače A

ADD A,(AY+d) (add location (AY+d) to acc.) - jako u předešlé instrukce, avšak s obsahem registru IY

ADD HL,ss (add reg.pair ss to HL) - obsah registrového páru ss je přičten k obsahu registrového páru HL

ADD IX,pp (add reg.pair pp to IX) - obsah registrového páru pp je přičten k obsahu registru IX

ADD IY,rr (add reg.pair rr to IY) - obsah registrového páru rr je přičten k obsahu registru IY

AND s (logical "and" of operand s and acc.) - logický součin operandu s a střadače; výsledek je uložen ve střadači

BIT b,(HL) (test bit b of location (HL)) - po vykonání tohoto příkazu obsahuje příznak Z v registru F doplněk bitu indikovaného obsahem registrového páru HL

BIT b,(IX+d) (test bit b of location (IX+d)) - po vykonání tohoto příkazu obsahuje příznak Z v registru F doplněk bitu specifikovaného registrem IX a dvojkovým doplňkem d

BIT b,(IY+d) (test bit b of location (IY+d)) - jako u předešlé instrukce, ale specifikace dána součtem obsahu registru IY a dvojkového doplňku d

BIT b,r (test bit b of reg.r) - po vykonání příkazu obsahuje příznak Z v registru F doplněk indikovaného bitu s daným registrem r

CALL cc,nn (call subroutine at location nn if condition cc is true) - je-li splněna podmínka cc, uloží instrukce běžný obsah programového čítače PC na vrch paměťového zásobníku, pak vloží operand nn do PC k indikaci adresy v paměti, na níž začíná příslušná subrutina

CALL nn (unconditional call subroutine at location nn) - nepodmíněné volání po subrutině, začínající na adrese nn

CCF (complement carry flag) - příznak C v registru F je invertován

CP s (compare operand s with acc.) - obsah operandu s je srovnán s obsahem střadače; při shodě je nahozen příznak

CPD (compare location (HL) and acc., decrement HL and BC) - obsah paměťové buňky specifikované obsahem

registrového páru HL, je srovnán s obsahem střadače. Při shodě je nahozen podmínkový bit. Registrové páry LH a BC jsou dekrementovány.

CPDR (compare location (HL) and acc., decrement HL and BC, repeat until BC = 0) - obsah paměťové buňky, specifikované obsahem HL, je porovnáván s obsahem střadače A. Při shodě je nahozen podmínkový bit. Registrové páry BC a HL jsou dekrementovány. Jestliže BC - po dekrementování - má nulový obsah či je-li A = (HL), provádění instrukce je ukončeno. Pokud BC nemá nulový obsah a A je různé od (HL), je PC umenšen o 2 a průběh je opakován

CPI (compare location (HL) and acc., increment HL an decrement BC) - obsah paměťové buňky, specifikované obsahem HL, je porovnáván s obsahem střadače. V případě shody je nahozen podmínkový bit. HL je inkrementován a čítač slabik = reg. pár BC je dekrementován

CPIR (compare location (HL) and acc., increment HL, decrement BC, repest until BC = 0) - stejné jako u předcházející instrukce, avšak je-li BC nenulové a A $\neq$ (HL), programový čítač je dekrementován dvěma a průběh se opakuje. Jestliže po dekrementování obsah BC = 0 či A = = (HL), příkaz je ukončen

CPL (complement acc.) - obsah střadače je invertován - jedničkový doplněk

DAA (decimal adjust acc.) - tato instrukce pomíněně upravuje střadač pro operace sčítání a odečítání s BCD čísly

DEC m (decrement operand m) - slabika, specifikovaná operandem m, je dekrementována

DEC IX (decrement IX) - obsah indexového registru IX je dekrementován

DEC IY (decrement IY) - obsah indexového registru IY je dekrementován

DEC ss (decrement reg.pair ss) - obsah registrového páru ss je dekrementován

DI (disable interrupts) - zamezuje maskovatelné přerušení resetováním uvolňovacích klopných obvodů přerušení IFF_1 a IFF_2

DJNZ e (decrement B and jump relative if B ≠ 0) - registr B je dekrementován a jestliže jeho obsah je nenulový, hodnota doplňku e je přičtena k PC. Je vyzdvižena následující instrukce z místa specifikovaného novým obsahem PC. Jestliže obsah dekrementovaného registru B je nulový, provede se instrukce následující za příkazem DJNZ e.

EI (enable interrupts) - příkaz uvolňuje maskovatelné přerušení nahozením klopných obvodů IFF_1 a IFF_2

EX (SP),HL (exchange the location (SP) and HL) - dolní slabika obsažená v registrovém páru HL je vyměněna s obsahem paměťové buňky specifikované obsahem registrového páru SP, a horní slabika HL je vyměněna s obsahem následující paměťové buňky specifikované obsahem (SP+1)

EX (SP),IX (exchange the location (SP) and IX) - dolní slabika obsažená v registrovém páru IX je vyměněna s obsahem paměťové buňky specifikované obsahem registrového páru SP, horní slabika (= slabika vyššího řádu) pak s obsahem buňky specifikované obsahem (SP+1)

EX (SP),IY (exchange the location (SP) and IY) - jako u předcházející instrukce, vztaženo však na obsah IY

EX AF, AF′ (exchange the contents of AF and AF′) - příkazem jsou vyměněny dvouslabičné obsahy registrových párů AF a AF′

EX DE,HL (exchange the contents of DE and HL) - příkazem jsou vyměněny dvouslabičné obsahy registrových párů DE a HL

EXX (exchange the contents of BC,DE,HL with contents of BC′, DE′, HL′ respectively) - příkazem jsou vyměněny všechny dvouslabičné obsahy odpovídajících registrových párů

HALT (wait for interrupt or reset) - při příkazu HALT jsou operace mikroprocesoru dočasně zastaveny až do příjmu přerušení či iniciace. Během stavu HALT generuje mikroprocesor prázdné instrukce NOP za účelem osvěžování

IM 0 (set interrupt mode 0) - zavedení přerušovacího způsobu 0

IM 1 (set interrupt mode 1) - zavedení přerušovacího způsobu 1, restart na adrese 0038H

IM2 (set interrupt mode 2) - zavedení přerušovacího způsobu 2, nepřímé volání po kterémkoliv místě v paměti

IN A,(n) (load the acc. with input from device n) - příkazem je umístěn operand n na dolní polovinu adresové sběrnice za účelem adresování V/V zařízení na jednom z možných 256 kanálů. Horní slabika adresové sběrnice pak nese obsah střadače A; zvoleným kanálem je přenesena osmibitová informace na datovou sběrnici a zapsána pak do střadače A

IN r,(C) (load the reg.r with input from device (C)) - obsah registru C je umístěn na dolní polovinu adresové sběrnice za účelem adresování V/V zařízení. Horní slabika adresové sběrnice nese obsah registru B; specifikovaným kanálem je přenesena osmibitová informace z V/V zařízení na datovou sběrnici a zapsána pak do registru r mikroprocesoru

INC (HL) (increment location (HL)) - slabika, obsažená v buňce adresově specifikované obsahem registrového páru HL, je inkrementována

INC IX (increment IX) - obsah šestnáctibitového indexového registru IX je inkrementován

INC (IX+d) (increment location (IX+d)) - obsah indexového registru IX je přičten k dvojkovému doplňku d k specifikování adresy v paměti; tato adresa je pak inkrementována

INC IY (increment IY) - obsah šestnáctibitového indexového registru IY je inkrementován

INC (IY+d) (increment location (IY+d)) - obsah indexového registru IY je přičten k dvojkovému doplňku d k specifikování adresy v paměti; tato adresa je pak inkrementována

INC r (increment reg.r) - příkazem je inkrementován obsah registru r

INC ss (increment reg. pair ss) - příkazem je inkrementován obsah registrového páru ss

IND (load location (HL) with input from port (C), decrement HL end B) - obsah registru C je umístěn na dolní polovinu adresové sběrnice k výběru V/V zařízení. Registr B může být použit jako čítač slabik - jeho obsah je umístěn na horní polovinu adresové sběrnice. Informace z V/V zařízení je vložena na datovou sběrnici a převzata procesorem. Obsah registrového páru HL je umístěn pak na adresovou sběrnici a převzatý vstupní byte je zapsán do specifikované paměťové buňky párem HL. Dále pak je čítač slabik a HL dekrementován

INDR (load location (HL) with input from port (C), decrement HL and B, repeat until B = 0) - tímto příkazem je vyvoláno opakované provedení příkazu IND, které končí, jakmile je obsah B po dekrementování roven nule

INI (load location (HL) with input from port (C), increment HL and decrement B) - obsah registru C je umístěn na dolní polovinu adresové sběrnice k výběru V/V zařízení. Registr B může být použit jako čítač slabik - jeho obsah je umístěn na horní polovinu adresové sběrnice. Informace z V/V zařízení je vložena na datovou sběrnici a převzata procesorem. Obsah registrového páru HL je pak umístěn na adresovou sběrnici a převzatá vstupní informace je zapsána do paměťové buňky specifikované párem HL. Dále je pak čítač slabik dekrementován, registrový pár HL však inkrementován

INIR (load location (HL) with input from port (C), increment HL and decrement B, repeat until B = 0) - tímto příkazem je vyvoláno opakované provedení příkazu INI, jež končí, jakmile je obsah B po dekrementování roven nule

JP (HL) (unconditional jump to (HL)) - do PC je uložen obsah páru HL. Následující instrukce je dána adresou odpovídající novému obsahu programového čítače PC

JP (IX) (unconditional jump to (IX)) - do PC je uložen obsah registru IX. Následující instrukce je dána adresou odpovídající novému obsahu čítače PC

JP (IY) (unconditional jump to (IY)) - do PC je uložen obsah registru IY. Následující instrukce je dána adresou odpovídající novému obsahu programového čítače PC

JP cc,nn (jump to location nn if condition cc is true) - je-li splněna podmínka cc, je příkazem uložen operand nn do PC a program pokračuje na instrukci začínající na adrese nn. Při nesplnění podmínky cc je PC inkrementován a program pokračuje na následné instrukci po JP cc,nn

JP nn (unconditional jump to location nn) - operand nn je vložen do registrového páru PC a tak specifikuje adresu následující programové instrukce, jež bude provedena.

JR C,e (jump relative to PC if carry = 1) - je-li příznak C = 0, program sekvenčně pokračuje, je-li však C = 1, následuje skok. V tom případě displacement e je přičten k obsahu PC a následující instrukce je dána adresou odpovídající novému obsahu PC

JR e (unconditional jump relative to PC+e) - příkaz k nepodmíněnému skoku na adresu, specifikovanou obsahem PC s přičteným doplňkem e (v rozmezí -126 až +129 slabik)

JR NC,e (jump relative to PC+e if carry = 0) - je-li příznak C = 1, program sekvenčně pokračuje, je-li však C = 0, následuje skok na instrukci na adrese, specifikované obsahem PC s přičteným doplňkem e

JR NZ,e (jump relative to PC+e if non zero) - je-li příznak Z = 1, pak program sekvenčně pokračuje, je-li však Z = 0, následuje skok na adresu, specifikovanou obsahem PC s přičteným doplňkem e

JR Z,e (jump relative to PC+e if zero) - je-li příznak Z = 0, pak program sekvenčně pokračuje, je-li však Z = 1, následuje skok na adresu, specifikovanou obsahem PC s přičteným doplňkem e

LD A,(BC) (load acc. with location (BC)) - do střadače je uložen obsah paměťové buňky z adresy, specifikované obsahem registrového páru BC

LD A,(DE) (load acc. with location (DE)) - do střadače je uložen obsah paměťové buňky z adresy, specifikované obsahem registrového páru DE

LD A,I (load acc. with reg.I) - do střadače je uložen přerušovací vektor z registru I

LD A,(nn) (load acc. with location nn) - obsah paměťové buňky, jejíž adresa je specifikována operandem nn, je uložen do střadače A. První operand n je byte nižšího řádu dvouslabičné adresy

LD A,R (load acc. with reg.R) - do střadače je uložen obsah osvěžovacího registru R

LD (BC),A (load location (BC) with acc.) - příkazem je uložen obsah střadače A do paměťové buňky, specifikované obsahem registrového páru BC

LD (DE),A (load location (DE) with acc.) - viz výše, avšak registrovým párem DE

LD (HL),n (load location (HL) with value n) - příkazem je uložen integer n do paměťové buňky specifikované obsahem registrového páru HL

LD dd,nn (load reg.pair dd with value nn) - příkazem je uložen dvouslabičný integer nn do registrového páru dd (= BC, DE, HL nebo SP; OP kód určuje zvolený pár takto: 0 0 d d 0 0 0 1, kde dd je pro BC = 00, pro DE = 01, pro HL = 10, pro SP = 11)

LD HL,(nn) (load HL with location (nn)) - obsah paměťové buňky o adrese nn je uložen do registru L, obsah následné paměťové buňky o adrese (nn+1) je uložen do registru H

LD (HL),r (load location (HL) with reg.r) - obsah registru r je uložen do paměťové buňky specifikované obsahem registrového páru HL

LD I,A (load I with acc.) - obsah střadače A je uložen jako přerušovací vektor do registru přerušení I

LD IX,nn (load IX with value nn) - integer nn je uložen do indexového registru IX

LD IX,(nn) (load IX with location (nn)) - obsah buňky o adrese nn je uložen do dolní slabiky registrového páru IX, obsah buňky o adrese (nn+1) je uložen do horní slabiky IX

LD (IX+d),n (load location (IX+d) with value n) - operand n je uložen do buňky o adrese specifikované součtem obsahu registru IX a dvojkového doplňku d

LD dd,(nn) (load reg.pair dd with location (nn)) - obsah paměťové buňky o adrese nn je uložen do dolní slabiky reg. páru dd, obsah buňky o adrese (nn+1) je uložen do horní slabiky reg. páru dd

LD (IX+d),r (load location (IX+d) with reg.r) - obsah registru r je uložen do buňky o adrese, specifikované součtem obsahu registru IX a dvojkového doplňku d

LD IY,nn (load IY with value nn) - integer nn je uložen do indexového registru IY

LD IY,(nn) (load IY with location (nn)) - obsah buňky o adrese nn je uložen do dolní slabiky reg. páru IY, obsah buňky o adrese (nn+1) je uložen do horní slabiky IY

LD (IY+d),n (load location (IY+d) with value n) - operand n je uložen do buňky o adrese specifikované součtem obsahu reg. IY a dvojkového doplňku d

LD (IX+d),r (load location (IX+d) with reg.r) - obsah registru r je uložen do buňky o adrese specifikované součtem obsahu registru IY a dvojkového doplňku d

LD (nn),A (load location (nn) with acc.) - obsah střadače A je uložen do buňky o adrese specifikované operandem nn. První operand n je dolní byte místa nn

LD (nn),dd (load location (nn) with reg.pair dd) - dolní byte registrového páru dd je uložen do buňky o adrese nn, horní byte je uložen do buňky (nn+1)

LD (nn),HL (load location (nn) with reg. pair HL) - obsah registru L je uložen do buňky o adrese nn, obsah registru H je uložen do buňky o adrese (nn+1)

LD (nn),IX (load location (nn) with reg.pair IX) - obsah dolního byte reg. IX je uložen do buňky o adrese nn, horní byte pak do buňky (nn+1)

LD (nn),IY (load location (nn) with reg.pair IY) - obsah dolního byte reg. IY je uložen do buňky o adrese nn, horní byte pak do buňky (nn+1)

LD R,A (load R with acc.) - obsah střadače A je uložen do osvěžovacího registru R

LD r,(HL) (load reg.r with location (HL)) - obsah paměťové buňky o adrese specifikované obsahem reg.páru HL je uložen do registru r

LD r,(IX+d) (load reg.r with location (IX+d)) - součet obsahů registru IX a dvojkového doplňku d je uložen do registru r

LD r,(IY+d) (load reg.r with location (IY+d)) - součet obsahů registru IY a dvojkového doplňku d je uložen do registru r

LD r,n (load reg.r with value n) - operand n je uložen do registru r

LD r,r´ (load reg.r with reg.r´) - obsah registru r´ je uložen do registru r

LD SP,HL (load SP with HL) - obsah registrového páru HL je uložen do ukazatele sklípku - stack pointer

LD SP,IX (load SP with IX) - obsah reg. IX je uložen do SP

LD SP,IY (load SP with IY) - obsah reg. IY je uložen do SP

LDD (load location (DE) with location (HL), decrement DE, HL and BC) - touto dvouslabičnou instrukcí je způsoben přenos datové slabiky z buňky o adrese specifikované obsahem HL do buňky o adrese, specifikované obsahem páru DE. Oba registrové páry jsou pak včetně BC (čítače slabik) dekrementovány

LDDR (load location (DE) with location (HL), decrement DE, HL and BC, repeat until BC = 0) - příkaz vy-

volá stejnou funkci jako u předcházejícího příkazu LDD, ovšem s tím, že průběh je opakován, pokud obsah BC po dekrementaci není roven nule. Je-li obsah BC nulový, provádění příkazu se zastaví

LDI (load location (DE) with location (HL), increment DE, HL, decrement BC) - datová slabika je transferována z buňky adresované obsahem páru HL do buňky, adresované obsahem páru DE. Pak jsou oba páry inkrementovány, avšak BC je dekrementován

LDIR (load location (DE) with location (HL), increment DE, HL, decrement BC, repeat until BC = 0) - stejná funkce jako u LDI, ovšem s tím, že je opakována, pokud BC $\neq$ 0. Je-li obsah BC nulový, provádění se zastaví

NEG (negate acc. - 2's complement) - obsah střadače je negován, což je stejné, jako kdyby byl obsah střadače A odečten od nuly

NOP (no operation) - během tohoto příkazu neprovádí mikroprocesor nějakou operaci vyjma osvěžování

OR s (logical "OR" of operand s and acc.) - je prováděna bit po bitu logická operace "NEBO" se slabikou specifikovanou operandem s, a obsaženou ve střadači. Výsledek operace je uložen opět ve střadači A

OTDR (load output port (C) with location (HL), decrement HL and B, repeat until B = 0) - obsah páru HL je umístěn na adresovou sběrnici k určení místa v paměti. Slabika obsažená v tomto místě je dočasně uložena v procesoru. Potom, až je čítač slabik B dekrementován, obsah registru C je umístěn na dolní polovinu adresové sběrnice za účelem výběru V/V zařízení. Registr B může být použit jako čítač slabik, a jeho dekrementovaný obsah je umístěn do horní poloviny adresové sběrnice. Slabika, určená k vydání, je umístěna na datovou sběrnici a zapsána do vybrané periférie. Registrový pár HL je pak dekrementován. Pokud dekrementovaný registr B je nenulový, je PC dekrementován dvěmi a instrukce je zopakována. Je-li B = 0, provádění se zastaví. Přerušení je rozpoznáno po každém přenosu data

OTIR (load output port (C) with location (HL), increment HL, decrement B, repeat until B = 0) - obdobný průběh jako u příkazu "OTDR", ovšem s tím rozdílem, že registrový pár HL po transferu data je i n k r e m e n t o v á n

OUT (C),r (load output port (C) with reg.r) - obsah registru C je umístěn na dolní polovinu adresové sběrnice k určení V/V zařízení. Obsah registru B je umístěn na horní polovinu adresové sběrnice. Slabika, obsažená v registru r je umístěna na datovou sběrnici a zapsaná (předaná) perifernímu zařízení

OUT (n),A (load output port (n) with acc.) - operand d je umístěn na dolní polovinu adresové sběrnice k adresování V/V zařízení. Obsah střadače se objeví na horní polovině adresové sběrnice. Slabika, obsažená ve střadači je vydána na datovou sběrnici a převzata vybranou periférií

OUTD (load output port (C) with location (HL), decrement HL and B) - obsah registrového páru HL je umístěn na adresovou sběrnici za účelem určení místa v paměti. Slabika, obsažená v tomto místě je dočasně uložena v procesoru. Potom, až je čítač slabik B dekrementován, obsah registru C je vložen na dolní polovinu adresové sběrnice k vybrání V/V zařízení. Registr B může být použit jako čítač slabik a jeho dekrementovaná hodnota je umístěna na horní polovinu adresové sběrnice. Slabika, určená k vydání, je vložena na datovou sběrnici a převzata zvoleným periferním zařízením. Posléze je reg. pár HL dekrementován

OUTI (load output port (C) with location (HL), increment HL, decrement B) - stejný průběh jako u předcházející instrukce OUTD, ovšem s tím, že po transferu data je registrový pár HL i n k r e m e n t o v á n

POP IX (load IX with top of stack) - vrcholové dvě slabiky ze zásobníku (stack) jsou vloženy do registru IX,a to tak, že do dolní poloviny IX přijde SP, do horní poloviny IX pak (SP+1)

POP IY (load IY with top of stack) - vrcholové dvě slabiky ze zásobníku (stack) jsou vloženy do registru IY v pořadí IY_L = SP, IY_H = SP+1

POP qq (load reg.pair qq with top of stack) - vrcholové dvě slabiky ze zásobníku jsou vloženy do reg.páru qq v pořadí qqL = SP, qqH = SP+1. qq representuje páry BC, DE, HL nebo AF; příslušný OP kód je určen tvarem:

I	I	q	q	0	0	0	I
b7							b0

PUSH IX (load IX onto stack) - obsah indexového registru IX je vložen do zásobníku. Tento příkaz nejprve dekrementuje SP a uloží byte vyššího řádu IX do buňky o adrese specifikované nyní ukazatelem zásobníku SP; dále dekrementuje opět SP a uloží dolní byte (= nižšího řádu) do paměťového místa o odpovídající adrese z ukazatele zásobníku

PUSH IY (load IY onto stack) - příkaz je analogický k předcházejícímu, ovšem transfer se děje z registru IY

PUSH qq (load reg. pair qq onto stack) - obsah registrového páru qq je vložen do zásobníku. SP je dekrementován a do jím specifikované buňky o adrese (SP-1) je uložen horní byte páru qq. Pak je znovu SP dekrementován a do takto vzniklé adresy SP-2 je uložen dolní byte páru qq

RES b,m (reset bit b of operand m) - bit b v operandu m je resetován

RET (return from subroutine) - příkazem je způsoben návrat do hlavního programu z podprogramu, a to vložením SP (dolní byte) do PC-L a vložením obsahu (SP-1) (horní byte) do PC-H, kamž byla tato adresa návratu předtím umístěna příkazem "CALL"

RETI (return from interrupt) - příkaz je použit na konci obslužné přerušovací rutiny k:
1) upamatování obsahu programového čítače PC,
2) indikaci V/V zařízení, že rutina byla ukončena.
Příkaz současně resetuje klopné obvody IFF1 a IFF2

RETN (return from nonmaskable interrupt) - návrat z nemaskovatelného přerušení; pracuje stejně jako příkaz "RET". Stav obvodu IFF2 je zpětně kopírován do IFF1 do stavu, který měl před přijetím signálu "NMI"

RL m (rotate left through carry operand m) - operand m je rotován (o jeden bit) doleva . b7 je kopírován do přenosového příznaku C a předcházející obsah C je kopírován do b0

RLA (rotate left acc. through carry) - obsah střadače je rotován doleva , obsah b7 je kopírován do přenosového příznaku C a předcházející obsah C je kopírován do b0

RLC (HL) (rotate location (HL) left circular) - obsah buňky o adrese specifikované obsahem páru HL je rotován doleva. Obsah bitu b7 je kopírován do příznaku přenosu C a taktéž do bitu b0

RLC (IX+d) (rotate location (IX+d) left circular) - obsah buňky o adrese specifikované součtem obsahu reg. páru IX a doplňku d je rotován doleva . Obsah bitu b7 je kopírován do příznaku přenosu C a taktéž do bitu b0

RLC (IY+d) (rotate location (IY+d) left circular) - obsah buňky o adrese specifikované součtem obsahu reg. páru IY a doplňku d je rotován doleva . Obsah b7 je kopírován do příznaku C a taktéž do b0

RLC r (rotate reg.r left circular) - osmibitový obsah registru r je rotován doleva. Obsah b7 je kopírován do příznaku přenosu C a rovněž do b0

RLCA (rotate left circular acc.) - obsah střadače A je rotován doleva . Obsah bitu b7 je kopírován do příznaku přenosu C a rovněž i do bitu b0

RLD (rotate digit left and right between acc. and location (HL)) - obsah čtyř bitů nižšího řádu paměťového místa (HL) je kopírován do čtyř bitů vyššího řádu téhož paměťového místa. Předcházející obsah těch-

to čtyř bitů vyššího řádu je kopírován do čtyř bitů nižšího řádu střadače A a předcházející obsah těchto 4 bitů nižšího řádu střadače je kopírován do 4 bitů nižšího řádu paměťového místa (HL). Obsah 4 bitů vyššího řádu střadače A je nedotčen

RR m (rotate right through carry operand m) - obsah operandu m je rotován doprava . Obsah b0 je kopírován do příznaku přenosu C a předcházející obsah C je kopírován do b7

RRA (rotate right acc. through carry) - obsah střadače A je rotován doprava. Obsah b0 je kopírován do příznaku přenosu C a předcházející obsah C je kopírován do b7

RRC m (rotate operand m right circular) - obsah operandu m je rotován doprava . Obsah b0 je kopírován do příznaku přenosu C a taktéž do bitu b7

RRCA (rotate right circular acc.) - obsah střadače A je rotován doprava . Obsah bitu b0 je kopírován do bitu b7 a též do příznaku přenosu C

RRD (rotate digit right and left between acc. and location (HL)) - obsah dolní tetrády paměťového místa (HL) je kopírován do dolní tetrády střadače A. Předcházející obsah dolní tetrády střadače A je kopírován do horní tetrády místa (HL), a předcházející obsah horní tetrády (HL) je kopírován do dolní tetrády (HL). Obsah horní tetrády střadače je nedotčen

RST p (restart to location p) - obsah programového čítače PC je umístěn do paměťového zásobníku (stack) a paměťové místo nulté stránky, dané operandem p, je vloženo do PC. Program pak pokračuje s instrukcí na adrese označené obsahem PC. Instrukce dovoluje skok na jednu z osmi adres níže vyznačených: I I t t t I I I
b7 b0
kde pro parametr p a bity t platí:

p	00H	08H	10H	18H	20H	28H	30H	38H
ttt	000	001	010	011	100	101	110	111

RET cc (return from subroutine if condition cc is true) - je-li splněna podmínka cc, je řízení programu vráceno do hlavní části snesením předcházejícího obsahu PC uloženého na vrcholku zásobníku, tedy PC-L←← (SP), PC-H ←— (SP+1)

SBC A,s (subtract operand s from acc. with carry) - operand s a přenosový bit C je odečten od obsahu střadače, výsledek uložen ve střadači

SBC HL,ss (subtract reg.pair ss from HL with carry) - obsah registrového páru ss a přenosový bit C jsou odečteny od obsahu reg. páru HL, výsledek je uložen v HL

SCF (set carry flag - C = 1) - příznak přenosu (= bit C) v registru F je nahozen, tj. = I

SET b,(HL) (set bit b of location (HL)) - bit b v paměťovém místě určeném obsahem registrového páru HL, je nahozen

SET b,(IX+d) (set bit b of location (IX+d)) - bit b v paměťovém místě určeném obsahem reg. páru IX a doplňku d, je nahozen

SET b,(IY+d) (set bit b of location (IY+d)) - bit b v místě určeném obsahem reg. páru IY a doplňku d je nahozen

SET b,r (set bit b of reg.r) - bit b (kterýkoliv) v registru r je nahozen (= I)

SLA m (shift operand m left arithmetic) - obsah operandu m je posunut doleva , přičemž b0 je resetován a b7 vložen do přenosového bitu C

SRA m (shift operand m right arithmetic) - obsah operandu m je posunut doprava , přičemž bit b0 je vložen do příznaku přenosu C. Obsah bitu b7 zůstává zachován

SRL m (shift operand m right logical) - obsah operandu m je posunut doprava logicky. Bit b0 je vložen do příznaku přenosu C a bit b7 je resetován

SUB s (subtract operand s from acc.) - operand s je odečten od obsahu střadače a výsledek je uložen ve střadači A

XOR s (exclusive "OR" of operand s and acc.) - logická operace "výlučné nebo", bit po bitu, mezi bytem určeným operandem s a bytem, obsaženým ve střadači A

Legenda

r, r´ - jeden z registrů A, B, C, D, E, H nebo L
n - jednoslabičný výraz v rozsahu 0 až 255
nn - dvouslabičný výraz v rozsahu 0 až 65 535
d - osmibitový doplněk v rozmezí -128 až +127
b - výraz v rozmezí 0 až 7
e - osmibitový výraz v rozmezí -126 až +129
qq - některý z registrových párů BC, DE, HL či AF
ss - některý z registrových párů BC, DE, HL či SP
pp - některý z registrových párů BC, DE, IX či SP
rr - některý z registrových párů BC, DE, IY či SP
s - některý z výrazů r, n, (HL), (IX+d) či (IY+d)
dd - některý z registrových párů BC, DE, HL či SP
m - některý z výrazů r, (HL), (IX+d) či (IY+d)
(HL) - specifikuje paměťovou buňku (místo) o adrese udané obsahem registrového páru HL
(nn) - specifikuje paměťovou buňku o adrese dané dvouslabičným výrazem v nn
PC - programový čítač (program counter)
SP - ukazatel zásobníku (st k pointer)
t - výraz v rozmezí 0 až 7
C, N, P/V, H, Z, S - příznakové bity
cc - stav příznaků pro podmíněné příkazy JR a JP skoků

cc	podmínka	relevantní příznak
000	NZ non zero	Z
001	Z zero	Z
010	NC non carry	C
011	C carry	C
100	PO parity odd	P/V
101	PE parity even	P/V
110	P sign positive	S
111	M sign negative	S

IV.2. Instrukční soubor mikroprocesoru Z80 - znázorněný ve strojních cyklech M1 až M5

Legenda

IO - vnitřní operace CPJ
MR - čtení z paměti
MRH - čtení z paměti vyšší slabiky
MRL - čtení z paměti nižší slabiky
MW - zápis do paměti
MWH - zápis do paměti vyšší slabiky
MWL - zápis do paměti nižší slabiky
OCF - zachycení operačního kódu
ODH - čtení vyšší slabiky datového operandu } OD
ODL - čtení nižší slabiky datového operandu }
PR - čtení z portu (brány)
PW - zápis do portu (brány)
SRH - čtení vyšší slabiky zápisníkové paměti (stack)
SRL - čtení nižší slabiky zápisníkové paměti
SWH - zápis vyšší slabiky do zápisníkové paměti
SWL - zápis nižší slabiky do zápisníkové paměti
( ) - počet taktů T v daném strojním cyklu Mi

INSTRUCTION TYPE	BYTES	M1	M2	M3	M4	M5
LD r, s	1	OCF (4)				
LD r, n	2	OCF (4)	OD (3)			
LD r, (HL) LD (HL), r	1	OCF (4) OCF (4)	MR (3) MW (3)			
LD r, (IX+d) LD (IX+d), r	3	OCF (4)/OCF (4) OCF (4)/OCF (4)	OD (3) OD (3)	IO (5) IO (5)	MR (3) MW (3)	
LD (HL), n	2	OCF (4)	OD (3)	MW (3)		
LD A, $(^{BC}_{DE})$	1	OCF (4)	MR (3)			
LD $(^{BC}_{DE})$, A		OCF (4)	MW (3)			
LD A, (nn) LD (nn), A	3	OCF (4) OCF (4)	ODL (3) ODL (3)	ODH (3) ODH (3)	MR (3) MW (3)	
LD A, $^{I}_{R}$	2	OCF (4)/OCF(5)				
LD $^{I}_{R}$, A						
LD dd, nn	3	OCF (4)	ODL (3)	ODH (3)		
LD IX, nn	4	OCF (4)/OCF (4)	ODL (3)	ODH (3)		
LD HL, (nn) LD (nn), HL	3	OCF (4) OCF (4)	ODL (3) ODL (3)	ODH (3) ODH (3)	MRL (3) MWL (3)	MRH (3) MWH (3)

LD dd, (nn)
LD (nn), dd
LD IX, (nn)
LD (nn), IX
	4	OCF (4)/OCF (4)
		OCF (4)/OCF (4)
		OCF (4)/OCF (4)
		OCF (4)/OCF (4)
			ODL (3)
			ODL (3)
			ODL (3)
			ODL (3)
				ODH (3)
				ODH (3)
				ODH (3)
				ODH (3)
					MRL (3)
					MWL (3)
					MRL (3)
					MWL (3)
						MRH (3)
						MWH (3)
						MRH (3)
						MWH (3)

LD SP, HL	1	OCF (6)				
LD SP, IX	2	OCF (4)/OCF (6)				
PUSH qq	1	OCF (5) SP-1	SWH (3) SP-1	SWL (3)		
PUSH IX	2	OCF (4)/OCF (5) SP-1	SWH (3) SP-1	SWL (3)		
POP qq	1	OCF (4)	SRH (3) SP+1	SRL (3)	SP+1	
POP IX	2	OCF (4)/OCF (4)	SRH (3) SP+1	SRL (3)	SP+1	
EX DE, HL	1	OCF (4)				
EX AF, AF'	1	OCF (4)				

INSTRUCTION TYPE	BYTES	M1	M2	M3	M4	M5
EXX	1	OCF (4)				
EX (SP), HL	1	OCF (4)	SRL (3) SP+1 →	SRH (4)	SWH (3) SP-1 →	SWL (5)
EX (SP), IX	2	OCF (4)/OCF (4)	SRL (3) SP+1 →	SRH (4)	SWH (3) SP-1 →	SWL (5)
LDI LDD CPI CPD	2	OCF (4)/OCF (4)	MR (3)	MW (5)		
LDIR LDDR CPIR CPDR	2	OCF (4)/OCF (4)	MR (3)	MW (5)	IO (5)* *only if BC ≠ 0	
ALU A, r ADD ADC SUB SBC AND OR XOR CP	1	OCF (4)				
ALU A, n	2	OCF (4)	OD (3)			
ALU A, (HL)	1	OCF (4)	MR (3)			
ALU A, (IX+d)	3	OCF (4)/OCF (4)	OD (3)	IO (5)	MR (3)	
DEC INC r	1	OCF (4)				
DEC INC (HL)	1	OCF (4)	MR (4)	MW (3)		
DEC INC (IX+D)	2	OCF (4)/OCF (4)	OD (3)	IO (5)	MR (4)	MW (3)
DAA CPL CCF SCF NOP HALT DI EI	1	OCF (4)				
NEG IMO IM1 IM2	2	OCF (4)/OCF (4)				
ADD HL, ss	1	OCF (4)	IO (4)	IO (3)		
ADC HL, ss SBC HL, ss ADD IX, pp	2	OCF (4)/OCF (4)	IO (4)	IO (3)		
INC ss DEC ss	1	OCF (6)				
DEC IX INC IX	2	OCF (4)/OCF (6)				
RLCA RLA RRCA RRA	1	OCF (4)				
RLC r RL RRC RR SLA SRA SRL	2	OCF (4)/OCF (4)				

INSTRUCTION TYPE	BYTES	M1	M2	M3	M4	M5
RLC (HL) RL RRC RR SLA SRA SRL	2	OCF (4)/OCF (4)	MR (4)	MW (3)		
RLC (IX+d) RL RRC RR SLA SRA SRL	4	OCF (4)/OCF (4)	OD (3)	IO (5)	MR (4)	MW (3)
RLD RRD	2	OCF (4)/OCF (4)	MR (3)	IO (4)	MW (3)	
BIT b, r SET RES	2	OCF (4)/OCF (4)				
BIT b, (HL)	2	OCF (4)/OCF (4)	MR (4)			
SET b, (HL) RES	2	OCF (4)/OCF (4)	MR (4)	MW (3)		
BIT b, (IX+d)	4	OCF (4)/OCF (4)	OD (3)	IO (5)	MR (4)	
SET b, (IX+d) RES	4	OCF (4)/OCF (4)	OD (3)	IO (5)	MR (4)	MW (3)
JP nn JP cc, nn	3	OCF (4)	ODL (3)	ODH (3)		
JR e	2	OCF (4)	OD (3)	IO (5)		
JR C, e JR NC, e JR Z, e JR NZ, e	2	OCF (4)	OD (3)	IO (5)* * If condition is met		
JP (HL)	1	OCF (4)				
JP (IX)	2	OCF (4)/OCF (4)				
DJNZ, e	2	OCF (5)	OD (3)	IO (5)* * If B≠ 0		
CALL nn CALL cc, nn cc true	3	OCF (4)	ODL (3)	ODH (4) SP-1	SWH (3) SP-1	SWL (3)
CALL cc, nn cc false	3	OCF (4)	ODL (3)	ODH (3)		
RET	1	OCF (4)	SRL (3) SP+1	SRH (3)	SP+1	
RET cc	1	OCF (5)	SRL (3)* * If cc is true SP+1	SRH (3)*	SP+1	
RETI RETN	2	OCF (4)/OCF (4)	SRL (3) SP+1	SRH (3)	SP+1	
RST p	1	OCF (5) SP-1	SWH (3) SP-1	SWL (3)		

INSTRUCTION TYPE	BYTES	M1	M2	M3	M4	M5
IN A, (n)	2	OCF (4)	OD (3)	PR (4)		
IN r, (c)	2	OCF (4)/OCF (4)	PR (4)			
INI IND	2	OCF (4)/OCF (5)	PR (4)	MW (3)		
INIR INDR	2	OCF (4)/OCF (5)	PR (4)	MW (3)	IO (5)	
OUT (n), A	2	OCF (4)	OD (3)	PW (4)		
OUT (C), r	2	OCF (4)/OCF (4)	PW (4)			
OUTI OUTD	2	OCF (4)/OCF (5)	MR (3)	PW (4)		
OTIR OTDR	2	OCF (4)/OCF (5)	MR (3)	PW (4)	IO (5)	
INTERRUPTS						
NMI	–	OCF (5) * SP-1 →	SWH (3) SP-1 →	SWL (3)	*Op Code Ignored	
INT						
MODE 0	–	INTA (6) (CALL INSERTED)	ODL (3)	ODH (4) SP-1 →	SWH (3) SP-1 →	SWL (3)
	–	INTA (6) (RST INSERTED) SP-1 →	SWH (3) SP-1 →	SWL (3)		
MODE 1		INTA (7) (RST 38H INTERNAL) SP-1 →	SWH (3) SP-1 →	SWL (3)		
MODE 2	–	INTA (7) (VECTOR SUPPLIED) SP-1 →	SWH (3) SP-1 →	SWL (3)	MRL (3)	MRH (3)

IV.3 INSTRUKČNÍ SOUBOR Z80 (OP-CODE)

Seznam instrukcí Z80-CPU seřazených abecedně podle mnemonických příkazů assembleru

OBJ CODE	SOURCE STATEMENT
8E	ADC A,(HL)
DD8E05	ADC A,(IX+d)
FD8E05	ADC A,(IY+d)
8F	ADC A,A
88	ADC A,B
89	ADC A,C
8A	ADC A,D
8B	ADC A,E
8C	ADC A,H
8D	ADC A,L
CE20	ADC A,N
ED4A	ADC HL,BC
ED5A	ADC HL,DE
ED6A	ADC HL,HL
ED7A	ADC HL,SP
86	ADD A,(HL)
DD8605	ADD A,(IX+d)
FD8605	ADD A,(IY+d)
87	ADD A,A
80	ADD A,B
81	ADD A,C
82	ADD A,D
83	ADD A,E
84	ADD A,H
85	ADD A,L
C620	ADD A,N
09	ADD HL,BC
19	ADD HL,DE
29	ADD HL,HL
39	ADD HL,SP
DD09	ADD IX,BC
DD19	ADD IX,DE
DD29	ADD IX,IX
DD39	ADD IX,SP
FD09	ADD IY,BC
FD19	ADD IY,DE
FD29	ADD IY,IY
FD39	ADD IY,SP
A6	AND (HL)
DDA605	AND (IX+d)
FDA605	AND (IY+d)
A7	AND A
A0	AND B
A1	AND C
A2	AND D
A3	AND E
A4	AND H
A5	AND L
E620	AND N
CB46	BIT 0,(HL)
DDCB0546	BIT 0,(IX+d)
FDCB0546	BIT 0,(IY+d)
CB47	BIT 0,A
CB40	BIT 0,B
CB41	BIT 0,C
CB42	BIT 0,D
CB43	BIT 0,E
CB44	BIT 0,H
CB45	BIT 0,L
CB4E	BIT 1,(HL)
DDCB054E	BIT 1,(IX+d)
FDCB054E	BIT 1,(IY+d)
CB4F	BIT 1,A
CB48	BIT 1,B
CB49	BIT 1,C
CB4A	BIT 1,D
CB4B	BIT 1,E
CB4C	BIT 1,H
CB4D	BIT 1,L
CB56	BIT 2,(HL)
DDCB0556	BIT 2,(IX+d)
FDCB0556	BIT 2,(IY+d)
CB57	BIT 2,A
CB50	BIT 2,B
CB51	BIT 2,C
CB52	BIT 2,D
CB53	BIT 2,E
CB54	BIT 2,H
CB55	BIT 2,L
CB5E	BIT 3,(HL)
DDCB055E	BIT 3,(IX+d)
FDCB055E	BIT 3,(IY+d)
CB5F	BIT 3,A
CB58	BIT 3,B
CB59	BIT 3,C
CB5A	BIT 3,D
CB5B	BIT 3,E
CB5C	BIT 3,H
CB5D	BIT 3,L
CB66	BIT 4,(HL)
DDCB0566	BIT 4,(IX+d)
FDCB0566	BIT 4,(IY+d)
CB67	BIT 4,A
CB60	BIT 4,B
CB61	BIT 4,C
CB62	BIT 4,D
CB63	BIT 4,E
CB64	BIT 4,H
CB65	BIT 4,L
CB6E	BIT 5,(HL)
DDCB056E	BIT 5,(IX+d)
FDCB056E	BIT 5,(IY+d)
CB6F	BIT 5,A
CB68	BIT 5,B
CB69	BIT 5,C
CB6A	BIT 5,D
CB6B	BIT 5,E
CB6C	BIT 5,H
CB6D	BIT 5,L
CB76	BIT 6,(HL)
DDCB0576	BIT 6,(IX+d)
FDCB0576	BIT 6,(IY+d)
CB77	BIT 6,A
CB70	BIT 6,B
CB71	BIT 6,C
CB72	BIT 6,D
CB73	BIT 6,E
CB74	BIT 6,H
CB75	BIT 6,L
CB7E	BIT 7,(HL)
DDCB057E	BIT 7,(IX+d)
FDCB057E	BIT 7,(IY+d)
CB7F	BIT 7,A
CB78	BIT 7,B
CB79	BIT 7,C
CB7A	BIT 7,D
CB7B	BIT 7,E
CB7C	BIT 7,H
CB7D	BIT 7,L
DC8405	CALL C,NN
FC8405	CALL M,NN
D48405	CALL NC,NN
CD8405	CALL NN
C48405	CALL NZ,NN
F48405	CALL P,NN
EC8405	CALL PE,NN
E48405	CALL PO,NN
CC8405	CALL Z,NN
3F	CCF
BE	CP (HL)
DDBE05	CP (IX+d)
FDBE05	CP (IY+d)
BF	CP A
B8	CP B
B9	CP C
BA	CP D
BB	CP E
BC	CP H
BD	CP L
FE20	CP N
EDA9	CPD
EDB9	CPDR
EDA1	CPI
EDB1	CPIR
2F	CPL
27	DAA
35	DEC (HL)
DD3505	DEC (IX+d)
FD3505	DEC (IY+d)
3D	DEC A
05	DEC B
0B	DEC BC
0D	DEC C
15	DEC D
1B	DEC DE
1D	DEC E
25	DEC H
2B	DEC HL
DD2B	DEC IX
FD2B	DEC IY
2D	DEC L
3B	DEC SP
F3	DI
102E	DJNZ DIS
FB	EI
E3	EX (SP),HL
DDE3	EX (SP),IX
FDE3	EX (SP),IY
08	EX AF,AF
EB	EX DE,HL
D9	EXX
76	HALT
ED46	IM 0
ED56	IM 1
ED5E	IM 2
ED78	IN A,(C)
DB20	IN A,(N)
ED40	IN B,(C)
ED48	IN C,(C)
ED50	IN D,(C)
ED58	IN E,(C)
ED60	IN H,(C)
ED68	IN L,(C)
34	INC (HL)
DD3405	INC (IX+d)
FD3405	INC (IY+d)
3C	INC A
04	INC B
03	INC BC
0C	INC C
14	INC D
13	INC DE
1C	INC E
24	INC H
23	INC HL
DD23	INC IX
FD23	INC IY
2C	INC L
33	INC SP
EDAA	IND
EDBA	INDR
EDA2	INI
EDB2	INIR
E9	JP (HL)
DDE9	JP (IX)
FDE9	JP (IY)
DA8405	JP C,NN
FA8405	JP M,NN
D28405	JP NC,NN
C38405	JP NN
C28405	JP NZ,NN
F28405	JP P,NN
EA8405	JP PE,NN
E28405	JP PO,NN
CA8405	JP Z,NN
382E	JR C,DIS
182E	JR DIS
302E	JR NC,DIS
202E	JR NZ,DIS
282F	JR Z,DIS
02	LD (BC),A
12	LD (DE),A
77	LD (HL),A
70	LD (HL),B
71	LD (HL),C
72	LD (HL),D
73	LD (HL),E
74	LD (HL),H
75	LD (HL),L
3620	LD (HL),N
DD7705	LD (IX+d),A
DD7005	LD (IX+d),B
DD7105	LD (IX+d),C
DD7205	LD (IX+d),D
DD7305	LD (IX+d),E
DD7405	LD (IX+d),H
DD7505	LD (IX+d),L
DD360520	LD (IX+d),N
FD7705	LD (IY+d),A
FD7005	LD (IY+d),B
FD7105	LD (IY+d),C
FD7205	LD (IY+d),D
FD7305	LD (IY+d),E
FD7405	LD (IY+d),H
FD7505	LD (IY+d),L
FD360520	LD (IY+d),N
328405	LD (NN),A
ED438405	LD (NN),BC
ED538405	LD (NN),DE
228405	LD (NN),HL
DD228405	LD (NN),IX
FD228405	LD (NN),IY
ED738405	LD (NN),SP
0A	LD A,(BC)
1A	LD A,(DE)
7E	LD A,(HL)
DD7E05	LD A,(IX+d)
FD7E05	LD A,(IY+d)
3A8405	LD A,(NN)
7F	LD A,A
78	LD A,B
79	LD A,C
7A	LD A,D
7B	LD A,E
7C	LD A,H
ED57	LD A,I
7D	LD A,L
3E20	LD A,N
46	LD B,(HL)
DD4605	LD B,(IX+d)
FD4605	LD B,(IY+d)
47	LD B,A
40	LD B,B
41	LD B,C
42	LD B,D
43	LD B,E
44	LD B,H
45	LD B,L
0620	LD B,N
ED4B8405	LD BC,(NN)
018405	LD BC,NN
4E	LD C,(HL)
DD4E05	LD C,(IX+d)
FD4E05	LD C,(IY+d)
4F	LD C,A
48	LD C,B
49	LD C,C
4A	LD C,D
4B	LD C,E
4C	LD C,H
4D	LD C,L
0E20	LD C,N
56	LD D,(HL)
DD5605	LD D,(IX+d)
FD5605	LD D,(IY+d)
57	LD D,A
50	LD D,B
51	LD D,C
52	LD D,D
53	LD D,E
54	LD D,H
55	LD D,L
1620	LD D,N
ED5B8405	LD DE,(NN)
118405	LD DE,NN
5E	LD E,(HL)
DD5E05	LD E,(IX+d)
FD5E05	LD E,(IY+d)
5F	LD E,A
58	LD E,B
59	LD E,C
5A	LD E,D
5B	LD E,E
5C	LD E,H
5D	LD E,L
1E20	LD E,N
66	LD H,(HL)
DD6605	LD H,(IX+d)
FD6605	LD H,(IY+d)
67	LD H,A
60	LD H,B
61	LD H,C
62	LD H,D
63	LD H,E
64	LD H,H
65	LD H,L
2620	LD H,N
ED47	LD I,A
2A8405	LD HL,(NN)
218405	LD HL,NN
ED4F	LD R,A
F9	LD SP,HL
DDF9	LD SP,IX
FDF9	LD SP,IY
ED7B8405	LD SP,(NN)
318405	LD SP,NN
DD6605	LD IX,(IX+d)
FD6605	LD IX,(IY+d)
DD2A8405	LD IX,(NN)
DD218405	LD IX,NN
DD7705	LD IX+d),A
DD7005	LD IX+d),B
DD7105	LD IX+d),C
DD7205	LD IX+d),D
DD7305	LD IX+d),E
DD7405	LD IX+d),H
DD7505	LD IX+d),L
DD360520	LD IX+d),N
FD6605	LD IY,(IX+d)
FD6605	LD IY,(IY+d)
FD2A8405	LD IY,(NN)
FD218405	LD IY,NN
FD7705	LD IY+d),A
FD7005	LD IY+d),B
FD7105	LD IY+d),C
FD7205	LD IY+d),D
FD7305	LD IY+d),E
FD7405	LD IY+d),H
FD7505	LD IY+d),L
FD360520	LD IY+d),N
ED44	NEG
00	NOP
B6	OR (HL)
DDB605	OR (IX+d)
FDB605	OR (IY+d)
B7	OR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
F620	OR N
EDBB	OTDR
EDB3	OTIR
D320	OUT (N),A
ED79	OUT (C),A
ED41	OUT (C),B
ED49	OUT (C),C
ED51	OUT (C),D
ED59	OUT (C),E
ED61	OUT (C),H
ED69	OUT (C),L
EDAB	OUTD
EDA3	OUTI
F1	POP AF
C1	POP BC
D1	POP DE
E1	POP HL
DDE1	POP IX
FDE1	POP IY
F5	PUSH AF
C5	PUSH BC
D5	PUSH DE
E5	PUSH HL
DDE5	PUSH IX
FDE5	PUSH IY
CB86	RES 0,(HL)
DDCB0586	RES 0,(IX+d)
FDCB0586	RES 0,(IY+d)
CB87	RES 0,A
CB80	RES 0,B
CB81	RES 0,C
CB82	RES 0,D
CB83	RES 0,E
CB84	RES 0,H
CB85	RES 0,L
CB8E	RES 1,(HL)
DDCB058E	RES 1,(IX+d)
FDCB058E	RES 1,(IY+d)
CB8F	RES 1,A
CB88	RES 1,B
CB89	RES 1,C
CB8A	RES 1,D
CB8B	RES 1,E
CB8C	RES 1,H
CB8D	RES 1,L
CB96	RES 2,(HL)
DDCB0596	RES 2,(IX+d)
FDCB0596	RES 2,(IY+d)
CB97	RES 2,A
CB90	RES 2,B
CB91	RES 2,C
CB92	RES 2,D
CB93	RES 2,E
CB94	RES 2,H
CB95	RES 2,L
CB9E	RES 3,(HL)
DDCB059E	RES 3,(IX+d)
FDCB059E	RES 3,(IY+d)
CB9F	RES 3,A
CB98	RES 3,B
CB99	RES 3,C
CB9A	RES 3,D
CB9B	RES 3,E
CB9C	RES 3,H
CB9D	RES 3,L
CBA6	RES 4,(HL)
DDCB05A6	RES 4,(IX+d)
FDCB05A6	RES 4,(IY+d)
CBA7	RES 4,A
CBA0	RES 4,B
CBA1	RES 4,C
CBA2	RES 4,D
CBA3	RES 4,E
CBA4	RES 4,H
CBA5	RES 4,L
CBAE	RES 5,(HL)
DDCB05AE	RES 5,(IX+d)
FDCB05AE	RES 5,(IY+d)
CBAF	RES 5,A
CBA8	RES 5,B
CBA9	RES 5,C
CBAA	RES 5,D
CBAB	RES 5,E
CBAC	RES 5,H
CBAD	RES 5,L
CBB6	RES 6,(HL)
DDCB05B6	RES 6,(IX+d)
FDCB05B6	RES 6,(IY+d)
CBB7	RES 6,A
CBB0	RES 6,B
CBB1	RES 6,C
CBB2	RES 6,D
CBB3	RES 6,E
CBB4	RES 6,H
CBB5	RES 6,L
CBBE	RES 7,(HL)
DDCB05BE	RES 7,(IX+d)
FDCB05BE	RES 7,(IY+d)
CBBF	RES 7,A
CBB8	RES 7,B
CBB9	RES 7,C
CBBA	RES 7,D
CBBB	RES 7,E
CBBC	RES 7,H
CBBD	RES 7,L
C9	RET
D8	RET C
F8	RET M
D0	RET NC
C0	RET NZ
F0	RET P
E8	RET PE
E0	RET PO
C8	RET Z
ED4D	RETI
ED45	RETN
CB16	RL (HL)
DDCB0516	RL (IX+d)
FDCB0516	RL (IY+d)
CB17	RL A
CB10	RL B
CB11	RL C
CB12	RL D
CB13	RL E
CB14	RL H
CB15	RL L
17	RLA
CB06	RLC (HL)
DDCB0506	RLC (IX+d)
FDCB0506	RLC (IY+d)
CB07	RLC A
CB00	RLC B
CB01	RLC C
CB02	RLC D
CB03	RLC E
CB04	RLC H
CB05	RLC L
07	RLCA
ED6F	RLD
CB1E	RR (HL)
DDCB051E	RR (IX+d)
FDCB051E	RR (IY+d)
CB1F	RR A
CB18	RR B
CB19	RR C
CB1A	RR D
CB1B	RR E
CB1C	RR H
CB1D	RR L
1F	RRA
CB0E	RRC (HL)
DDCB050E	RRC (IX+d)
FDCB050E	RRC (IY+d)
CB0F	RRC A
CB08	RRC B
CB09	RRC C
CB0A	RRC D
CB0B	RRC E
CB0C	RRC H
CB0D	RRC L
0F	RRCA
ED67	RRD
C7	RST 0
CF	RST 8
D7	RST 10
DF	RST 18
E7	RST 20
EF	RST 28
F7	RST 30
FF	RST 38
DE20	SBC A,N
9E	SBC A,(HL)
DD9E05	SBC A,(IX+d)
FD9E05	SBC A,(IY+d)
9F	SBC A,A
98	SBC A,B
99	SBC A,C
9A	SBC A,D
9B	SBC A,E
9C	SBC A,H
9D	SBC A,L
ED42	SBC HL,BC
ED52	SBC HL,DE
ED62	SBC HL,HL
ED72	SBC HL,SP
CB3E	SRL (HL)
DDCB053E	SRL (IX+d)
FDCB053E	SRL (IY+d)
CB3F	SRL A
CB38	SRL B
CB39	SRL C
CB3A	SRL D
CB3B	SRL E
CB3C	SRL H
CB3D	SRL L
D620	SUB N
96	SUB (HL)
DD9605	SUB (IX+d)
FD9605	SUB (IY+d)
97	SUB A
90	SUB B
91	SUB C
92	SUB D
93	SUB E
94	SUB H
95	SUB L
AE	XOR (HL)
DDAE05	XOR (IX+d)
FDAE05	XOR (IY+d)
AF	XOR A
A8	XOR B
A9	XOR C
AA	XOR D
AB	XOR E
AC	XOR H
AD	XOR L
EE20	XOR N

OBJ CODE	SOURCE STATEMENT
49	LD C,C
0E20	LD C,N
4E	LD C,(HL)
DD5605	LD D,(IX+d)
FD5605	LD D,(IY+d)
57	LD D,A
50	LD D,B
51	LD D,C
52	LD D,D
53	LD D,E
54	LD D,H
55	LD D,L
1620	LD D,N
ED5B8405	LD DE,(NN)
118405	LD DE,NN
5E	LD E,(HL)
DD5E05	LD E,(IX+d)
FD5E05	LD E,(IY+d)
5F	LD E,A
58	LD E,B
59	LD E,C
5A	LD E,D
5B	LD E,E
5C	LD E,H
5D	LD E,L
1E20	LD E,N
66	LD H,(HL)
DD6605	LD H,(IX+d)
FD6605	LD H,(IY+d)
67	LD H,A
60	LD H,B
61	LD H,C
62	LD H,D
63	LD H,E
64	LD H,H
65	LD H,L
2620	LD H,N
2A8405	LD HL,(NN)
218405	LD HL,NN
E047	LD I,A
DD2A8405	LD IX,(NN)
DD218405	LD IX,NN
FD2A8405	LD IY,(NN)
FD218405	LD IY,NN
6E	LD L,(HL)
DD6E05	LD L,(IX+d)
FD6E05	LD L,(IY+d)
6F	LD L,A
68	LD L,B
69	LD L,C
6A	LD L,D
6B	LD L,E
6C	LD L,H
6D	LD L,L
2E20	LD L,N
ED7B8405	LD SP,(NN)
F9	LD SP,HL
DDF9	LD SP,IX
FDF9	LD SP,IY
318405	LD SP,NN
EDA8	LDD
EDB8	LDDR
EDA0	LDI
EDB0	LDIR
ED44	NEG
00	NOP
B6	OR (HL)
DDB605	OR (IX+d)
FDB605	OR (IY+d)
B7	OR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
F620	OR N
EDB8	OTDR
EDB3	OTIR
ED79	OUT (C),A
ED41	OUT (C),B
ED49	OUT (C),C
ED51	OUT (C),D
ED59	OUT (C),E
ED61	OUT (C),H
ED69	OUT (C),L
D320	OUT (N),A
EDAB	OUTD
EDA3	OUTI
F1	POP AF
C1	POP BC
D1	POP DE
E1	POP HL
DDE1	POP IX
FDE1	POP IY
F5	PUSH AF
C5	PUSH BC
D5	PUSH DE
E5	PUSH HL
DDE5	PUSH IX
FDE5	PUSH IY
CB86	RES 0,(HL)
DDCB0586	RES 0,(IX+d)
FDCB0586	RES 0,(IY+d)
CB87	RES 0,A
CB80	RES 0,B
CB81	RES 0,C
CB82	RES 0,D
CB83	RES 0,E
CB84	RES 0,H
CB85	RES 0,L
CB8E	RES 1,(HL)
DDCB058E	RES 1,(IX+d)
FDCB058E	RES 1,(IY+d)
CB8F	RES 1,A
CB88	RES 1,B
CB89	RES 1,C
CB8A	RES 1,D
CB8B	RES 1,E
CB8C	RES 1,H
CB8D	RES 1,L
CB96	RES 2,(HL)
DDCB0596	RES 2,(IX+d)
FDCB0596	RES 2,(IY+d)
CB97	RES 2,A
CB90	RES 2,B
CB91	RES 2,C
CB92	RES 2,D
CB93	RES 2,E
CB94	RES 2,H
CB95	RES 2,L
CB9E	RES 3,(HL)
DDCB059E	RES 3,(IX+d)
FDCB059E	RES 3,(IY+d)
CB9F	RES 3,A
CB98	RES 3,B
CB99	RES 3,C
CB9A	RES 3,D
CB9B	RES 3,E
CB9C	RES 3,H
CB9D	RES 3,L
CBA6	RES 4,(HL)
DDCB05A6	RES 4,(IX+d)
FDCB05A6	RES 4,(IY+d)
CBA7	RES 4,A
CBA0	RES 4,B
CBA1	RES 4,C
CBA2	RES 4,D
CBA3	RES 4,E
CBA4	RES 4,H
CBA5	RES 4,L
CBAE	RES 5,(HL)
DDCB05AE	RES 5,(IX+d)
FDCB05AE	RES 5,(IY+d)
CBAF	RES 5,A
CBA8	RES 5,B
CBA9	RES 5,C
CBAA	RES 5,D
CBAB	RES 5,E
CBAC	RES 5,H
CBAD	RES 5,L
CBB6	RES 6,(HL)
DDCB05B6	RES 6,(IX+d)
FDCB05B6	RES 6,(IY+d)
CBB7	RES 6,A
CBB0	RES 6,B
CBB1	RES 6,C
CBB2	RES 6,D
CBB3	RES 6,E
CBB4	RES 6,H
CBB5	RES 6,L
CBBE	RES 7,(HL)
DDCB05BE	RES 7,(IX+d)
FDCB05BE	RES 7,(IY+d)
CBBF	RES 7,A
CBB8	RES 7,B
CBB9	RES 7,C
CBBA	RES 7,D
CBBB	RES 7,E
CBBC	RES 7,H
CBBD	RES 7,L
C9	RET
D8	RET C
F8	RET M
D0	RET NC
C0	RET NZ
F0	RET P
E8	RET PE
E0	RET PO
C8	RET Z
ED4D	RETI
ED45	RETN
CB16	RL (HL)
DDCB0516	RL (IX+d)
FDCB0516	RL (IY+d)
CB17	RL A
CB10	RL B
CB11	RL C
CB12	RL D
CB13	RL E

OBJ CODE	SOURCE STATEMENT
CB14	RL H
CB15	RL L
17	RLA
CB06	RLC (HL)
DDCB0506	RLC (IX+d)
FDCB0506	RLC (IY+d)
CB07	RLC A
CB00	RLC B
CB01	RLC C
CB02	RLC D
CB03	RLC E
CB04	RLC H
CB05	RLC L
07	RLCA
ED6F	RLD
CB1E	RR (HL)
DDCB051E	RR (IX+d)
FDCB051E	RR (IY+d)
CB1F	RR A
CB18	RR B
CB19	RR C
CB1A	RR D
CB1B	RR E
CB1C	RR H
CB1D	RR L
1F	RRA
CB0E	RRC (HL)
DDCB050E	RRC (IX+d)
FDCB050E	RRC (IY+d)
CB0F	RRC A
CB08	RRC B
CB09	RRC C
CB0A	RRC D
CB0B	RRC E
CB0C	RRC H
CB0D	RRC L
0F	RRCA
ED67	RRD
C7	RST 0
D7	RST 10H
DF	RST 18H
E7	RST 20H
EF	RST 28H
F7	RST 30H
FF	RST 38H
CF	RST 8
9E	SBC A,(HL)
DD9E05	SBC A,(IX+d)
FD9E05	SBC A,(IY+d)
9F	SBC A,A
98	SBC A,B
99	SBC A,C
9A	SBC A,D
9B	SBC A,E
9C	SBC A,H
9D	SBC A,L
DE20	SBC A,N
ED42	SBC HL,BC
ED52	SBC HL,DE
ED62	SBC HL,HL
ED72	SBC HL,SP
37	SCF
CBC6	SET 0,(HL)
DDCB05C6	SET 0,(IX+d)
FDCB05C6	SET 0,(IY+d)
C9C7	SET 0,A
CBC0	SET 0,B
CBC1	SET 0,C
CBC2	SET 0,D
CBC3	SET 0,E
CBC4	SET 0,H
CBC5	SET 0,L
CBCE	SET 1,(HL)
DDCB05CE	SET 1,(IX+d)
FDCB05CE	SET 1,(IY+d)
CBCF	SET 1,A
CBC8	SET 1,B
CBC9	SET 1,C
CBCA	SET 1,D
CBCB	SET 1,E
CBCC	SET 1,H
CBCD	SET 1,L
CBD6	SET 2,(HL)
DDCB05D6	SET 2,(IX+d)
FDCB05D6	SET 2,(IY+d)
CBD7	SET 2,A
CBD0	SET 2,B
CBD1	SET 2,C
CBD2	SET 2,D
CBD3	SET 2,E
CBD4	SET 2,H
CBD5	SET 2,L
CBD8	SET 3,B
CBDE	SET 3,(HL)
DDCB05DE	SET 3,(IX+d)
FDCB05DE	SET 3,(IY+d)
CBDF	SET 3,A
CBD9	SET 3,C
CBDA	SET 3,D
CBDB	SET 3,E
CBDC	SET 3,H
CBDD	SET 3,L
CBE6	SET 4,(HL)
DDCB05E6	SET 4,(IX+d)
FDCB05E6	SET 4,(IY+d)
CBE7	SET 4,A
CBE0	SET 4,B
CBE1	SET 4,C
CBE2	SET 4,D
CBE3	SET 4,E
CBE4	SET 4,H
CBE5	SET 4,L
CBEE	SET 5,(HL)
DDCB05EE	SET 5,(IX+d)
FDCB05EE	SET 5,(IY+d)
CBEF	SET 5,A
CBE8	SET 5,B
CBE9	SET 5,C
CBEA	SET 5,D
CBEB	SET 5,E
CBEC	SET 5,H
CBED	SET 5,L
CBF6	SET 6,(HL)
DDCB05F6	SET 6,(IX+d)
FDCB05F6	SET 6,(IY+d)
CBF7	SET 6,A
CBF0	SET 6,B
CBF1	SET 6,C
CBF2	SET 6,D
CBF3	SET 6,E
CBF4	SET 6,H
CBF5	SET 6,L

OBJ CODE	SOURCE STATEMENT
CBFE	SET 7,(HL)
DDCB05FE	SET 7,(IX+d)
FDCB05FE	SET 7,(IY+d)
CBFF	SET 7,A
CBF8	SET 7,B
CBF9	SET 7,C
CBFA	SET 7,D
CBFB	SET 7,E
CBFC	SET 7,H
CBFD	SET 7,L
CB26	SLA (HL)
DDCB0526	SLA (IX+d)
FDCB0526	SLA (IY+d)
CB27	SLA A
CB20	SLA B
CB21	SLA C
CB22	SLA D
CB23	SLA E
CB24	SLA H
CB25	SLA L
CB2E	SRA (HL)
DDCB052E	SRA (IX+d)
FDCB052E	SRA (IY+d)
CB2F	SRA A
CB28	SRA B
CB29	SRA C
CB2A	SRA D
CB2B	SRA E
CB2C	SRA H
CB2D	SRA L
CB3E	SRL (HL)
DDCB053E	SRL (IX+d)
FDCB053E	SRL (IY+d)
CB3F	SRL A
CB38	SRL B
CB39	SRL C
CB3A	SRL D
CB3B	SRL E
CB3C	SRL H
CB3D	SRL L
96	SUB (HL)
DD9605	SUB (IX+d)
FD9605	SUB (IY+d)
97	SUB A
90	SUB B
91	SUB C
92	SUB D
93	SUB E
94	SUB H
95	SUB L
D620	SUB N
AE	XOR (HL)
DDAE05	XOR (IX+d)
FDAE05	XOR (IY+d)
AF	XOR A
A8	XOR B
A9	XOR C
AA	XOR D
AB	XOR E
AC	XOR H
AD	XOR L
EE20	XOR N

IV.4 INSTRUKČNÍ SOUBOR Z80 (OP-CODE)

Seznam instrukcí Z80-CPU seřazených podle vzrůstajícího operačního kódu (hex)

OBJ CODE	SOURCE STATEMENT	OBJ CODE	SOURCE STATEMENT	OBJ CODE	SOURCE STATEMENT
00	NOP	218405	LD HL, NN	42	LD B, D
018405	LD BC, NN	228405	LD (NN), HL	43	LD B, E
02	LD (BC), A	23	INC HL	44	LD B, H, NN
03	INC BC	24	INC H	45	LD B, L
04	INC B	25	DEC H	46	LD B, (HL)
05	DEC B	2620	LD H, N	47	LD B, A
0620	LD B, N	27	DAA	48	LD C, B
07	RLCA	282E	JR Z, DIS	49	LD C, C
08	EX AF, AF'	29	ADD HL, HL	4A	LD C, D
09	ADD HL, BC	2A8405	LD (HL), (NN)	4B	LD C, E
0A	LD A,(BC)	2B	DEC HL	4C	LD C, H
0B	DEC BC	2C	INC L	4D	LD C, L
0C	INC C	2D	DEC L	4E	LD C, (HL)
0D	DEC C	2E20	LD L, N	4F	LD C, A
0E20	LD C, N	2F	CPL	50	LD D, B
0F	RRCA	302E	JR NC, DIS	51	LD D, C
102E	DJNZ DIS	318405	LD SP, NN	52	LD D, D
118405	LD DE, NN	328405	LD (NN), A	53	LD D, E
12	LD (DE), A	33	INC SP	54	LD D, H
13	INC DE	34	INC (HL)	55	LD D, L
14	INC D	35	DEC (HL)	56	LD D, (HL)
15	DEC D	3620	LD (HL), N	57	LD D, A
1620	LD D, N	37	SCF	58	LD E, B
17	RLA	382E	JR C, DIS	59	LD E, C
182E	JR DIS	39	ADD HL, SP	5A	LD E, D
19	ADD HL, DE	3A8405	LD A, (NN)	5B	LD E, E
1A	LD A, (DE)	3B	DEC SP	5C	LD E, H
1B	DEC DE	3C	INC A	5D	LD E, L
1C	INC E	3D	DEC A	5E	LD E, (HL)
1D	DEC E	3E20	LD A, N	5F	LD E, A
1E20	LD E, N	3F	CCF	60	LD H, B
1F	RRA	40	LD B, B	61	LD H, C
202E	JR NZ, DIS	41	LD B, C	62	LD H, D

OBJ CODE	SOURCE STATEMENT	OBJ CODE	SOURCE STATEMENT	OBJ CODE	SOURCE STATEMENT
63	LD H, E	A5	AND L	E9	JP (HL)
64	LD H, H	A6	AND (HL)	EA8405	JP PE, NN
65	LD H, L	A7	AND A	EB	EX DE, HL
66	LD H, (HL)	A8	XOR B	EC8405	CALL PE, NN
67	LD H, A	A9	XOR C	EE20	XOR N
68	LD L, B	AA	XOR D	EF	RST 28H
69	LD L, C	AB	XOR E	F0	RET P
6A	LD L, D	AC	XOR H	F1	POP AF
6B	LD L, E	AD	XOR L	F28405	JP P, NN
6C	LD L, H	AE	XOR (HL)	F3	DI
6D	LD L, L	AF	XOR A	F48405	CALL P, NN
6E	LD L, (HL)	B0	OR B	F5	PUSH AF
6F	LD L, A	B1	OR C	F620	OR N
70	LD (HL), B	B2	OR D	F7	RST 30H
71	LD (HL), C	B3	OR E	F8	RET M
72	LD (HL), D	B4	OR H	F9	LD SP, HL
73	LD (HL), E	B5	OR L	FA8405	JP M, NN
74	LD (HL), H	B6	OR (HL)	FB	EI
75	LD (HL), L	B7	OR A	FC8405	CALL M, NN
76	HALT	B8	CP B	FE20	CP N
77	LD (HL), A	B9	CP C	FF	RST 38H
78	LD A, B	BA	CP D	CB00	RLC B
79	LD A, C	BB	CP E	CB01	RLC C
7A	LD A, D	BC	CP H	CB02	RLC D
7B	LD A, E	BD	CP L	CB03	RLC E
7C	LD A, H	BE	CP (HL)	CB04	RLC H
7D	LD A, L	BF	CP A	CB05	RLC L
7E	LD A, (HL)	C0	RET NZ	CB06	RLC (HL)
7F	LD A, A	C1	POP BC	CB07	RLC A
80	ADD A, B	C28405	JP NZ, NN	CB08	RRC B
81	ADD A, C	C38405	JP NN	CB09	RRC C
82	ADD A, D	C48405	CALL NZ, NN	CB0A	RRC D
83	ADD A, E	C5	PUSH BC	CB0B	RRC E
84	ADD A, H	C620	ADD A, N	CB0C	RRC H
85	ADD A, L	C7	RST 0	CB0D	RRC L
86	ADD A, (HL)	C8	RET Z	CB0E	RRC (HL)
87	ADD A, A	C9	RET	CB0F	RRC A
88	ADC A, B	CA8405	JP Z, NN	CB10	RL B
89	ADC A, C	CC8405	CALL Z, NN	CB11	RL C
8A	ADC A, D	CD8405	CALL NN	CB12	RL D
8B	ADC A, E	CE20	ADC A, N	CB13	RL E
8C	ADC A, H	CF	RST 8	CB14	RL H
8D	ADC A, L	D0	RET NC	CB15	RL L
8E	ADC A, (HL)	D1	POP DE	CB16	RL (HL)
8F	ADC A, A	D28405	JP NC, NN	CB17	RL A
90	SUB B	D320	OUT (N), A	CB18	RR B
91	SUB C	D48405	CALL NC, NN	CB19	RR C
92	SUB D	D5	PUSH DE	CB1A	RR D
93	SUB E	D620	SUB N	CB1B	RR E
94	SUB H	D7	RST 10H	CB1C	RR H
95	SUB L	D8	RET C	CB1D	RR L
96	SUB (HL)	D9	EXX	CB1E	RR (HL)
97	SUB A	DA8405	JP C, NN	CB1F	RR A
98	SBC A, B	DB20	IN A, (N)	CB20	SLA B
99	SBC A, C	DC8405	CALL C, N	CB21	SLA C
9A	SBC A, D	DE20	SBC A, N	CB22	SLA D
9B	SBC A, E	DF	RST 18H	CB23	SLA E
9C	SBC A, H	E0	RET PO	CB24	SLA H
9D	SBC A, L	E1	POP HL	CB25	SLA L
9E	SBC A, (HL	E28405	JP PO, NN	CB26	SLA (HL)
9F	SBC A, A	E3	EX (SP), HL	CB27	SLA A
A0	AND B	E48405	CALL PO, NN	CB28	SRA B
A1	AND C	E5	PUSH HL	CB29	SRA C
A2	AND D	E620	AND N	CB2A	SRA D
A3	AND E	E7	RST 20 H	CB2B	SRA E
A4	AND H	E8	RET PE	CB2C	SRA H

OBJ CODE	SOURCE STATEMENT
CB2D	SRA L
CB2E	SRA (HL)
CB2F	SRA A
CB38	SRL B
CB39	SRL C
CB3A	SRL D
CB3B	SRL E
CB3C	SRL H
CB3D	SRL L
CB3E	SRL (HL)
CB3F	SRL A
CB40	BIT 0, B
CB41	BIT 0, C
CB42	BIT 0, D
CB43	BIT 0, E
CB44	BIT 0, H
CB45	BIT 0, L
CB46	BIT 0, (HL)
CB47	BIT 0, A
CB48	BIT 1, B
CB49	BIT 1, C
CB4A	BIT 1, D
CB4B	BIT 1, E
CB4C	BIT 1, H
CB4D	BIT 1, L
CB4E	BIT 1, (HL)
CB4F	BIT 1, A
CB50	BIT 2, B
CB51	BIT 2, C
CB52	BIT 2, D
CB53	BIT 2, E
CB54	BIT 2, H
CB55	BIT 2, L
CB56	BIT 2, (HL)
CB57	BIT 2, A
CB58	BIT 3, B
CB59	BIT 3, C
CB5A	BIT 3, D
CB5B	BIT 3, E
CB5C	BIT 3, H
CB5D	BIT 3, L
CB5E	BIT 3, (HL)
CB5F	BIT 3, A
CB60	BIT 4, B
CB61	BIT 4, C
CB62	BIT 4, D
CB63	BIT 4, E
CB64	BIT 4, H
CB65	BIT 4, L
CB66	BIT 4, (HL)
CB67	BIT 4, A
CB68	BIT 5, B
CB69	BIT 5, C
CB6A	BIT 5, D
CB6B	BIT 5, E
CB6C	BIT 5, H
CB6D	BIT 5, L
CB6E	BIT 5, (HL)
CB6F	BIT 5, A
CB70	BIT 6, B
CB71	BIT 6, C
CB72	BIT 6, D
CB73	BIT 6, E
CB74	BIT 6, H
CB75	BIT 6, L
CB76	BIT 6, (HL)
CB77	BIT 6, A
CB78	BIT 7, B
CB79	BIT 7, C
CB7A	BIT 7, D
CB7B	BIT 7, E
CB7C	BIT 7, H
CB7D	BIT 7, L
CB7E	BIT 7, (HL)
CB7F	BIT 7, A
CB80	RES 0, B
CB81	RES 0, C
CB82	RES 0, D
CB83	RES 0, E
CB84	RES 0, H
CB85	RES 0, L
CB86	RES 0, (HL)
CB87	RES 0, A
CB88	RES 1, B
CB89	RES 1, C
CB8A	RES 1, D
CB8B	RES 1, E
CB8C	RES 1, H
CB8D	RES 1, L
CB8E	RES 1, (HL)
CB8F	RES 1, A
CB90	RES 2, B
CB91	RES 2, C
CB92	RES 2, D
CB93	RES 2, E
CB94	RES 2, H
CB95	RES 2, L
CB96	RES 2, (HL)
CB97	RES 2, A
CB98	RES 3, B
CB99	RES 3, C
CB9A	RES 3, D
CB9B	RES 3, E
CB9C	RES 3, H
CB9D	RES 3, L
CB9E	RES 3, (HL)
CB9F	RES 3, A
CBA0	RES 4, B
CBA1	RES 4, C
CBA2	RES 4, D
CBA3	RES 4, E
CBA4	RES 4, H
CBA5	RES 4, L
CBA6	RES 4, (HL)
CBA7	RES 4, A
CBA8	RES 5, B
CBA9	RES 5, C
CBAA	RES 5, D
CBAB	RES 5, E
CBAC	RES 5, H
CBAD	RES 5, L
CBAE	RES 5, (HL)
CBAF	RES 5, A
CBB0	RES 6, B
CBB1	RES 6, C
CBB2	RES 6, D
CBB3	RES 6, E
CBB4	RES 6, H
CBB5	RES 6, L
CBB6	RES 6, (HL)
CBB7	RES 6, A
CBB8	RES 7, B
CBB9	RES 7, C
CBBA	RES 7, D
CBBB	RES 7, E
CBBC	RES 7, H
CBBD	RES 7, L
CBBE	RES 7, (HL)
CBBF	RES 7, A
CBC0	SET 0, B
CBC1	SET 0, C
CBC2	SET 0, D
CBC3	SET 0, E
CBC4	SET 0, H
CBC5	SET 0, L
CBC6	SET 0, (HL)
CBC7	SET 0, A
CBC8	SET 1, B
CBC9	SET 1, C
CBCA	SET 1, D
CBCB	SET 1, E
CBCC	SET 1, H
CBCD	SET 1, L
CBCE	SET 1, (HL)
CBCF	SET 1, A
CBD0	SET 2, B
CBD1	SET 2, C
CBD2	SET 2, D
CBD3	SET 2, E
CBD4	SET 2, H
CBD5	SET 2, L
CBD6	SET 2, (HL)
CBD7	SET 2, A
CBD8	SET 3, B
CBD9	SET 3, C
CBDA	SET 3, D
CBDB	SET 3, E
CBDC	SET 3, H
CBDD	SET 3, L
CBDE	SET 3, (HL)
CBDF	SET 3, A
CBE0	SET 4, B
CBE1	SET 4, C
CBE2	SET 4, D
CBE3	SET 4, E
CBE4	SET 4, H
CBE5	SET 4, L
CBE6	SET 4, (HL)
CBE7	SET 4, A
CBE8	SET 5, B
CBE9	SET 5, C
CBEA	SET 5, D
CBEB	SET 5, E
CBEC	SET 5, H
CBED	SET 5, L
CBEE	SET 5, (HL)
CBEF	SET 5, A
CBF0	SET 6, B
CBF1	SET 6, C
CBF2	SET 6, D
CBF3	SET 6, E
CBF4	SET 6, H
CBF5	SET 6, L
CBF6	SET 6, (HL)
CBF7	SET 6, A
CBF8	SET 7, B
CBF9	SET 7, C
CBFA	SET 7, D
CBFB	SET 7, E
CBFC	SET 7, H
CBFD	SET 7, L
CBFE	SET 7, (HL)
CBFF	SET 7, A
DD09	ADD IX, BC
DD19	ADD IX, DE
DD218405	LD IX, NN
DD228405	LD (NN), IX
DD23	INC IX
DD29	ADD IX, IX
DD2A8405	LD IX, (NN)
DD2B	DEC IX
DD3405	INC (IX + d)
DD3505	DEC (IX + d)
DD360520	LD (IX + d), N
DD39	ADD IX, SP
DD4605	LD B, (IX + d)
DD4E05	LD C, (IX + d)
DD5605	LD D, (IX + d)
DD5E05	LD E, (IX + d)
DD6605	LD H, (IX + d)
DD6E05	LD L, (IX + d)
DD7005	LD (IX + d), B
DD7105	LD (IX + d), C
DD7205	LD (IX + d), D
DD7305	LD (IX + d), E
DD7405	LD (IX + d), H
DD7505	LD (IX + d), L
DD7705	LD (IX + d), A
DD7E05	LD A, (IX + d)
DD8605	ADD A, (IX + d)
DD8E05	ADC A, (IX + d)
DD9605	SUB (IX + d)
DD9E05	SBC A, (IX + d)
DDA605	AND (IX + d)
DDAE05	XOR (IX + d)
DDB605	OR (IX + d)
DDBE05	CP (IX + d)
DDE1	POP IX
DDE3	EX (SP), IX
DDE5	PUSH IX
DDE9	JP (IX)
DDF9	LD SP, IX
DDCB0506	RLC (IX + d)
DDCB050E	RRC (IX + d)
DDCB0516	RL (IX + d)
DDCB051E	RR (IX + d)
DDCB0526	SLA (IX + d)
DDCB052E	SRA (IX + d)
DDCB053E	SRL (IX + d)
DDCB0546	BIT 0, (IX + d)
DDCB054E	BIT 1, (IX + d)
DDCB0556	BIT 2, (IX + d)
DDCB055E	BIT 3, (IX + d)
DDCB0566	BIT 4, (IX + d)
DDCB056E	BIT 5, (IX + d)
DDCB0576	BIT 6, (IX + d)
DDCB057E	BIT 7, (IX + d)
DDCB0586	RES 0, (IX + d)
DDCB058E	RES 1, (IX + d)
DDCB0596	RES 2, (IX + d)
DDCB059E	RES 3, (IX + d)
DDCB05A6	RES 4, (IX + d)
DDCB05AE	RES 5, (IX + d)
DDCB05B6	RES 6, (IX + d)
DDCB05BE	RES 7, (IX + d)
DDCB05C6	SET 0, (IX + d)
DDCB05CE	SET 1, (IX + d)
DDCB05D6	SET 2, (IX + d)
DDCB05DE	SET 3, (IX + d)
DDCB05E6	SET 4, (IX + d)
DDCB05EE	SET 5, (IX + d)
DDCB05F6	SET 6, (IX + d)
DDCB05FE	SET 7, (IX + d)
ED40	IN B, (C)
ED41	OUT (C), B
ED42	SBC HL, BC
ED438405	LD (NN), BC
ED44	NEG
ED45	RETN
ED46	IM 0
ED47	LD I, A
ED48	IN C, (C)
ED49	OUT (C), C
ED4A	ADC HL, BC
ED4B8405	LD BC, (NN)
ED4D	RETI
ED50	IN D, (C)
ED51	OUT (C), D
ED52	SBC HL, DE
ED538405	LD (NN), DE
ED56	IM 1
ED57	LD A, I
ED58	IN E, (C)
ED59	OUT (C), E
ED5A	ADC HL, DE
ED5B8405	LD DE, (NN)
ED5E	IM 2
ED60	IN H, (C)
ED61	OUT (C), H
ED62	SBC HL, HL
ED67	RRD
ED68	IN L, (C)
ED69	OUT (C), L
ED6A	ADC HL, HL
ED6F	RLD
ED72	SBC HL, SP
ED738405	LD (NN), SP
ED78	IN A, (C)
ED79	OUT (C), A
ED7A	ADC HL, SP
ED7B8405	LD SP, (NN)
EDA0	LDI
EDA1	CPI
EDA2	INI
EDA3	OUTI
EDA8	LDD
EDA9	CPD
EDAA	IND
EDAB	OUTD
EDB0	LDIR
EDB1	CPIR
EDB2	INIR
EDB3	OTIR
EDB8	LDDR
EDB9	CPDR
EDBA	INDR
EDBB	OTDR
FD09	ADD IY, BC
FD19	ADD IY, DE
FD218405	LD IY, NN
FD228405	LD (NN), IY
FD23	INC IY
FD29	ADD IY, IY
FD2A8405	LD IY, (NN)
FD2B	DEC IY
FD3405	INC (IY + d)
FD3505	DEC (IY + d)
FD360520	LD (IY + d), N
FD39	ADD IY, SP
FD4605	LD B, (IY + d)
FD4E05	LD C, (IY + d)
FD5605	LD D, (IY + d)
FD5E05	LD E, (IY + d)
FD6605	LD H, (IY + d)
FD6E05	LD L, (IY + d)
FD7005	LD (IY + d), B
FD7105	LD (IY + d), C
FD7205	LD (IY + d), D
FD7305	LD (IY + d), E
FD7405	LD (IY + d), H
FD7505	LD (IY + d), L
FD7705	LD (IY + d), A
FD7E05	LD A, (IY + d)
FD8605	ADD A, (IY + d)
FD8E05	ADC A, (IY + d)
FD9605	SUB (IY + d)
FD9E05	SBC A, (IY + d)
FDA605	AND (IY + d)
FDAE05	XOR (IY + d)
FDB605	OR (IY + d)
FDBE05	CP (IY + d)
FDE1	POP IY
FDE3	EX (SP), IY
FDE5	PUSH IY
FDE9	JP (IY)
FDF9	LD SP, IY
FDCB0506	RLC (IY + d)
FDCB050E	RRC (IY + d)
FDCB0516	RL (IY + d)
FDCB051E	RR (IY + d)
FDCB0526	SLA (IY + d)
FDCB052E	SRA (IY + d)
FDCB053E	SRL (IY + d)
FDCB0546	BIT 0, (IY + d)
FDCB054E	BIT 1, (IY + d)
FDCB0556	BIT 2, (IY + d)
FDCB055E	BIT 3, (IY + d)
FDCB0566	BIT 4, (IY + d)
FDCB056E	BIT 5, (IY + d)
FDCB0576	BIT 6, (IY + d)
FDCB057E	BIT 7, (IY + d)
FDCB0586	RES 0, (IY + d)
FDCB058E	RES 1, (IY + d)
FDCB0596	RES 2, (IY + d)
FDCB059E	RES 3, (IY + d)
FDCB05A6	RES 4, (IY + d)
FDCB05AE	RES 5, (IY + d)
FDCB05B6	RES 6, (IY + d)
FDCB05BE	RES 7, (IY + d)
FDCB05C6	SET 0, (IY + d)
FDCB05CE	SET 1, (IY + d)
FDCB05D6	SET 2, (IY + d)
FDCB05DE	SET 3, (IY + d)
FDCB05E6	SET 4, (IY + d)
FDCB05EE	SET 5, (IY + d)
FDCB05F6	SET 6, (IY + d)
FDCB05FE	SET 7, (IY + d)

V. Aplikace

V.1. Jednoduché jednodeskové mikropočítače

Na obr. 46 je zapojení velmi jednoduchého mikroprocesorového systému, určeného pro nesložité řídicí úlohy. Mimo vlastní mikroprocesor a nezbytný generátor taktu (zde stabilizovaný křemenným výbrusem) má minimální paměť programu (2758 s kapacitou 1 KB) a dat (2x 4045 s kapacitou 1 KB), dále pak dvě osmibitové vstupně-výstupní brány a nerozsáhlou logiku řídicích signálů /1/. Jako vstupní brána slouží oddělovací zesilovač typu 74 LS 244, jako výstupní osmibitový latch typu 74 LS 273. Tyto brány jsou spojeny přímo s řídicími signály $\overline{\text{IORD}}$ a $\overline{\text{IOWR}}$ bez vazby s nějakou adresou; to proto, že pro každý směr přenosu dat je zde jen jedna brána. Zbývající řídicí signály $\overline{\text{MEMRD}}$ a $\overline{\text{MEMWR}}$ pro výběr a uvolnění pamětí jsou získávány za kombinačními obvody 74 LS 32 a 74 LS 00. Posledně jmenovaný (1/2 74LS00) slouží též k automatickému nulování programového čítače CPJ ve spojení s rezistorem R_1 a kondenzátorem C_1, a to po připojení napájecího napětí +5 V.

Další ukázku jednodeskového mikroprocesorového systému, vhodného pro celou řadu úloh řídicího charakteru, přináší obr. 47. Je osazen celkem devíti integrovanými obvody na desce tzv. evropského formátu, přičemž na ní zbývá ještě dostatek místa pro eventuální použití uživatele /17/. Takovýto systém - stejně jako předcházející - není ovšem určen pro zkoušení, "ladění" či vývoj programů (i když principiálně při vhodném monitorovacím programu a odpovídajících periferiích by to v daném případě bylo možné), ale jako flexibilní náhrada zapojení "pevné logiky" či pro již zmíněné řízení. Systém tedy sestává z vlastní CPJ (IO5) a k ní příslušného generátoru taktu (IO8) s iniciačním obvodem (IO6), programové paměti IO_1 a paměti dat IO_2; dvou programovatelných paralelních stykových obvodů IO3 a IO4 jakož i nezbytných dekodérů-demultiplexorů IO7.

Jimi je m.j. dáno adresové umístění podpůrných obvodů mikroprocesoru, které je následující:
IO3 - data A - 00H, řízení 02H, data B - 01H, řízení 03H,
IO4 - data A - 10H, řízení 12H, data B - 11H, řízení 13H,
IO2 - RAM - 8000H,
IO1 - EPROM - 0000H.

Z obou paralelních portů má IO3 prioritu; proto - při použití (=osazení) jen jednoho dvoukanálového portu Z80-PIO je nutno tento obvod vložit do patice IO3.

Pro oba druhy pamětí bylo použito typů s kapacitou 2 KB. Zapojení - jakož i vlastní deska - umožňuje eventuální použití polovodičových pamětí s větší kapacitou, a to 4 KB (typu 2732 a 6132) po přemístění naznačených propojek.

Jsou-li osazeny obě paralelní brány IO3 a IO4, stojí pak uživateli k dispozici celkem 32 vstupně-výstupních vedení. Tato vedení - spolu ještě s některými dalšími řídicími signály - jsou vyvedena dvěma vícepólovými konektory. Jeden z nich je dvacetipólový a slouží pro IO4, druhý z nich má 31 pól a mimo signálů IO3 slouží i pro zemnění desky, přívod napájení a vnější iniciaci (reset). Obr. 48 znázorňuje rozložení součástí včetně obou komentovaných konektorů. Asi jedna třetina volné plochy desky může nést pro uživatele potřebné obvody, jako jsou např. oddělovací optoelektronické členy, analogově-digitální či digitálně-analogové převodníky apod.

Následující ukázka jednodeskového mikroprocesorového systému nejen že poskytuje uživateli ještě více vstupně-výstupních linek (celkem 52 + 6), ale kromě paralelních bran používá též sériové komunikace s okolím, a rovněž umožňuje testování stavu šesti linek, nehledě na oddělení mikroprocesorových sběrnic. Posledně uvedená skutečnost má svůj význam pro eventuální rozšíření daného systému.

Ze zapojení na obr. 49 zjistíme, že systém nepoužívá paralelních a sériových podpůrných obvodů ZILOG, ale - mnohem

rozšířenějších - programovatelných periferních obvodů fy INTEL, a sice dvojici paralelních portů 8255 a dvojici sériových obvodů 8251.

Mikroprocesor Z80 pracuje s hodinovým taktem o kmitočtu 2,4576 MHz; tento nezvyklý takt (stabilizovaný křemenným krystalem v obvodu IO11) však umožňuje jednoduchou výstavbu generátoru přenosové rychlostí připojením standardní děličky IO12 (SN74 LS 393), z jejíž vývodů 1 až 5 lze odebírat následující frekvence: 1 - 153,6 kHz, 2 - 76,5 kHz, 3 - 38,4 kHz, 4 - 19,2 kHz a 5 - 9,6 kHz. Protože přenosová rychlost každého kanálu obou UARTů (IO8 a IO9) je dána použitou frekvencí a programově nastaveným dělícím činitelem - který může být 1, 16 nebo 64 - lze dosáhnout rychlostí v rozmezí od 150 do 9600 b/s, což je zřejmé z níže uvedené tabulky:

nastavení	koeficient	přen. rychlost /bitů.s^{-1}/
1	1	nepoužívat
1	16	9600
1	64	2400
2	1	nepoužívat
2	16	4800
2	64	1200
3	1	nepoužívat
3	16	2400
3	64	600
4	1	nepoužívat
4	16	1200
4	64	300
5	1	nepoužívat /9600/
5	16	600
5	64	150

Je obvyklé použití činitele 16 (programově), čímž je možné dosáhnout pěti běžných rychlostí od 600 do 9600 bitů/s. (V případě použití další děličky bylo by možné dosáhnout nižších rychlostí při doporučeném činiteli 16.)

Kapacita programové paměti je 2 Kbyty a je dána typem 2716-IO10; kapacita paměti dat je taktéž 2 Kbyty; je realizována čtyřmi kusy standardních obvodů 2114 (s organizací á 4x1024 bitů). IO2 až IO5.IO6 a IO7 poskytují uživateli celkem 48 TTL kompatibilních vedení, která mohou sloužit jako vstupní či výstupní k řízení relé, převodníků atd., či obousměrně nebo ke styku se sběrnicí IEC (handshaking), apod. Při zapojení napájecího napětí jsou tyto obvody iniciovány účinkem kapacitněodporového členu R9C3, generujícím automaticky signál RESET. (Obdobně - iniciaci IO1 - mikroprocesoru zajišťuje automaticky odporověkapacitní člen C7R8 signálem $\overline{\text{RESET}}$.) IO14 pracuje jako adresový dekodér pro všechny v/v obvody; protože linky A0 a A1 jsou použity pro vnitřní adresování IO6 a IO7, je dekodér připojen až na adresová vedení A2, A3 a A4. Z možných osmi výstupů je jich využito však jen pět a zbývající tři jsou volné. Adresování stykových obvodů je tedy toto:

adresa	v/v obvod	funkce	pozn.
00H	IO6	port A	8255
01H	IO6	port B	
02H	IO6	port C	
03H	IO6	řízení	
04H	IO8	data	8251
05H	IO8	řízení	
08H	IO15	jen čtení	74LS367
10H	IO7	port A	8255
11H	IO7	port B	
12H	IO7	port C	
13H	IO7	řízení	
14H	IO9	data	8251
15H	IO9	řízení	

Obvod IO15 dovoluje prověřovat na jeho šesti vstupních vedeních úroveň příslušných signálů, které mohou být buď simulovány sepnutím či rozepnutím spínačů S1 až S6, nebo přiváděny z vnějšího okolí.

IO13 je stejný typ dekodéru, jak IO14 (74LS138); tento slouží však pro výběr jednotlivých bloků paměti ROM či RAM, a to signály Y0, Y1 či Y2. (Zbývající signály Y3 až Y7 je možno použít při eventuálním rozšíření o externí paměť na další desce.) Z uvedeného zapojení vyplývá umístění jednotlivých paměťových bloků:

paměť provozního programu IO10 - EPROM-00H až 7FFH,
paměť dat RAM 1 KB - 1000H až 13FFH, IO4 a IO5,
paměť dat RAM 1 KB - 2000H až 23FFH, IO2 a IO3.

Adresování pamětí je v krocích po dvou Kbytech; vzhledem k umístění prvního bloku RAM od EPROM je možné - pokud by to provozní program svým rozsahem vyžadoval - použít na IO10 i typ s dvojnásobnou kapacitou 4 KB - 2732, pochopitelně po nutné změně na desce pro její adresování, spočívající v oddělení přívodu napětí +5 V k špičce č. 21,a připojení téže k adresové lince A11.

Datová sběrnice D0 až D7, připojená k obvodům IO2 až IO10 a IO15, je chráněná proti přetížení oddělovacími zesilovači IO17 a IO18, zapojenými antiparalelně, kde směr přenosu dat je určován řídicími signály $\overline{RD}$ a $\overline{WR}$. (Vhodnější by bylo použití jednoho obousměrného osmibitového oddělovacího zesilovače 74LS245 na tomto místě, obzvláště je-li jeho cena v zahraničí rovnocenná s jedním typem 74LS244.) Rovněž dolní část adresové sběrnice A0 až A7 je chráněná stejným typem - IO16 - oddělovacího zesilovače.

Přerušovací vstup $\overline{INT}$ může být použit uživatelem dle jeho potřeby, popřípadě - po vložení kondenzátoru C10 do desky a jeho zapájení, a tím i připojení ke generátoru přenosové rychlosti - po každém 256tém taktu. V tom případě musí být programátorem respektován přerušovací způsob (mode "1"), (při němž CPJ po převzetí požadavku provádí skok na adresu 038H, na níž pak musí začínat přerušovací rutina). Uvedené dovoluje programové použití jako časové základny s periodou 0,1042 ms /3/.

V.2. mc-CP/M mikropočítač

Poslední ukázkou této kapitoly je klasický osmibitový mikroprocesorový systém na bázi Z80 s dynamickou pamětí o kapacitě 64 KB, pamětí provozního programu 4 KB a dalšími sériovými a paralelními stykovými obvody; hlavní předností komentovaného systému však je, že byl autorem /13/ vytvořen pro použití s populárním operačním systémem CP/M - samozřejmě v disketové verzi.

CP/M je provozní operační systém pro mikropočítače a mikroprocesorové systémy řady 80XX, tj. opírající se o mikroprocesory rodiny INTEL (8080, 8085, 8086, ...) nebo ZILOG (Z80, Z8, Z8000, Z800). Tento systém má za úkol předložit uživateli standardní programové připojení pro přenos dat mezi obrazovkovým terminálem a pružným diskem (disketou) či pružnými disky - viz dále. (Provozní program komentovaného mikropočítače pracuje se standardním CP/M, jenž se vyznačuje uživatelskou startovací adresou na buňce 100H.)

Mikropočítač CP/M sestává ze tří desek evropského formátu o rozměrech každé 100x160 mm. Jsou to:

1. deska CPJ, paměti RAM 64 KB a EPROM 4 KB včetně oddělovacích zesilovačů a přepínací logiky (bootstrap)
2. deska sériového a paralelního styku
3. deska disketového řadiče s kontrolérem 1797

Propojením desek 1 a 2 již vzniká provozuschopný mikroprocesorový systém, komunikující s vnějším okolím prostřednictvím dvou sériových kanálů podle normy V 24 (RS 232C) a dvou paralelních osmibitových kanálů. Provozní program - monitor - používá pouze oba sériové kanály (jeden pro obrazovkový terminál s klávesnicí, druhý případně pro tiskárnu), takže oba paralelní jsou plně k dispozici uživateli. Vlastní monitor používá na tři desítky povelů /1,13/ a obsahuje i rutiny pro obsluhu pružných disků. Pro řízení disket je však třeba třetí desky, poměrně hustě osazené, s níž teprve vznikne vlastní CP/M systém.

A nyní k vlastní desce procesoru několik slov: jejím jádrem je Z80-CPU, umožňující jak elegantní programování využitím rozsáhlejší množiny instrukcí, tak i převzetí většiny již vytvořených programů pro starší mikroprocesor I 8080A.

Mikroprocesor pracuje s hodinovým taktem 4 MHz, přestože v zapojení desky na obr. 50 je použit krystal o resonanci 8 MHz. Umístěním vhodné propojky za dělič F1 či F2 získáme odpovídající pracovní kmitočet, a to s ohledem na vybavovací dobu použitých velkokapacitních pamětí. Zapojení je navrženo pro rychlou verzi mikroprocesoru, typu Z80B; v tom případě činí takt 6 MHz - jemu či jeho násobku musí pak odpovídat použitý krystal (tzn. 12 nebo 6 MHz). CPJ je propojena s paměťmi vnitřními sběrnicemi, které - pro styk s vnějším okolím včetně návazných desek - jsou opatřeny obousměrnými oddělovacími zesilovači B1, B2 a B3.

Pro paměť dat a uživatelských programů bylo použito v současné době nejmodernějších dynamických pamětí, osmi kusů, typu HM 4864-3, z nichž každé pouzdro má organizci 65 536 x 1 bit. Nutné osvěžování jejich obsahu obstarává procesor sám. Proto, aby při iniciaci procesoru nebyl ztracen obsah pamětí RAM, musí mít nulovací impuls ($\overline{\text{RESET}}$) definovanou délku. To zajišťuje monostabilní obvod F3, který po nabití kondenzátoru C1 přes odpor R1 při dosažení prahové hodnoty vyšle jeden impuls o trvání daném hodnotami C2 a R2, tedy účinný, avšak dostatečně krátký.

Nulovací impuls je veden nejen na příslušný vstup mikroprocesoru (26), ale též na přepínací logiku, a sice na vstup hradla N2 tvořícího s N1 klopný obvod RS. Tehdy je výstup N2 na úrovni logické "1" a přepínací logika je aktivována. Jeden vstup hradla 09 má tak signál "0", a tak je stav adresy A15 transferován na vstupy hradel 05 a N7. Je-li úroveň adresového vedení A15 rovna log "0", pak - při výskytu signálu $\overline{\text{MREQ}}$ - je na výstupu součtového členu 05 též úroveň log "0", a tak je paměť Sp1 uvolněna (EPROM 2732). Současně je však výstup hradla N7 na log "1", čímž je výstup součtového členu 03 tak-

též na log "1". Z toho plyne, že paměť Sp2 - RAM je naopak nepřipojena. Změní-li se úroveň adresového vedení A15 na log "1", je "odpojena" EPROM a uvolněna paměť RAM.

Proč vlastně tato manipulace? Na začátku, tj. po iniciaci tlačítkem reset či po spuštění počítače nachází se v oblasti 0000H až 0FFFH provozní program-monitor včetně přepínacích rutin, přičemž paměť RAM má volných pouze horních 32 KB, tj. od adresy 7FFFH do FFFFH. Potom se provádí přesun celého monitoru (pomocí makroinstrukce pro přesun bloku) do nejvyšší části rozsahu RAM, načež následuje skok do monitoru - nacházejícího se nyní již však na "vrcholu" celkové kapacity RAM, tzn. nyní již od adresy EFE3H. Dolních 32 KB paměti (od 000H do 7FFFH) je uvolněno při "odpojení" EPROM, takže nyní má uživatel k dispozici souvislých 60 KB paměti RAM od adresy 0000H. Výměna spodních 32 KB však je alokována pouze požadavkem v/v čtení v rozsahu 4000H až 7FFFH, což monitor programově automaticky sám zajišťuje. Hardwareově je zajištěno odpojení přepínací (zavlékací) logiky propojením invertoru N3 na hradlo N4, jakož i součtových členů O1 a O2 na klopný obvod RS. (Výstup O2 má signál s úrovní log "0",je-li požadován uvedený adresový rozsah, tzn. při úrovni A14 rovné log "1".) Stavem klopného obvodu RS je pak přepínací logika deaktivována.

Datová sběrnice je připojena trvale přes oddělovací zesilovače B4 na v s t u p y paměti Sp2, avšak její výstupy jsou připojovány přes B5 jen tehdy, vznikl-li požadavek čtení. Adresová sběrnice paměti Sp2 je multiplexována, a to pomocí multiplexorů S1 a S2, jimiž je jednou připojena dolní osmibitová polovina adresové sběrnice, jednou horní osmibitová polovina.

Na obr. 51 je zakresleno časování, týkající se popisovaného multiplexování. Multiplexory S1 a S2 při výběrovém signálu sel = log "1" připojují na vstupy paměti Sp2 adresová vedení A0 až A7. S příchodem signálu MREQ, vedeným na vstup RAS (4) je připravován výdej dat z adresované řady a osvěžení.

(RAS přebírá dolní adresu.) Signál MREQ prochází též zpožďovacími členy D1 až D4; jeho posunutá (zpožděná) hrana představuje již přechod výběrového signálu sel do úrovně log "0". Tím jsou multiplexory přepnuty pro průchod adresových signálů A8 až A15. Členy D5 a D6 je však signál MREQ ještě více zpožděn, a v této podobě přichází na vstup součtového členu 03. Pakliže je 03 průchozí, objeví se zpožděný signál na vstupu CAS (15) paměti Sp2, kde způsobí vydání dat nyní již kompletně adresovaných. Zpožděný signál prochází též součtovým členem 04 (spolu se signálem RD), čímž uvolní budič B5 pro čtení z Sp2. Se stoupající hranou (tj. výstupní) signálu RAS je současně ukončeno i osvěžení.

Sběrnicová logika, sestávající z členů a hradel J7, 06, 07, N5, 08 a N6, přebírá řízení směru propustnosti oddělovacích zesilovačů - budičů sběrnic B1, B2 a B3. To je potřebné pro rozlišení vnějších a vnitřních sběrnic; uvedené přichází v úvahu např. při provozu DMA = přímém přístupu do paměti (indikovaným signálem BUSAK), kdy adresové budiče musí pracovat zvnějšku dovnitř - tedy obráceně, než je obvyklé.

To je prakticky vše, co bylo třeba říci k desce CPJ. Z technického hlediska - pro případné aplikanty - zbývá pouze upozornit, že citované dynamické paměti - v šestnáctivývodových pouzdrech DIL - mají napájení +5 V na vývodu č. 8 a zemnění na vývodu č. 16 - tedy právě o b r á c e n ě, než je zvykem u obvodů TTL.

Obraťme nyní svou pozornost k druhé desce - desce rozhraní, jejíž kompletní schéma je na obr. 52.

Zapojení této desky je celkem jednoduché: sestává z jednoho programovatelného obvodu Z80-SIO pro sériový styk, jednoho programovatelného obvodu Z80-PIO pro paralelní styk, dvojitého generátoru přenosové rychlosti BD11a BD2 (2x4702BPC fy Fairchild), oddělovacího zesilovače datové sběrnice B1 (74LS 85), dvojice stykových obvodů RS 232C (2x 1489 + 1488) - B2 a B3, a konečně jednoho čtyřnásobného dvouvstupového hradla NAND (74LS00) pro pomocné funkce - I1, I2 a N2.

Styková deska, byť i jen částečně osazená - sériovým obvodem SIO s příslušenstvím - zajišťuje již funkční provozuschopnost mikropočítače CP/M.

SIO a PIO jsou připojeny k vnější datové sběrnici přes oddělovač B1, jehož uvolňování je řízeno logikou sestávající z I1, N1, D1, V1 a I2. Pokud se použije na desce CPJ mikroprocesoru verze A (4 MHz), pak jak SIO, tak i PIO musí této rychlostní verzi odpovídat, což se týká stejně i verze B (6 MHz). Oba obvody vyžadují celkem osm adres, jež se pro A4 až A7 dají nastavit na desce propojkami J4 až J7; pokud se použije originální monitor, pak propojky odpadají a adresy pro stykové obvody jsou následující:

OF0H - data SIO, kanál A
OF1H - status SIO, kanál A
OF2H - data SIO, kanál B
OF3H - status SIO, kanál B
OF4H - data PIO, kanál A
OF5H - komando PIO, kanál A
OF6H - data PIO, kanál B
OF7H - komando PIO, kanál B

Hradlem N1 přes invertor I1 je oddělovač B1 jen tehdy uvolněn, je-li - ze strany mikroprocesoru - osloven výše uvedený adresový rozsah. Směr přenosu dat oddělovacím zesilovačem B1 je dán pak úrovní řídicího signálu $\overline{RD}$. Výstup na standardní sběrnici V24 (RS 232C) je zajištěn výkonovými zesilovači - převodníky úrovně B2 a B3. Přenosovou rychlost v Bd je možno pro každý kanál SIO nastavit odděleně - proto jsou v zapojení použity dva generátory BD1 a BD2, jež však jsou buzeny společným taktem, odvozeným ze stabilizovaného kmitočtu 2,4576 MHz.

Monitor

Má-li kterýkoli mikropočítač či mikroprocesorový systém být provozuschopný, musí být vybaven provozním programem - uloženým v pevné paměti ROM - neboli tzv. monitorem. Tento provozní program pak umožňuje nejen komunikaci mezi obsluho-

(dálnopis) všechny funkce, tj. řídicí konsoly, čtecího a vydávacího zařízení jakož i tiskárny protokolu. Jinými slovy řečeno, povelem A je ovlivněn obsah IOBYT, což je jméno paměťové buňky signalizující mikroprocesorovému systému, který v/v kanál má být obsloužen.

Další povely

B
Stisknutím písmene B z klávesnice udělujeme počítači povel, jímž je celá vstupní klávesnice blokována s výjimkou CTRL N; jimi je opět uvolněna.

C (počáteční adresa), (konečná adresa)
Povelem C přikazujeme srovnání obsahu paměti daného adresového rozsahu se čtenou oblastí přes stykový v/v obvod.

D (počáteční adresa), (konečná adresa)
Povelem D přikazujeme vydání obsahu paměti daného adresového rozsahu, a to v šestnáctkovém kódu. Vydávání může být zastaveno stisknutím kláves CTRL C.

E /(konečná adresa)/
Tímto povelem se generuje EOF (end of file) znaménko pro šestnáctkový formát INTEL (přes kanál děrovačky); případně zadaná adresa je při zavlékací rutině interpretována jako adresa počátku.

F (počáteční adresa), (konečná adresa), (konstanta)
Povelem F se způsobí zaplnění paměti daného adresového rozsahu zvolenou konstantou.

G (počáteční adresa), /(break 1)/, /(break 2)/
Povel pro odstartování programu (po CR) od zadané počáteční adresy, popřípadě od prvního či druhého testovacího bodu (break); tyto body však mohou být zaměřeny jen na první byty instrukcí.

H (hodnota 1), (hodnota 2)
Povelem se vytvoří (po CR) součet a rozdíl obou hodnot v šestnáctkové aritmetice.

J (počáteční adresa), (konečná adresa)
Povelem J se vyvolá rychlý test daného paměťového bloku. Test nepůsobí rušivě na případný obsah.

L (počáteční adresa)
Binární vkládání ze čtecího kanálu, a sice od udané adresy. Každých osm bitů tvoří jednotku; povel je protikladem povelu U.

M (počáteční adresa), (konečná adresa), (cílová adresa)
Transportní povel pro přenos daného paměťového bloku na místo počínající cílovou adresou; tato musí být vždy vně paměťového bloku, jinak došlo by k narušení obsahu.

N
Povel pro generování 72 znaků "0" přes kanál děrovače, popřípadě k synchronizaci záznamu na kasetový magnetofon.

P (počáteční adresa)
Vkládání znaků ASCII do paměti od udané počáteční adresy. Při případném omylu může být klávesou "-" zrušen poslední zadaný znak; klávesami CTRL D je vkládání ukončeno a monitorem je vydána poslední adresa.

QI /adresa brány)
QO (adresa brány), (hodnota)
Povelem Q ve spojení se znakem I (input) nebo O (output) lze přečíst z adresované brány vstupní hodnotu, nebo adresovanou bránu naplnit danou hodnotou.

R /(bias)/, /(reladr)/
Povelem R je možné přečíst šestnáctkový soubor dat ve formátu INTEL přes čtecí kanál. V případě zadání hodnoty "bias" ukládá se soubor na adresu obsaženou v souboru, avšak zvětšenou o hodnotu "bias". V případě zadání relativní adresy je soubor uložen v paměti počínaje touto adresou, pokud se jedná o soubor uložený v TDL relokačním formátu (obsahujícím informace o skokových adresách a adresových vztazích tak, že je možné p r o v o z u s c h o p n é uložení daného programu na libovolném místě /1/).

S (adresa)
Povelem S (po SP = mezerníku) je vydán obsah paměťové buňky dané adresy. Znakem "-" (baekstep) je vydán obsah předcházející buňky, uložením šestnáctkového výrazu (po SP) je původní obsah přepsán. Klávesnicí CR (carriage return = návrat vozíku) je povel ukončen.

T (počáteční adresa), (konečná adresa)
Vydání paměťového bloku při znázornění ve znacích ASCII (tedy např. místo 41H písmeno A, místo 31H číslice 1 atd.). Pokud není možné přiřadit dvojici nibblů (tetrád) odpovídající znak ASCII, je vydána tečka.

U (počáteční adresa), (konečná adresa)
Vydání binárního bloku adresované paměti přes kanál děrovače; povelem L lze blok opět načíst.

V (počáteční adresa), (konečná adresa), (cílová adresa)
Adresovaný blok paměti je byt po bytu srovnáván s obsahem bloku začínajícím na cílové adrese. Případné rozdíly jsou vydány konsolou (terminálem).

W (počáteční adresa), (konečná adresa)
Vydání adresovaného bloku paměti v šestnáctkovém formátu INTEL (tzv. dvoutečkovém) přes kanál děrovače. Blok lze načíst povelem R.

X /(jméno registru), (nový obsah)/
Povelem X (po CR) jsou vydány obsahy registrů A, B, C, D, E, F, H, L, M, P, S, I. Povelem X´ jsou vydány obsahy registrů A´, B´, C´, D´, E´, F´, H´, L´, M´, X, Y, R. Povelem XC může být např. modifikován nový obsah tohoto registru.

Y (hodnota 1), /(hodnota 2)/, /.../
Povelem Y je spuštěno (po CR) hledání sledu znaků o délce max 255 bytů daného vzoru hodnot; např.: Y20, 30, 40 vyhledává v celém rozsahu paměti vzor šestnáctkových hodnot 20, 30, 40; ukončení klávesami CTRL C.

Z
Označuje šestnáctkově adresu nejvýše uloženého místa souvislého paměťového bloku; (pro mikropočítač mc-CP/M to tedy je EFFFH).

I
Povel pro zavedení pružného disku (tzv. floppy boot start). Po něm se hlásí monitor (např. u verze 3.1, blíže popsané v /44/ dotazem, zda v konfiguraci je použit standardní - osmipalcový - pružný disk či tzv. minifloppy o průměru 5a1/4 palce); dále přebírá aktivitu BIOS (basic input-output systém) jakožto podstatná programová část operačního systému CP/M - viz /9/. Zmíněný BIOS - ať již ve verzi V1.0 dle /13c/ nebo upravené V3.1 dle /44/ - spolupracuje s monitorem, jehož některých rutin výhodně využívá.

Vlastní monitor mikropočítače mc-CP/M, jehož výpis dále přinášíme, vznikl na podkladě dvoukilobytového monitoru fy Technical Design Labs /1/. Liší se od něj povelem I, za nímž se ovšem skrývají potřebné rutiny komunikace s disketami, což se mj. projevilo na bytovém rozsahu, který činí 3,5 KB. V uveřejněné verzi pracuje se standardním osmipalcovým pružným diskem jednoduché nebo dvojité hustoty. Podstatné na monitoru ovšem je, že se sám zapisuje do horních čtyř KB systémové paměti (tzn. do paměťového bloku od adresy F000H až do FFFFH) a odtud skokem startuje, přičemž EPROM - z níž byl generován - odpojí. Současně se monitor hlásí přes kanál SIO ležící na adrese 00F0H. Touto technikou je umožněno poskytnutí uživateli souvislé oblasti paměti RAM od 0000H do EFFFH, kde právě počáteční část (od 0000H) - obvykle obsazená monitorem - je tak vyhrazena k dispozici pro uložení parametrů operačního systému CP/M (od 0000H do 00FFH). Na následujícím vyobrazení je znázorněno rozdělení operační paměti pro systém CP/M - obr. 53.

Tab. 21

Provozní program - monitor /4 KB/

```
ron  abs                                                          checksun
0000 EFE3  31 FF FF 3E C9 32 00 F0 CD 00 F0 3B 3B                  += 068B
0010 EFF0  D1 21 12 00 19 11 00 F0 01 FF 0F ED B0 C3 00 F0         += 067D
0020 F000  C3 6B F0 C3 C2 F6 C3 D5 F6 C3 4C F5 C3 7E F5 C3        += 0C24
0030 F010  60 F5 C3 C5 F5 C3 E4 F3 C3 D8 F1 C3 50 F6 C3 2C        += 0BF0
0040 F020  F7 C3 87 F8 C3 87 F8 C3 87 F8 C3 85 F8 6B F8 12        += 0B75
0050 F030  FE 0D 0A 00 00 00 07 20 52 44 4B 2D 5A 41 50 50        += 0385
0060 F040  4C 45 20 4D 43 20 43 4F 4D 50 55 54 45 52 20 56        += 0446
0070 F050  32 2E 32 20 20 43 4F 50 59 52 49 47 48 54 20 28        += 03D3
0080 F060  43 29 20 31 39 38 32 20 0D 0A 00 0E F1 06 08 21        += 02C5
0090 F070  7F F0 ED 83 0E F3 06 08 21 87 F0 ED B3 18 10 01        += 077F
00A0 F080  00 03 C1 04 4C 05 68 01 00 03 E1 04 4C 05 68 3E        += 0361
00B0 F090  CF D3 F5 3E 03 D3 F5 3A 00 70 AF 32 7D F8 31 A2        += 0873
00C0 F0A0  F0 C3 5E F6 A6 F0 F9 EB 01 23 00 21 16 F8 ED B0        += 0971
00D0 F0B0  EB 01 A1 FF 09 E5 21 00 00 06 0A E5 10 FD 0E 1A        += 05C5
00E0 F0C0  CD 4C F5 06 3A CD 0C F5 11 C8 F0 D5 CD BD F5 0E        += 0947
00F0 F0D0  3E CD 4C F5 CD A4 F7 E6 7F 28 F9 D6 41 F8 FE 1A        += 0A61
0100 F0E0  D0 87 21 F2 F0 85 6F 7C CE 00 67 7E 23 66 6F 0E        += 0783
0110 F0F0  02 E9 90 F1 DD F1 0A F2 2B F2 42 F2 5E F2 6B F2        += 0A34
0120 F100  22 F6 45 F1 B9 F2 84 F8 F1 F6 07 F2 AA F5 7E F8        += 0C3A
0130 F110  EB F1 C5 F7 E2 F2 9B F3 C4 F3 95 F5 F0 F7 30 F4        += 0046
0140 F120  6D F4 E8 F3 3D F5 0D 0A 53 4F 46 54 42 4F 4F 54        += 06F5
0150 F130  20 3D 20 31 0D 0A 49 2D 42 45 46 45 48 4C 20 3D        += 033E
0160 F140  20 49 0D 0A 00 CD 3D FA 21 26 F1 CD 86 F1 CD C2        += 078F
0170 F150  F6 FE 49 CA 91 F8 FE 31 C0 3E F1 32 10 00 3E DB        += 08F9
0180 F160  32 11 00 3E 40 32 12 00 3E FB 32 13 00 3E C9 32        += 03BC
0190 F170  14 00 21 80 00 16 00 1E 01 06 01 0E 00 CD 87 F8        += 034B
01A0 F180  DA 21 F5 C3 80 00 7E B7 C8 4F CD 4C F5 23 18 F6        += 08BE
01B0 F190  CD A4 F7 21 02 F8 01 00 04 11 05 00 BE 28 06 19        += 04A3
01C0 F1A0  0C 10 F9 18 15 59 CD A4 F7 FE 3D 20 F9 CD A4 F7        += 08BF
01D0 F1B0  01 00 04 23 BE 28 06 0C 10 F9 C3 21 F5 3E 03 1C        += 045F
01E0 F1C0  1D 28 08 CB 21 CB 21 17 17 18 F5 2F 57 CD B3 F6        += 065C
01F0 F1D0  30 F8 3A 70 F8 A2 B1 4F 79 32 7D F8 C9 CD BD F5        += 09E4
0200 F1E0  CD 9E F7 FE 1E 20 F9 D1 C3 BE F0 CD E3 F5 CD BD        += 0C00
0210 F1F0  F5 E1 CD 9E F7 FE 04 CA 44 F5 FE 5F 28 08 77 4F        += 0990
0220 F200  23 CD 4C F5 18 EC 2B 4E 18 F7 CD B0 F5 CD 36 F5        += 092F
0230 F210  BE C4 19 F2 CD 12 F6 18 F4 47 CD 47 F5 7E CD 33        += 093C
0240 F220  F6 CD 4A F5 78 CD 33 F6 C3 BD F5 CD B8 F5 CD 44        += 0B70
0250 F230  F5 CD 4A F5 7E CD 33 F6 CD 12 F6 7D E6 0F 20 F1        += 09CD
0260 F240  18 EC CD E3 F5 CD 77 F5 0E 3A CD 7E F5 AF CD 97        += 0A7D
0270 F250  F6 E1 CD 92 F6 21 00 00 CD 92 F6 C3 AA F5 CD D8        += 0AA9
0280 F260  F5 71 CD 18 F6 30 FA D1 C3 C8 F0 CD B3 F6 38 40        += 0AA5
0290 F270  28 10 CD 0A F6 D1 21 34 00 39 72 2B 73 78 FE 0D        += 05F7
02A0 F280  28 2E 16 02 21 35 00 39 E5 CD E3 F5 58 C1 E1 78        += 06F9
02B0 F290  B1 28 0A 71 23 70 23 0A 77 23 3E FF 02 7B FE 0D        += 0573
02C0 F2A0  28 03 15 20 E3 3E C3 32 38 00 21 1E F0 22 39 00        += 0438
02D0 F2B0  CD BD F5 D1 21 16 00 39 E9 CD B8 F5 7E 47 2F 77        += 088E
02E0 F2C0  AE 28 0E D5 50 5F CD 47 F5 CD D7 F7 CD B0 F5 42        += 09CD
02F0 F2D0  D1 70 CD 12 F6 18 E5 CD D8 F5 7E 02 03 CD 12 F6        += 0985
0300 F2E0  18 F8 CD E3 F5 78 D6 0D 47 4F D1 28 04 CD E3 F5        += 0948
0310 F2F0  C1 EB D9 CD BD F5 CD EB F6 D6 3A 47 E6 FE 20 F6        += 0C03
0320 F300  57 CD 7C F3 5F CD 7C F3 F5 CD 7C F3 D9 D1 5F C5        += 0B2D
0330 F310  D5 E5 19 E3 DD E1 D9 E1 CD 7C F3 3D 78 C1 20 03        += 0A03
0340 F320  09 DD 09 1C 1D 28 19 3D 28 22 CD 7C F3 CD 8F F3        += 067B
0350 F330  28 F8 CD 7C F3 28 BF DD E5 E1 CD 2E F6 C3 21 F5        += 08A8
0360 F340  7C B5 C8 EB 21 34 00 39 72 2B 73 C9 2E 01 CD 6C        += 0683
0370 F350  F3 38 07 CD 8F F3 20 F6 18 D8 4F CD 6C F3 47 D9        += 0922
0380 F360  C5 D9 E3 09 7D CD 8F F3 7C E1 18 E7 2D 20 07 CD        += 08D3
0390 F370  7C F3 1D 67 2E 08 CD 7C F3 CB 24 C9 C5 CD 7F F6        += 0924
03A0 F380  07 07 07 07 4F CD 7F F6 B1 4F 82 57 79 C1 C9 DD        += 0766
03B0 F390  77 00 DD BE 00 20 A0 DD 23 1D C9 CD E3 F5 E1 7E        += 08BC
03C0 F3A0  CD 33 F6 CD AE F6 D8 28 0F FE 5F 28 14 E5 CD 0A        += 08CB
03D0 F3B0  F6 D1 E1 73 78 FE 0D C8 23 7D E6 07 CC 44 F5 18        += 0918
03E0 F3C0  DE 2B 18 F5 CD B8 F5 CD 44 F5 06 40 7E E6 7F FE        += 09BD
03F0 F3D0  20 30 02 3E 2E FE 7C 30 FA 4F CD 4C F5 CD 12 F6        += 0794
0400 F3E0  10 EA 18 E3 3A 7D F8 C9 16 00 CD E3 F5 E1 65 E5        += 0953
```

```
040D F3F0  33 14 78 D6 0D 20 F3 47 4F 67 6A 2D 39 E5 C5 C5   += 06F1
041D F400  CD 8D F5 C1 E1 DD E1 5A DD 7E 00 ED 81 E2 28 F4   += 0B33
042D F410  DD E5 E5 10 28 0B DD 7E FF BE 20 E8 23 DD 2B 18   += 085A
043D F420  F2 E1 E5 2B C5 CD 2E F6 C1 10 D4 33 1D 20 FC C9   += 097B
044D F430  CD B8 F5 CD 77 F5 01 3A 00 CD 7E F5 D5 E5 04 CD   += 0989
045D F440  18 F6 38 24 3E 18 90 20 F5 E1 CD 58 F4 D1 18 E3   += 0823
046D F450  57 78 CD 97 F6 CD 92 F6 AF CD 97 F6 7E CD 97 F6   += 0B5F
047D F460  23 10 F9 AF 92 C3 97 F6 E1 D1 AF 18 E3 CD A4 F7   += 0A81
048D F470  21 39 F8 FE 0D 28 5A FE 27 20 0A 21 52 F8 CD A4   += 070A
049D F480  F7 FE 0D 28 4C BE 28 09 CB 7E C2 21 F5 23 23 18   += 06E4
04AD F490  F4 CD 4A F5 23 7E 47 E6 3F EB 6F 26 00 39 EB 23   += 07D4
04BD F4A0  1A CD 33 F6 CB 78 28 05 1B 1A CD 33 F6 CD AF F6   += 081C
04CD F4B0  D8 28 19 E5 C5 CD 0A F6 E1 F1 C5 F5 7D 12 C1 CB   += 0A37
04DD F4C0  78 28 03 13 7C 12 C1 E1 78 FE 0D C8 CB 7E C0 18   += 0752
04ED F4D0  C3 CD BD F5 CD 4A F5 7E 23 B7 F8 4F CD 4C F5 0E   += 0A09
04FD F4E0  3D CD 4C F5 7E 47 E6 3F 23 EB 6F 26 00 39 EB C8   += 07C7
050D F4F0  70 20 0F 1A CD 33 F6 CB 78 28 D9 1B 1A CD 33 F6   += 071E
051D F500  18 D2 E5 1A 67 1B 1A 6F 7E E1 18 F1 21 31 F8 4E   += 06EC
052D F510  23 CD 4C F5 10 F9 CD C5 F5 B7 C8 CD 9E F7 FE 03   += 0AA3
053D F520  C0 CD 5D F6 11 EA FF 19 F9 0E 07 CD 4C F5 0E 2A   += 0847
054D F530  CD 4C F5 C3 C8 F0 CD D5 F6 30 E6 0A C9 CD 3D F6   += 08E2
055D F540  01 23 00 09 CD BD F5 CD 2E F6 0E 20 3A 7D F8 E6   += 0760
056D F550  03 20 0A DB F1 E6 04 28 FA 79 D3 F0 C9 C3 71 F8   += 0936
057D F560  3A 7D F8 E6 C0 CA 53 F5 FE 80 C2 77 F8 DB F3 E6   += 08CA
058D F570  04 28 FA 79 D3 F2 C9 0E 0D CD 7E F5 0E 0A 3A 7D   += 0757
059D F580  F8 E6 30 CA 4C F5 FE 10 C2 74 F8 DB F3 E6 04 28   += 0A35
05AD F590  FA 79 D3 F2 C9 CD B8 F5 CD 47 F6 CD 42 F6 4E CD   += 08A5
05BD F5A0  7E F5 CD 18 F6 30 F7 CD 42 F6 CD 47 F6 C9 E6 0F   += 0A42
05CD F5B0  C6 90 27 CE 40 27 4F C9 CD E5 F5 D1 E1 E5 06 05   += 0913
05DD F5C0  CD 0C F5 E1 C9 3A 7D F8 E6 03 20 09 DB F1 E6 01   += 08EC
05ED F5D0  C8 3E FF B7 C9 C3 7A F8 0C CD E5 F5 CD B0 F5 C1   += 08AD
05FD F5E0  D1 E1 C9 0E 01 21 00 00 CD A4 F7 47 CD B2 F6 38   += 07D7
060D F5F0  08 29 29 29 29 B5 6F 18 EF E3 E5 78 CD B6 F6 30   += 07C0
061D F600  02 0D C8 C2 21 F5 0D 20 DC C9 0E 01 21 00 00 C3   += 0574
062D F610  EB F5 CD 18 F6 D0 D1 C9 23 7C B5 37 C8 7B 95 7A   += 0A02
063D F620  9C C9 CD B8 F5 E5 19 CD 47 F5 E1 B7 ED 52 7C CD   += 0806
064D F630  33 F6 7D F5 0F 0F 0F 0F CD 3C F6 F1 CD AE F5 C3   += 08FA
065D F640  4C F5 01 FF 08 18 03 01 00 48 CD 7E F5 10 FB C9   += 06C1
066D F650  E5 CD 5D F6 7D D6 3C 30 01 25 44 E1 C9 C5 21 FF   += 08BD
067D F660  FF 24 7E 2F 77 BE 2F 77 20 F7 24 7C FE F0 28 08   += 0780
068D F670  7E 2F 77 BE 2F 77 28 F2 25 01 DD FF 09 C1 C9 CD   += 0804
069D F680  EB F6 D6 30 D8 FE 17 38 D8 FE 0A 3F D0 D6 07 FE   += 09DD
06AD F690  0A C9 7C CD 97 F6 7D F5 0F 0F 0F 0F CD AE F5 CD   += 0894
06BD F6A0  7E F5 F1 F5 CD AE F5 CD 7E F5 F1 82 57 C9 0E 2D   += 08D7
06CD F6B0  CD 4C F5 CD A4 F7 FE 20 C8 FE 2C C8 FE 0D 37 C8   += 0858
06DD F6C0  3F C9 3A 7D F8 E6 03 20 09 DB F1 E6 01 28 FA DB   += 0879
06ED F6D0  F0 C9 C3 6B F8 3A 7D F8 E6 0C CA C2 F6 FE 04 C2   += 08C6
06FD F6E0  6E F8 DB F3 E6 01 28 FA DB F2 C9 CD 36 F5 E6 7F   += 0830
070D F6F0  C9 CD E3 F5 E1 CD BD F5 16 FF 06 04 CD 36 F5 20   += 0A05
071D F700  F9 10 F9 CD 36 F5 28 FB 77 3E 07 23 CD 36 F5 28   += 081C
072D F710  03 77 18 F7 1E 01 CD 36 F5 20 09 1C 3E 07 BB 20   += 0505
073D F720  F5 C3 2E F6 72 23 1D 20 FB 77 18 DF E5 D5 C5 F5   += 098B
074D F730  CD 5D F6 E8 21 0A 00 39 06 04 EB 2B 72 2B 73 D1   += 0670
075D F740  10 F9 C1 0B F9 21 25 00 39 7E 91 23 20 04 7E 90   += 0581
076D F750  28 0C 23 23 7E 91 20 05 23 7E 90 28 01 03 21 20   += 034C
077D F760  00 39 73 23 72 23 23 71 23 70 C5 0E 40 CD 4C F5   += 05AC
078D F770  E1 CD 2E F6 21 25 00 39 01 00 02 5E 71 23 56 71   += 050D
079D F780  23 7B B2 28 02 7E 12 23 10 F1 08 D9 E5 D5 C5 F5   += 0783
07AD F790  DD E5 FD E5 ED 57 47 ED 5F 4F C5 C3 C8 F0 CD C2   += 0B99
07BD F7A0  F6 E6 7F C9 CD 9E F7 C8 3C F8 3D FE 0D C8 FE 4E   += 0ADE
07CD F7B0  C8 FE 6E 28 0D C5 4F CD 4C F5 79 C1 FE 40 D8 FE   += 09D9
07DD F7C0  7B D8 E6 5F C9 CD A4 F7 FE 4F 28 1C FE 49 C2 21   += 097C
07ED F7D0  F5 CD E3 F5 C1 ED 58 06 08 CD 4A F5 CB 23 3E 18   += 08FE
07FD F7E0  8F 4F CD 4C F5 10 F5 C9 CD E5 F5 D1 C1 ED 59 C9   += 0B02
```

pokračování 2

```
0800 F7F0  CD D8 F5 0A 8E 28 05 C5 CD 19 F2 C1 03 CD 12 F6   += 08C5
0810 F800  18 F1 43 54 56 42 55 52 54 50 43 55 50 54 50 43   += 0552
0820 F810  55 4C 54 56 4C 55 C1 79 ED 4F 78 ED 47 FD E1 DD   += 08C9
0830 F820  E1 F1 C1 D1 E1 08 D9 D1 C1 F1 E1 F9 00 21 00 00   += 09A4
0840 F830  C3 00 00 00 00 00 00 00 00 00 41 15 42 13 43 12 44   += 0207
0850 F840  11 45 18 46 14 48 31 4C 30 4D F1 50 B4 53 97 49   += 052A
0860 F850  03 C1 41 09 42 0B 43 0A 44 0D 45 0C 46 08 48 0F   += 02EF
0870 F860  4C 0E 4D CF 58 87 59 85 52 02 C1 C3 C9 F6 C3 C9   += 0856
0880 F870  F6 C3 53 F5 C3 53 F5 C3 53 F5 C3 CC F5 00 3D 7D   += 0855
0890 F880  32 14 3D CD 6A 1A 21 DB 40 FB CD 0C F9 F3 C9 DB   += 0874
08A0 F890  44 E6 20 20 03 3E 04 C9 AF C9 7B D3 42 79 E6 0F   += 06EE
08B0 F8A0  5F 0E 43 06 80 CD 8F F8 C6 88 D3 40 7B D3 44 DB   += 0858
08C0 F8B0  44 E6 80 CA AF F8 ED A2 C2 AF F8 18 FE 7B D3 42   += 0A89
08D0 F8C0  79 E6 0F 5F 0E 43 06 80 CD 8F F8 C6 A8 D3 40 7B   += 07F4
08E0 F8D0  DB 44 E6 80 CA D8 F8 ED A3 C2 D0 F8 18 FE C5 01   += 0B0D
08F0 F8E0  F4 01 0B 78 B1 C2 E2 F8 C1 C9 CD DE F8 79 E6 0F   += 0A60
0900 F8F0  D3 44 7B D3 42 7A D3 43 3E 1C D3 40 18 FE CD DE   += 0865
0910 F900  F8 79 E6 0F D3 44 3E 0C D3 40 18 FE 79 3C FE 01   += 07A4
0920 F910  28 0E FE 02 28 0A FE 03 20 04 3E 04 18 02 3E 08   += 032F
0930 F920  4F AF 32 32 FA 78 FE 00 C2 33 F9 CD FE F8 E6 88   += 09F1
0940 F930  C0 AF C9 C5 79 CD F8 F9 4F 3A 33 FA CD F8 F9 89   += 0B61
0950 F940  C1 CA 80 F9 E5 C5 21 34 FA 3A 33 FA CD F8 F9 4F   += 0A71
0960 F950  06 00 09 DB 41 77 C1 E1 E5 C5 21 34 FA 79 CD F8   += 0878
0970 F960  F9 4F 06 00 09 7E D3 41 C1 E1 79 32 33 FA F5 C5   += 081D
0980 F970  79 E6 0F D3 44 01 65 0B 0B 78 B1 C2 78 F9 C1 F1   += 080F
0990 F980  CD 11 FA 30 08 CD 07 FA CD 11 FA 38 FB DB 41 BA   += 088F
09A0 F990  CA 89 F9 E5 D5 C5 CD EA F8 C1 D1 E1 E6 90 28 19   += 0B04
09B0 F9A0  E5 D5 C5 CD FE F8 C1 D1 E1 3A 32 FA 3C 32 32 FA   += 0A85
09C0 F9B0  FE 04 20 DF 3E FF B7 37 C9 78 FE 02 CA DB F9 E5   += 09F0
09D0 F9C0  D5 C5 CD 9A F8 C1 D1 E1 E6 9C C8 3A 32 FA 3C 32   += 0A8A
09E0 F9D0  32 FA FE 04 20 CA 3E FF B7 37 C9 E5 D5 C5 CD BD   += 0A15
09F0 F9E0  F8 C1 D1 E1 E6 FC C8 3A 32 FA 3C 32 32 FA FE 04   += 0A17
0A00 F9F0  C2 A0 F9 3E FF B7 37 C9 FE 04 20 04 3E 01 18 06   += 06D2
0A10 FA00  FE 08 20 02 3E 02 C9 E5 D5 C5 CD EA F8 C1 D1 E1   += 09D2
0A20 FA10  C9 C5 DB F4 E6 03 47 79 CD F8 F9 FE 01 20 08 78   += 0963
0A30 FA20  E6 01 28 0B C1 AF C9 78 E6 02 28 03 C1 AF C9 37   += 074E
0A40 FA30  C1 C9 00 01 00 00 00 00 00 00 00 00 3E 03 D3   += 029F
0A50 FA40  51 3E FF 32 07 FC C9 3E 03 D3 51 3E 02 D3 51 3E   += 0693
0A60 FA50  03 D3 51 DB 51 E6 04 28 FA C9 3E 81 D3 51 3E 80   += 07C9

0A70 FA60  D3 51 3E 83 D3 51 DB 51 E6 04 28 FA C9 3E 83 D3   += 081E
0A80 FA70  51 DB 51 E6 02 20 FA DB 51 E6 01 28 FA DB 50 F5   += 08D4
0A90 FA80  CD 47 FA F1 C9 3E 83 DB 51 E6 02 C0 CD 47 FA 18   += 0983
0AA0 FA90  F4 37 3E 01 C9 3E 01 D3 51 3E 01 D3 50 CD 5A FA   += 0719
0AB0 FAA0  3E 00 D3 50 CD 5A FA CD 6D FA E6 80 C2 91 FA AF   += 0A18
0AC0 FAB0  C9 3E 03 32 0F FE CD 85 FA 3E 02 D3 50 CD 5A FA   += 0819
0AD0 FAC0  3E 00 D3 50 CD 5A FA 7B D3 50 CD 5A FA 7A D3 50   += 08DE
0AE0 FAD0  CD 5A FA CD 6D FA E6 80 C2 ED FA 01 00 02 DB 50   += 0992
0AF0 FAE0  77 23 CD 47 FA 0B 78 B1 C2 DE FA AF C9 3A 0F FE   += 0935
0B00 FAF0  3D 32 0F FE C2 B6 FA C3 91 FA 22 10 FE 3E 03 32   += 07DF
0B10 FB00  0F FE 2A 10 FE CD 85 FA 3E 01 D3 51 3E 03 D3 50   += 0758
0B20 FB10  CD 5A FA 3E 00 D3 50 CD 5A FA 7B D3 50 CD 5A FA   += 0962
0B30 FB20  7A D3 50 CD 5A FA 01 00 02 7E 23 D3 50 CB 5A FA   += 07A6
0B40 FB30  0B 78 B1 C2 29 FB CD 6D FA E6 80 C8 3A 0F FE 3D   += 0900
0B50 FB40  32 0F FE C2 02 FB C3 91 FA E5 D5 C5 1A E6 03 21   += 08EF
0B60 FB50  0A FC 01 80 00 B7 28 04 09 3D 20 FC 22 0A FE 13   += 0509
0B70 FB60  13 1A 0F 0F E6 C0 47 1B 1A 0F 0F E6 3F B1 4F ED 47 1A   += 04C1
0B80 FB70  0F 0F E6 C0 4F 1B 1A 0F 0F E6 3F B1 4F ED 43 0C   += 05C7
0B90 FB80  FE C1 D1 E1 C9 78 32 0E FE CD 49 FB 3A 07 FC FE   += 0A3C
0BA0 FB90  FF 28 13 D5 E5 2A 08 FC ED 5B 0C FE AF ED 52 7D   += 08DF
0BB0 FBA0  B4 E1 D1 CA DC FB 3A 07 FC FE 01 C2 C3 FB E5 D5   += 0B7D
0BC0 FBB0  ED 5B 08 FC 21 0A FC CD FA FA D1 E1 DA 91 FA AF   += 0AFA
0BD0 FBC0  32 07 FC E5 D5 ED 5B 0C FE ED 53 08 FC 21 0A FC   += 08AC
0BE0 FBD0  CD B1 FA D1 E1 DA 91 FA AF 32 07 FC 3A 0E FE FE   += 0AB7
0BF0 FBE0  01 CA EB FB FE 02 CA F7 FB AF C9 ED 5B 0A FE 01   += 0A36
0C00 FBF0  80 08 EB ED B0 AF C9 3E 01 32 07 FC ED 5B 0A FE   += 0844
0C10 FC00  01 80 00 ED B0 AF C9 FF 00 00 19 34 D1 F1 E6 7F   += 0809
0C20 FC10  F5 87 FA 2E 32 2A F0 3F 7E B7 CA 1D 32 F1 C5 47   += 087A
```

V.4. Provozní systém CP/M

Po vzniku mikroprocesorů, v okamžiku kdy již byly na trhu provozuschopné mikropočítače, byl ten či onen vybaven provozním systémem (rozsáhlým provozním programem), umožňujícím jeho více či méně komfortní provoz a ovládání. Sestava a obsah provozního systému byl většinou tajemstvím výrobce; proto bylo na snadě, že jednotlivé systémy různých výrobců (ale i různých mikropočítačů stejného výrobce) se mezi sebou lišily. Struktura systému či jeho jednotlivé funkce též nebyly běžně známé. Přizpůsobení či převedení na jiný počítač, či stejný počítač, ale s jinými periferiemi, nebylo možné (bohužel, obdobné nesourodé systémy ještě i dnes existují). Avšak s příchodem systému CP/M (Control Program for Mikroprozessors), jehož "otcem" je Gary A. Kindall, byl tento nešvar pro mikropočítače opírající se o mikroprocesory rodiny 8080/8085/Z80 konečně odstraněn. První verze systému vznikla v roce 1974, její komerční nasazení však následovalo o rok později. V roce 1976 vzniká doplněná verze 1.3, nyní se používá vylepšená verze 2.2 nebo tzv. CP/M+ = verze 3.0.

Mnoho výrobců - po vzniku CP/M a seznámením se s jeho klady - pak následně vyvíjelo a vyrábělo počítače orientované na zmíněný provozní systém, čímž se současně dosáhlo i jeho rozšíření. Ovšem potvrzení úspěšnosti a životaschopnosti systému se dostalo až v roce 1982, kdy i velcí výrobci počítačů (ZENIT, IBM) přicházejí na trh s osobními mikropočítači (cenové třídy od 3000,- $) funkčně schopnými pracovat s CP/M.

A protože rychlé osobní počítače jsou osazovány šestnáctibitovými mikroprocesory - mezi nimiž převládá typ 8086 (80186 či či 80286) - byla vyvinuta pro tento mikroprocesor příslušná verze provozního systému, jež se nazývá CP/M-86. Proti verzi CP/M osmibitových mikroprocesorů ovšem není tak rozšířená, nicméně pro ni existuje celá řada programů.

Mikropočítač - obecně pojatý - sestává ze dvou principiálních částí: z technického vybavení (hardware) a programového

vybavení (software). Programové vybavení přitom plní úlohu dostupnosti problémového řešení s danou technikou. U malých mikropočítačů sestává programové vybavení - jak jsme již v předcházejícím vzpomínali - často jen z jediného jednoduchého monitorovacího (provozního) programu, jímž mohou být zadávány šestnáctkové instrukce uživatelských programů, prohlíženy a měněny obsahy registrů a pamětí, a startovány uživatelské programy, popřípadě i ukládány na vnější paměťové médium - zpravidla tvořené kasetovým magnetofonem - a rovněž i z něho čteny. Ukládání dat je obvykle realisováno šestnáctkovou klávesnicí, výstup pak na vícemístném sedmisegmentovém displeji LED. S takovýmto zařízením lze provádět vývoj programů minimálního rozsahu pro školní, předváděcí či řídicí účely; vyvinuté a odladěné (= ověřené) programy jsou však provozovatelné jen na daném technickém zařízení a nejsou tudíž přenositelné. Změní-li se např. monitor z nějakých příčin (např. jeho doplněním o další potřebné rutiny), nejsou zpravidla dříve vyvinuté programy již provozuschopné.

Aby bylo možno vyvíjet programy, které nejsou určeny jen pro jeden počítač, je nejprve potřebné vytvořit definovaná připojovací místa. U skutečně komfortních systémů jsou všechny potřebné podprogramy zahrnuty do seznamu skoků na začátku monitoru. Tento seznam, sestávající ze samých "JMP adresa", se nazývá tabulkou vektorů. Nyní může uživatel systému používat jednotlivé rutiny, pokud ovšem jsou uvedeny v seznamu. Pokud se pak mění verze takto vytvořeného monitoru, což je většinou přidáním dalších rutin, pak mohou být staré podprogramy dále používány, neboť tabulka vektorů se obvykle jen rozšíří (adresy nových rutin se skokovými příkazy se vždy připisují na konec tabulky na pro ně tam předvídavě vynechané místo - místo prázdných instrukcí NOP) - viz tab. č. 22.

Při koncepci systému CP/M se vycházelo z jasných představ o tom, že má být vytvořen systém skutečně univerzální, u nějž tabulka vektorů (= seznam potřebných programů) má zahrnovat rozmanitá, potřebná a náročným úlohám odpovídající připojova-

```
                                ; START DES MONITORPROGRAMMS
F000    C3 F06B                 JP BEGIN
F003    C3 F6C2                 JP CI
F006    C3 F6D5                 JP RI
F009    C3 F54C                 JP CO
F00C    C3 F57E                 JP POO
F00F    C3 F560                 JP LO
F012    C3 F5C5                 JP CSTS
F015    C3 F3E4                 JP IOBYTE
F018    C3 F1D8                 JP IOSET
F01B    C3 F650                 JP MEMCK
F01E                            TRAP:
F01E    C3 F72C                 JP RESTART
F021    C3 F887                 JP SOFT         ;-FLOPPY EXEC VEKTOR HARD
F024    C3 F887                 JP SOFT         ;FLOPPY EXEC VEKTOR SOFT
F027    C3 F887                 JP SOFT         ;-FLOPPY EXEC VEKTOR MINI
F02A    C3 F885                 JP IMISYS       ;PLATTE EXEC VECTOR
                                ; HIER NUR SOFT,IMISYS IMPLEMENTIERT
                                ; BEI IMISYS INDIREKTE TRK,SEKTOR ADR IN DE
                                ; VIER BYTES LANG LINEAR 0LXN
                                ;
F02D    F86B                    DEFW TABSTART   ;TABELLEN START USER BEREICH
                                                ;SPRUNGTABELLE , IMMER
                                                ;INDIREKT VERWENDEN
F02F    FE12                    DEFW LASTMON    ;ADRESSE LETZTE BELEGTE ZELLE
                                                ;DES MONITORS DAMACH PATCH FREI
                                ;


                                ;
F031                            MSG:
F031    0D 0A 00 00             DEFB 0DH,0AH,0,0,0
F035    00
F036    07                      DEFB 7 ; BELL
F037    20 52 44 4B             DEFM ' RDK-ZAPPLE MC COMPUTER'
F03B    2D 5A 41 50
F03F    50 4C 45 20
F043    4D 43 20 43
F047    4F 4D 50 55
F04B    54 45 52
F04E    20 56 32 2E             DEFM ' V2.2 '
F052    32 20
F054    20 43 4F 50             DEFM ' COPYRIGHT (C) 1982 '
F058    59 52 49 47
F05C    48 54 20 28
F060    43 29 20 31
F064    39 38 32 20
F068    0D 0A 00                DEFM 0DH,0AH,0
003A                            MSGL    EQU     $-MSG
                                ; INIT TEIL +++

F06B                            BEGIN:

                                ;    ------ INIT ------
F06B    0E F1                   LD C,SIOASTS
F06D    06 08                   LD B,8
F06F    21 F07F                 LD HL,TABSIO
F072    ED B3                   OTIR
F074    0E F3                   LD C,SIOBSTS
F076    06 08                   LD B,8
F078    21 F087                 LD HL,TABSIB
F078    ED B3                   OTIR
F07D    18 10                   JR SKSKL
F07F                            TABSIO:
F07F    01 00                   DEFB 1,0
F081    03 C1                   DEFB 3,11000001B        ;KEIN -CTS UND -DCD ENABLE
F083    04 4C                   DEFB 4,01001100B
F085    05 68                   DEFB 5,01101000B
                                ;
F087                            TABSIB:                 ;-CTS UND -DCD ENABLE
```

Tab. č. 22 - Počátek monitoru v assembleru, převedeného na adresu F000H, vyznačuje se tabulkou skoků na různé rutiny; struktura monitoru odpovídá části BIOS provozního systému CP/M

cí místa a způsoby ovládání. Při tom pochopitelně bylo upuštěno od použití šestnáctkových klávesnic a sedmisegmentových displejů. Nezanedbatelným předpokladem pro CP/M je však určitá vybavenost daného mikropočítače: musí být k dispozici připojitelný (nebo vestavěný) obrazovkový displej - terminál, dále pak vnější velkokapacitní paměťové médium. Druhý požadavek je u mikropočítačů zpravidla představován pružným diskem (8", 5al/4" nebo 3al/2"), popřípadě pevným diskem typu Winchester.

Protože hardware může být koncipováno velmi rozdílně, bylo třeba nalézt maximálně všeobecnou formu software. Tak například u obrazovkových terminálů existuje celá řada velmi rozdílných zařízení. V principu však je jedno všem společné: klávesnicí se zadávají znaky, jež jsou dále transportovány k počítači. Při výstupu z počítače k obrazovkovému terminálu musí být umožněn rovněž transport znaků. Totéž ovšem platí i o eventuálně připojitelné tiskárně. Je tedy možné se dohodnout na druhu rozhraní, jež předává jednotlivé znaky. Úlohou takového rozhraní pak je předání znaku, jenž se nachází v určitém registru procesoru, do vnějšího okolí a opačně, tj. převzetí znaku, který přichází z vnějšku. Přitom ovšem existuje několik důležitých omezení. U systému CP/M z možných kódových zobrazení znaků byl zvolen kód ASCII (= ISO 7 /5/); není tedy možné použít terminálu pracujícím s kódem EBCDIC bez převodníku. Další podmínkou je, že všechny znaky ASCII musí být použitým terminálem generovatelné. A protože paritní bit je u CP/M vždy roven nule, je to právě 128 znaků. Další podmínkou je, aby při vkládání do počítače jednotlivé znaky přicházely sekvenčně, rychlostí blízkou tempu strojopisu, tedy nikoliv naráz po blocích. Pokud se vydávání týče, není nutné, aby použitý terminál zobrazoval znaky malé abecedy. Vzhledem k tomu, že software CP/M má být vždy nezávislé na hardware, je u většiny programů možné jejich přizpůsobení různým typům terminálů.

CP/M je souborově orientovaný provozní systém. Soubor (file) u CP/M je souhrn dat téměř libovolné délky, uživatelem opatřený libovolným názvem do max osmi znaků s následným tří-

znakovým označením typu souboru za oddělovací tečkou. Soubor může být představován více typy, např.:
provozuschopným programem (název souboru.COM),
programem v šestnáctkovém kódu INTEL (název souboru.HEX),
programem ve zdrojovém kódu BASIC (název souboru.BAS), nebo
programem v assembleru (název souboru.ASM) a podobně.

Označení souboru a jeho vyhledání spočívá pro uživatele v pouhém využití odpovídající funkce provozního systému. Se skutečným fyzickým umístěním programu na disketě nemá uživatel co činit - to za něj automaticky obstarává CP/M, který sestává ze tří modulů:
1) CCP - console command processor,
2) BDOS - basic disk operating system, a
3) BIOS - basic input/output system.
(Uložení modulů CP/M v operační paměti mikropočítače bylo již schematicky znázorněno na konci statě "monitor" na obr. 53.)

1. CCP

Po nahrání provozního systému CP/M z diskety do operační paměti mikropočítače přebírá jeho modul CCP řízení celého systému. Hlásí se na obrazovce terminálu jako dvojice znaků "A " (tzv. prompt); tím sděluje uživateli, že je připraven k převzetí povelu. Nyní mohou být nahrávány programy (z vnějších paměťových médií) a postupně spouštěny. Nahrávání se děje zadáním označení disketové stanice (v CP/M jich může být použito až 16) a jména souboru. Nachází-li se program na disketě ve stanici označené promptem, může označení stanice (pohonného ústrojí diskety, drive) odpadnout. Volaný program obsahuje označení typu souboru, např. A > B:NAVTR.COM, po jeho nalezení modulem CCP je uložen do operační paměti od adresy 100H a též spuštěno jeho provedení.

CCP používá šest následujících povelů, jsou to: REN, ERA, DIR, SAVE, TYPE a USER.
REN ⟨zds.: jméno souboru.XYZ⟩
Tento povel dovoluje přejmenování zvoleného souboru libovolně vybraným názvem, (zds. = označení disketové stanice A až P).

ERA <zds.: jméno souboru.XYZ>
Tímto povelem mohou být na disketě zrušeny soubory či soubor. (Jméno souboru je ze seznamu "directory" vypuštěno, avšak data zůstávají. Teprve novým souborem, uloženým na místo zrušeného, jsou data přepsána.)

DIR <zds.:>
Pro zvolenou disketovou stanici, označenou "zds.", je uvedeným povelem vypsán seznam (= directory) souborů diskety v něm se nacházející.

SAVE nn. <zds.: jméno souboru.XYZ>
Tímto povelem je možné libovolně rozsáhlý obsah uživatelské paměti RAM od adresy 100H pod zvoleným názvem uložit na disketu v blocích po 256 bytech.

TYPE <zds.: jméno souboru.XYZ>
Povel znázorní obsah textového souboru na obrazovce terminálu (= konsoly).

USER nn
Od verze 2.2 CP/M může být paměťový prostor na disketě rozdělen až na 16 uživatelských částí. Uvedeným funkčním povelem s označením části (nn) může být vybraná část vyvolána.

2. BDOS

Jádrem hardwareově nezávislé části CP/M je BDOS. Předkládá 37 funkcí k použití, jež všechny jsou vyvolatelné uživatelskými programy. Funkce 1 až 12 obsluhují konsolu (= terminál), tiskárnu, čtečku a děrovačku děrné pásky. Další funkce se týkají disketových stanic. Univerzálnost CP/M právě spočívá v přesně stejně stanovených definicích rozhraní od uživatele k BDOS; přitom přizpůsobení na použitý počítač přes BIOS nemá na BDOS žádný vliv. Použití funkcí BDOS probíhá principiálně dle stejného schématu a sestává ze tří "kroků":
a) do registru C procesoru je uloženo číslo vyvolávané funkce BDOS,

b) podle druhu volané funkce je registrový pár DE naplněn adresou či nějakým znakem určeným k vydání,
c) následuje volání BDOS (CALL) na adresu 0005H.

Po vykonání funkce - podle jejího druhu - obdrží střadač A procesoru nějakou hodnotu nebo registr HL nějakou adresu. Dále pak BDOS přezkouší každé zadání z terminálu (konsoly) a rozpozná "CTRL C" (= "horký" start) a "CTRL P" (= printer off/on).

3. BIOS

Na rozdíl od předcházející je tato část systému CP/M závislá na technickém vybavení použitého mikropočítače. BIOS tedy v každém případě respektuje konfiguraci použitého prostředku a realizuje přizpůsobení dle specifických poměrů systému. BIOS začíná tabulkou zahrnující 17 rutin, jež jsou níže uvedeny (nemusí být vždy všechny zastoupeny):

1. COLD BOOT - tato rutina obstarává iniciaci systémových parametrů po zavedení systému
2. WARM BOOT - po stisknutí kláves "CTRL" a "C" je BDOS a CCP nově zaveden
3. CONSOLE STATUS - dotaz po stavu konsoly, zda bude následovat vydání znaku (rutina přezkušuje obsah střadače A)
4. CONSOLE INPUT - vložení znaku z konzoly (do registru A)
5. CONSOLE OUTPUT - vydání znaku z konsoly (uloženého v registru C)
6. LIST OUTPUT - výstup na tiskárnu (přes registr C), popřípadě na obrazovku konsoly (= terminálu)
7. PUNCH - registr C obsahuje znak k vydání, např. pro děrovačku pásky (případně vydání na obrazovce konsoly)
8. READER - vložení data z vstupního zařízení do střadače, např. z čtečky děrné pásky. (Rutiny 7. a 8. jsou uživatelským softwareem zřídka využívány; je však možné je použít pro komunikaci mezi dvěma počítači)
9. HOME - právě připojená disketová stanice hledá stopu 00
10. SELECT DISK - je vybírána jedna z možných (připojených) disketových stanic

11. SET TRACK - je volena určitá stopa diskety
12. SET SECTOR - je volen určitý sketor (úsek stopy) diskety
13. SET DMA ADDRESS - určuje oblast v uživatelské (operační) paměti RAM, z níž či do níž bude v následující disketové operaci psáno či čteno
14. READ - čtení z diskety do oblasti DMA, á 128 bytů
15. WRITE - vydání z oblasti DMA do diskety, á 128 bytů
16. LIST STATUS - dotaz o připravenosti tiskárny (ověřením obsahu střadače A)
17. SECTOR - převod logické pozice sektoru diskety do fyzické (reálné)

Organizace paměti provozního systému CP/M

Na obr. 54 je znázorněno uložení popisovaných částí provozního systému CP/M obecně v paměti mikropočítače. Nejdůležitější adresy jsou FBASE - označující počátek BDOSu - a TBASE, označující počátek operační paměti, stojící k dispozici uživateli, a to od adresy 100H. Prvních 256 bytů, neboli tzv. nultá stránka paměti (zero page - viz tabulku v příloze) je vyhrazeno pro systém CP/M, a sice následovně:

0 až 2H skok na rutinu WARM BOOT
3H obsahuje stavové slovo IOBYTE
4H číslo disketové stanice v provozu (počínajíc nulou)
5 až 7H skok do části BDOS; byty 6 a 7 obsahují adresu FBASE
8H RST 1
10H RST 2
18H RST 3
20H RST 4
28H RST 5
30H RST 6
38H RST 7 - je používán programem DDT, ostatní adresy restartů nejsou využity
5C až 7CH pro CCP k uložení tzv. FCB (= file control block) sestávajícího z 36 bytů, obsahujícího údaje o datovém souboru daného jména
80 až FFH pro CCP k použití jako vstupně-výstupní buffer datového souboru (DMA-buffer).

Tři části CP/M jsou umístěny v operační paměti počítače směrem od nejvyšší adresy (high mem.). Přesná hodnota adresy FBASE je závislá na rozsahu části BIOS a pochopitelně na celkové kapacitě paměti dat (min. 32 KB, 48 či 64 KB, apod.). Pro systém CP/M fy Digital Research pro zařízení MDS-Intellec-system (fy INTEL) je adresa FBASE (při 64KB operační paměti) EC00H; u zařízení HEATH-ZENITH, jehož CP/M verze 2.2.02 má BIOS o rozsahu 5 KB (opět při operační paměti 64 KB) nachází se FBASE na adrese DC06. Délka CCP je vždy stejná (800H), rovněž tak i BDOS (E00H).

Formát řídicího bloku souboru je na obr. 55; význam jeho jednotlivých bytů je následující:

ET byt 0 obsahuje číslo disketové stanice

FN byt 1 - 8 obsahuje až osmiznakové označení datového souboru (vlevo od dělící tečky). Není-li jméno zadáno, jsou slabiky vyplněny mezerami v kódu ASCII (= 20H)

FT byt 9 - 11 třímístné pole - vpravo od dělicí tečky označení datového souboru XYZ (extension), vyjadřující typ souboru, např.: .PAS (pascal), .MAC (makroassembler), .SUB (pro zásobníkové zpracování) apod.

EX byt 12 obsahuje údaj, o kolikátou část datového souboru se jedná v případě, že soubor dat je delší než 16 KB, (0 až 31)

00 byt 13 - 14 rezervovány pro interní potřeby systému, obvykle obsahují nulu

RC byt 15 vyjadřuje počet záznamů (records) obsažených v daném FCB

DM byt 16 - 31 obsahuje čísla jednokilobytových bloků, z nichž se datový soubor (nebo jeho část) na disketě skládá

NR byt 32 je použit částí BDOS pro započítání záznamů (records) při čtení či záznamu na disketě

RP byt 33 - 35 používají se systémem pro náhodné záznamy

Pomocné prostředky a řídicí znaky CP/M

Provozní systém CP/M poskytuje uživateli řadu pomocných prostředků = programů, jež je možno podle potřeby nahrát z diskety do operační paměti (TPA) daného mikropočítače, a pak využít k potřebným účelům. Jsou to:

ED.COM řádkově orientovaný textový editor

ASM.COM dvouprůchodový assembler pro intelovské 8080 standardní soubory ASCII. Produkuje šestnáctkové soubory v šestnáctkovém formátu Intel

LOAD.COM přeměňuje soubory v šestnáctkovém formátu Intel do provozuschopných strojově kódovaných programů

STAT.COM zobrazuje obsah diskety včetně názvu souborů, počtu záznamů, kB a zbývající kapacity diskety. Příklad výpisu, vyvolaného programem STAT je uveden v následující tabulce:

```
A > stat * *.
```

recs	Kbytes	ext	f.name
64	8	1	ASM.COM
38	5	1	DDT.COM
4	1	1	DUMP.COM
24	3	1	DUP.COM
82	11	1	ED80.COM
18	3	1	FORMAT.COM
238	30	2	KBASIC.COM
70	9	1	L80.COM
143	18	2	M80.COM
58	8	1	PIP.COM
41	6	1	STAT.COM
13	2	1	SYSGEN.COM
80	10	1	ZSID.COM

Bytes Remaining on A: 127 KB

DDT.COM odlaďovací program (debugger) s množstvím funkcí

SUBMIT.COM pomocný program k stohovému zpracování pod CCP

XSUB.COM v souvislosti se SUBMIT dovoluje stohové zpracování povelových zadání ve SUBMITem vyvolaných programech

PIP.COM program pro přenos datových souborů, kopírování, vydávání; jeden z nejvíce používaných

SYSGEN.COM tímto programem lze číst a popisovat systémové stopy diskety

MOVCPM.COM tento program umožňuje přizpůsobení systému CP/M na různě rozsáhlé (uživatelské) operační paměti RAM

K uvedeným souborům, které jsou zakončeny třímístným polem "COM", třeba zdůraznit, že je možné je přenést z diskety bez zadání extenze (tedy jen pouhým názvem), a dále, že mají charakter přímo proveditelných p o v e l ů ! (Vyvolaný soubor je uložen do TPA od adresy 100H a spuštěn.) Daný soubor povelů systému CP/M lze případně rozšířit o další datové soubory typu -.COM, jež si uživatel sám vytvoří či převezme.

K podpoře používání klávesnice terminálu v systému CP/M existují řídicí znaky (klávesy): tak klávesa RUBOUT maže posledně zadaný znak. Totéž dosáhneme stiskem kláves CTRL a H (nebo BACK-SPACE). Klávesami CTRL a U je zrušena celá řádka. Podobně pracuje CTRL-X, přitom však kursor (ukazovátko) je umístěno na začátek řádky. (Pozn.: tzv. řádka dat může mít až 255 znaků včetně mezer)

CTRL-U je určeno pro uživatele dálnopisu, rovněž tak CTRL-R. Posledně uvedenými řídicími znaky je vložený řádek informace opět vyslán.

CTRL-E způsobí uložení řídicího znaku CR (carriage return = návrat vozíku) do paměti, aniž by bylo ukončeno zadání povelů, které normálně jsou CR ukončeny

CTRL-J (linefeed) slouží též (mimo RETURN) k ukončení zadané řádky (pozn.: tzv. řádka dat může mít až 255 znaků včetně mezer)

CTRL-C způsobí nové "naloudování" systému CP/M. Tento "příkaz" se používá též při výměně disket

CTRL-Z označuje ukončení zadání z konsoly a je mnohými pomocnými programy, jako je PIP či ED, očekáván jako ukončující znak

CTRL-P připojuje tiskárnu paralelně ke konsoli. Tak lze protokolovat zadání a výsledky

CTRL-S zastavuje vydávání z konsole; jím je možno zastavit rychle probíhající "listing". Následným CTRL-S je vydávání opět uvolněno

Organizace diskety

Provozní systém CP/M je přizpůsobitelný pro různé formáty disket, jejich velikostí a organizací, jakož i jejich řadičů, popř. se již prodává na disketách různé velikosti (standard - 8", mini - 5 a 1/4", mikro - 3 a 1/2") a hustoty (SD - single density, DD - double density); tím je vyhověno potřebám uživatelů s odlišnými periferiemi pružných disků.

K představě čtenáře o organizaci na disketě bude v dalším komentováno standardní osmipalcové provedení diskety jednoduché zápisové hustoty (SD), programově sektorizované.

Osmipalcová disketa má 77 mezikružných nezávislých stop (track), každá stopa je dělena na 26 sektorů s kapacitou á 128 slabik - toto dělení odpovídá všeobecně rozšířenému standardu IBM 3740. Celková kapacita takovéto diskety tedy obnáší 77 x 26 x 128 = 256 256 bytů (neformátovaných), viz obr. 56. Pozice každého sektoru je určena indexovým otvorem, respektive jeho úhlovou odchylkou od jeho centrální spojnice, která označuje sektor 0. (Uvedené platí pro diskety s programově zajišťovanou orientací sektorů; u disket s hardwareovou orientací je pozice sektorů určena dalšími otvory - sector holes - které jsou při otáčení diskety postupně načítány, s počátkem u indexového otvoru.)

Každá disketa systému CP/M je "rozdělena" do tří odlišných oblastí, z nichž první má vyhrazenu stopu 00 a 01 následovně:

stopa 00 - sektor 1 : cold start loader
stopa 00 - sektor 2 až 17 : CCP
stopa 00 - sektor 18 až
stopa 01 - sektor 19 : BDOS
zbytek : BIOS
Druhou oblast tvoří pole přehledu souboru (file directory area), přičemž šestnáct sektorů stopy 02 je stále rezervováno pro přehled. Tato oblast může být ovšem odpovídajícími změnami v BIOSu rozšířena. Poslední - třetí oblast vytváří pak zbývající stopy pro ukládání dat a programů. Datový transport na a z diskety se děje po blocích á 128 slabik. Logicky na sebe vázané bloky souboru dat však nejsou ukládány v odpovídajícím sledu na disketu. Řadič diskety po každém bloku přezkušuje správnost přenosu; přiřazení logické pozice k fyzické se děje již zmíněnou funkcí 17 (SECTOR) BIOSu. Správa paměťových míst diskety je při CP/M dynamická, tzn., že v okamžiku otevření nového souboru nemusí být udána jeho možná délka. Paměťové místo je každému souboru přidělováno v "porcích" po 1 KB. Prostředky, jež CP/M ke správě diskety využívá, jsou:
1. řídicí blok souboru (FCB)
2. vyvolávací mapa bitů (ABM - allocation bit map)
3. seznam souborů (directory)

Šestnáct sektorů druhé stopy každé diskety obsahuje BDOSem spravovaný seznam datových souborů. Ten obsahuje každé první 32 slabiky (= byty) řídicích bloků (FCB´s) ke všem na disketě se nacházejícím souborům. Při každém novém volání diskety čte BDOS zmíněných 16 sektorů a spočítá z informace v části DM (v řídicím bloku souboru, viz jeho formát na obr. 56) zaplnění diskety. Informaci o zaplnění diskety přidá BDOS k mapě ABM a uloží ji do operační paměti RAM jako tabulku sestávající z 243 bitů a zobrazující stav zaplnění diskety.

Průběh otevření souboru na disketě modulem BDOS lze tedy popsat následovně:
a) BDOS prohledá ABM,až najde bit 0
b) nulový bit nahradí jedničkou a zanese číslo skupiny (1 až 243) do oblasti DM řídicího bloku FCB

c) před každou operací zápisu vypočítá BDOS z posledního skupinového čísla a následujícího čísla záznamu (RC) stopu a logický sektor, do nějž se má zapsat příští záznam
d) po zaplnění všech osmi sektorů jedné skupiny hledá BDOS následující nulu v mapě ABM a pak pokračuje podle b)

Z uvedeného plyne, že minimální rozsah paměťové kapacity pro jeden soubor je 8 sektorů á 128 bytů = 1 KB (i když třeba daný soubor sestává např. jen z šestnácti slabik). A protože BDOS stále prohlíží tabulku ABM odpředu, mohou logicky spolu související části jednoho souboru být fyzicky libovolně rozmístěny na disketě. Pokud tedy se nacházejí na disketě volné bloky (zatím nezaplněné či po nějakém zrušeném souboru), může je BDOS využít k zápisu bez ohledu na fyzické pořadí sektorů či stop. Z toho je zřejmé, že CP/M provádí veškeré přidělování sektorů a stop, aniž by k tomu požadoval součinnosti uživatele a aniž by jej tím zatěžoval - tedy zcela autonomně.

Jak již bylo výše naznačeno, je pro BIOS na obou systémových stopách k dispozici 7 sektorů, tedy necelý 1 KB. Pro BIOS menšího rozsahu to může být dostačující, avšak pro využití všech přizpůsobovacích možností je to přeci jen málo. Nicméně existují cesty k zabezpečení systému rozsáhlejším BIOSem. Tak např. CP/M verze 2.2 u fy Heath-Zenit vůbec neumísťuje BIOS na systémové stopy. Naproti tomu nahrazuje tzv. "Cold Start Loader" (zavlékací programová rutina "studeného" startu) zavlékacím programem pro BIOS (tzv. BIOS-Loader). Ten nejprve nahrává do uživatelské paměti RAM vlastní BIOS, nacházející se na disketě jako normální datový soubor. Dále pak - již pod kontrolou BIOSu - nahrává se do RAM zbývající část provozního systému CP/M, tj. BDOS a CCP. Tato metoda má tu výhodu, že pouhým vyměněním souboru BIOS lze dosáhnout zcela j i n é systémové přizpůsobení, aniž by byl nutný kontakt se systémovými stopami. Je ovšem třeba zdůraznit, že v daném řešení musí být soubor BIOS relokatibilní. Nicméně - naznačenou cestou - je možné uložit a používat libovolně rozsáhlý BIOS.

V.5. Sběrnice IEC

Tato sběrnice byla vyvinuta pro snadné propojení měřicích přístrojů s řídicími bez přídavných stykových obvodů. V USA je tato sběrnice normalisována pod názvem IEEE-488 Bus (Institute of Electrical and Electronic Engineers). Sběrnice IEC = IEEE-488 (u nás označována jako IMS-2, viz /8/, /16/ a /64/) je použita a některých mikropočítačů jako standardní přípojné místo. Přes toto normované místo může mikropočítač komunikovat s periferními přístroji, ovšem jen pokud jsou vybaveny totožným připojovacím místem.

Sběrnice IEC a IEEE-488 se liší jen korektory a počtem vývodů, viz obr. 57 , IEC je 25 žilná, zatímco IEEE-488 má o jedno zemnící vedení méně, tedy 24 žíly, tab. č. 23.

Na sběrnici může být maximálně připojeno 16 přístrojů dohromady. Sběrnice pracuje s negativní logikou, což znamená, že 0 V představuje signál log "1" a +5 V pak signál log "0" (neaktivní). Signály jsou TTL kompatibilní, proto jsou zde též povoleny jen krátké přívody, a sice 2 m na připojené zařízení (celkem maximálně 20 m vedení). Na sběrnici se dosahuje přenosové rychlosti až 500 Kbyte/s.

Ke sběrnici mohou být připojeny tři typy přístrojů: řídicí jednotka (řidič Ř), mluvčí M (talker) a posluchač P (listener). Řídicí jednotka vydává sběrnicové povely na ostatní přístroje a řídí celkový provoz na sběrnici. Může se sama nastavit do pozice mluvčího či posluchače. Mezi posluchače náleží zařízení, jež mohou přijímat data z jiných zařízení, např. tiskárna, obrazovkový displej. Mluvčí představují ty přístroje, jež mohou vysílat zprávy a informace, např. disketa či měřicí přístroje. Každé zařízení má pevnou adresu, přes níž může být dosaženo. Řídicí jednotka - obvykle mikropočítač - řídí přenos a rozhoduje, který přístroj smí na sběrnici vysílat.

Samotná sběrnice sestává z šestnácti vedení, z nichž osm je datových, tři řídící přenos dat (handshaking se signály DAV, NRFD, NDAC) a pět slouží k řídicím účelům (IFC, ATN,

SRQ, REN, EOI). Úlohy jednotlivých vedení jsou následující:

ATN (attention) - určuje, zda na sběrnici se nachází povely ("1") nebo data ("0")

IFC (interface clear) - způsobuje uvedení všech zařízení na sběrnici do definované úrovně (reset)

REN (remote enable) - umožňuje nastavení připojených přístrojů pro dálnopisný provoz ("1"). Při více řídicích jednotkách na sběrnici je REN aktivováno jen jednou

EOI (end of identifity) - má dvě funkce: jednak označuje konec přenosu dat (ATN=0), jednak signálem EOI - při ATN=1 - může řídicí jednotka zahájit dotazování k identifikaci přístrojů

SQR (service request) - je k dispozici všem přístrojům. Vedení je aktivováno vždy jen zařízením, žádajícím obsloužení řídicí jednotkou (M má zprávu pro Ř nebo P žádá informaci od Ř). Tehdy většinou přeruší řídicí jednotka průběh programu a zahájí sériové dotazování

DAV (data valid) - indikuje, že data na datové sběrnici jsou platná

NRFD (not ready for data) - signál, který je vyslán zařízením P zatím ještě nepřipraveným k převzetí dat

NDAC (no data accepted) - signál, vyslaný zařízením P, jež ještě nepřevzalo data

DIO1 až DIO8 - datová vedení, po nichž jsou transportována data nebo povely

Na obr. 58 je znázorněn přenos dat s kvitováním (handshaking) pomocí citovaných signálů DAV, NRFD a NDAC. Mluvčí nejdříve zkouší, zda jsou všichni posluchači připraveni. Pak jsou vložena data na sběrnici a signál DAV je aktivován (= 0). Nyní může přijímač (posluchač) převzít data, případně více posluchačů. Přijímače vyšlou nejprve NRFD = 0. Vysílač/mluvčí musí čekat, až přijímače - pracující různou rychlostí - převezmou data. To znamená až do toho okamžiku, kdy i ten nejpomalejší přístroj neaktivuje vedení NDAC (tj. NDAC = 1). Vysílač může nyní vedení DAV opět učinit neaktivní a vše se opakuje. Přenosová rychlost pak je dána nejpomalejším zařízením na sběrnici. Detailnější popis přesahuje rámec této práce, bližší např. v /16/.

U mnohých počítačů (Commodore, HP, atd.) může být sběrnice IEC velmi elegantně řízena instrukcemi jazyka BASIC, jako jsou: OPEN, INPUT # , PRINT # , CLOSE, neboť tímto jazykem jsou již výrobcem vybaveny.

V.6. Rozhraní RS-232C - V.24

Má-li kterýkoliv mikropočítač komunikovat s vnějším okolím, musí být schopen transportovat data z nejrůznějších periferních zařízení. Aby přístroje různých výrobců mohly být mezi sebou propojeny, musí přípojná místa odpovídat normovému předpisu, nebo musí být alespoň přibližně totožných vlastností.

EIA standard (Electronic industries association) RS-232 týká se standardizovaného schématu pro sériový přenos dat. Jeho podstatou je jednak použití stejného konektoru se vždy stejným rozmístěním funkčních vývodů, jednak použití stejných napěťových úrovní pro oba logické signály "0" a "1". RS-232C používá tedy pro signál logické nuly napětí v rozmezí +3 V až +15 V, pro signál logické jedničky -3 V až -15 V. (RS-232C je odvozen ze staršího standardu RS-232B, kde rozmezí pro oba logické signály byla širší, a sice pro logickou nulu +5 až +25 V, pro logickou jedničku -5 až -25 V, viz obr. 59.) Oblast -3 V až +3 V je přechodovou nepoužívanou oblastí.

Modemy, obrazovkové terminály a monitory, některé typy tiskáren a dálnopisů (např. model 43) jsou vybaveny z výše uvedeného důvodu pětadvacetižilným konektorem včetně příslušného stykového zapojení RS-232C. Z hlediska druhu a rychlosti přenosu budiž řečeno, že RS-232C je rozhraním pro asynchronní sériový přenos dat, kde jednotlivé znaky jsou přenášeny jako sled osmi bitů uvedených vždy jedním startovacím bitem s nulovou úrovní a ukončených jedním nebo dvěma stopbity s jedničkovou úrovní, viz obr. 60. Používané přenosové rychlosti se pohybují od 50 bitů.s^{-1} až do 19 200 bitů.s^{-1}.

Označení jednotlivých vývodů konektoru je vyznačeno v tabulce č. 24. Z možných dvaceti pěti vývodů mají největší

důležitost vývody č. 1 - zemnění přístroje, č. 2 - výstup, č. 3 - sériový vstup a č. 7 - zemnění signálu.

Tabulka č. 24 - Označení vývodů a jejich funkce konektoru DB-25 pro rozhraní RS-232C/V.24

Vývod č.	Označení		Funkce
	EIA RS-232C	CCITT V.24	
1	AA	101	zemnění přístroje
2	BA	103	výstup vysílaných dat
3	BB	104	vstup přijímaných dat
4	CA	105	RTS (request to send) - požadavek vysílání
5	CB	106	CTS (clear to send) - připravenost k vysílání
6	CC	107	DSR (data set ready) - připravenost k provozu
7	AB	102	signal ground - zemnění signálu
8	CF	109	carrier detector - detekce úrovně přijímaného signálu
9			nedefinováno
10			nedefinováno
11	CK	126	select transmit frequency (200 Bd modem) - volba přenosové frekvence
12	SCF	122	secondary carrier detector - úroven signálu přídavného kanálu
13	SCB	121	secondary clear to send - připravenost k vysílání příd. kanálu
14	SBA	118	secondary transmitted data - výstup přídavného kanálu
15	DB	114	transmitter signal element timing, transmit clock from modem DCE - vysílací takt
16	SBB	119	secondary received data - vstup přídavného kanálu
17	DD	115	receiver signal element timing - přijímací takt
18			nedefinováno
19	SCA	120	secondary request to send - požadavek vysílání přídavného kanálu
20	CD	108.2	DTR (data terminal ready) - terminal připraven k provozu
		108.1	connect data set to line - připojit přenosové vedení
21	CG	110	signal quality detector - jakost příjmu

22	CE	125	ring indicator - přicházející volání
23	CH	111	DTE (data signal rate selector) - volba přenosové rychlosti
24	DA	113	transmitter signal element timing - přenosový takt k modemu DTE
25			nedefinováno

Z tabulky je patrná shoda mezi americkým standardem RS-232C a evropským V.24 (CCITT); pro V.24 jsou některé funkce pojaty do normy, zatímco u RS-232C nejsou definovány (funkce vývodů 11, 12, 13, 14, 16, 19, 21 a 23), dále pak se liší pouze označením.

Pomocí vývodu č. 4 sděluje mikropočítač modemu, že je připraven k přenosu dat, vývod č. 5 pak signalizuje připravenost modemu k převzetí dat. Někdy se pro tento účel používá vývod č. 6 (např. u tiskáren). Vývody č. 6, 8 a 20 se používají k odpovídajícímu propojení s univerzálním sériovým stykovým obvodem UART.

Vedením, připojeným k vývodům č. 2, 3, 4, 5 - včetně nezbytného zemnícího č. 7 - lze jednoduše realizovat styk s potvrzením (handshake): je-li přijímač (např. tiskárna) připraven k převzetí dat, vyšle na vývod č. 5 signál log "1", pak může počítač reagovat vysláním jednoho znaku. Opačně indikuje počítač signálem jedničkové úrovně na vývodu č. 4 svou připravenost k příjmu dat.

Obvodová technika přeměny signálu úrovně RS-232C na úroveň TTL a opačně je poměrně jednoduchá. Tak na obr. 61a je zapojení přijímače signálu RS-232C; signál je omezen odporem R_1 a jeho záporná půlvlna je zkratována diodou D_1. Tranzistor T_1 signál dále invertuje a přizpůsobuje následujícím obvodům TTL. Druhá polovina obrázku, 61b, zachycuje alternativní zapojení vstupního přijímače s jedním invertorem běžného obvodu TTL. Rezistor R^+ (vyznačen čárkovaně) je potřebný jen tehdy, je-li tento stykový člen buzen optickým vazebním obvodem (např. WK 164 12 či WK 164 13 apod.).

Zapojení vysílací části je zachyceno taktéž ve dvou alternativách. V té první - obr. 62a - je dvoutranzistorový převodník úrovně vysílaného signálu napájen ze zdroje +12 V a -12 V, což jsou napětí, jež obvykle v mikropočítačových zařízeních staršího data bývají k dispozici. Druhá alternativa - s menším rozkmitem vysílaného signálu RS-232C, je napájena +5 V a -12 V. Vzhledem k tomu, že propojovací vedení může dosahovat délky řádu desítek metrů, a proto budicí (vysílací) strana musí krýt případné ztráty a též být odolná proti zkratu při nízké výstupní impedanci, vyhovuje tomuto účelu lépe tato verze. Její výstupní signál může přes diodu D3 - v případě potřeby - budit diodu LED optického vazebního členu, ovšem po zvětšení rezistoru Rc na 15 ohmů. Zapojení umožňuje (právě díky napájení ze zdroje kladného napětí jen +5 V) připojením diody D4 zrušit funkci transponování úrovně vysílaného signálu, a tedy zachování úrovně TTL. (Dioda D4 omezuje výstupní napětí na -0,6 V; bez ní je tedy rozkmit výstupního signálu od +5 V do -12 V.) V poslední době se používá pro realizaci převodníku úrovně RS-232C speciálních integrovaných obvodů, např. typů TL 188 a 189, 1489 a 1488 nebo 75189 a 75188.

Proudová smyčka 20 mA (TTY)

Proudová smyčka představuje jedno z nejstarších sériových rozhraní. Používala se (používá se) k řízení dálnopisů (teletype = TTY) a starších typů tiskáren. Uspořádání sériového sledu bitů odpovídá tvaru vyznačenému v obr. 60 pro RS-232C. Nacházíme zde dva páry vedení, jeden pro vysílání, druhý pro příjem. Signál logické jedničky je představován proudem 20 mA, logické nuly žádným nebo proudem 4 mA.

Zapojení přijímací a výstupní části rozhraní je na obr. 63a, přičemž je běžné, že potřebný proud je dodáván mikropočítačem, a připojená periférie je pasívní (tzn. odebírá proud). Při propojení dvou počítačů nebo aktivních zařízení může dojít ke kolisím, zvláště je-li rozhraní tranzistorové. Odstra-

nění problémů poskytuje zapojení styku pomocí optických vazebních členů, umožňujících spolehlivé propojení do délky vedení cca 20 m, viz obr. 63b.

Rozhraní CENTRONICS

Rozhraní CENTRONICS je paralelní rozhraní určené k připojení tiskáren, které bylo navrženo výrobní firmou CENTRONICS, a které během času se rozšířilo po celém světě a stalo se tak neoficiálním standardem.

Toto rozhraní pracuje s napěťovými úrovněmi logiky TTL, z čehož vyplývá, že délka přípojného vedení nesmí překročit dva metry. Každé signálové vedení dále má vlastní zemnicí spoj, s nímž tvoří - po stočení - samostatnou zkroucenou dvojici (twisted pair). Pro připojení tiskárny se běžně používá 36 pólového konektoru (fa Amphenol, typ 57-30360), viz obr. 64. Obložení vývodů je jen částečně normalizováno; avšak vždy jedna řada kontaktů (1 až 18) má připojena signálová vedení, zatímco druhá (kontakty 9 až 36) pak zemnící spoje, stáčené se signálovými do jednotlivých párů (které jsou slepeny do plochého kabelu). V následující tabulce je vyznačeno funkční obložení jednotlivých vývodů konektoru, tak jak je obvyklé např. u renomovaného výrobce tiskáren EPSON. Přitom označení funkcí vývodů 1 až 11 a 16 je totožné u všech známých druhů tiskáren s paralelním vstupem. Přičlenění funkcí ostatním vývodům konektoru se však může dle výrobců lišit.

Vedení $\overline{\text{STROBE}}$, $\overline{\text{BUSY}}$, a $\overline{\text{ACKNLG}}$ jsou určena pro styk s potvrzením (handshake protocoll). Ten probíhá následovně, viz obr. 65. Počítač vloží data na vedení DATA 1 až DATA 8 a pak vedení $\overline{\text{STROBE}}$ krátce nuluje. Klesající hranou signálu $\overline{\text{STROBE}}$ je vybuzen signál BUSY, který zůstane tak dlouho aktivní (v jedničkové úrovni), až je tiskárna připravena opět převzít data. Vedení $\overline{\text{ACKNLG}}$ (acknowledge) přejde krátkodobě do nuly po zpracování dat tiskárnou. Klesající hranou $\overline{\text{ACKNLG}}$ je ukončena aktivita signálu $\overline{\text{BUSY}}$, jehož klesající hranou je zas vyvolán přechod $\overline{\text{ACKNLG}}$ do jedničkové úrovně. Pak může být vyslán další znak (jako nová data) do tiskárny a děj se opakuje.

Tabulka č. 25 - Označení vývodů a jejich funkce konektoru Amphenol 57-30360 pro rozhraní CENTRONICS

Označení párových vývodů		Signál	Směr přenosu	Význam
1	19	STROBE	vstup	aktivní v nule, realizující předání dat do tiskárny, šíře pulsu větší 0,5 µs
2	20	DATA 1	vstup	tyto signály representují osmibitovou slabiku dat na datové sběrnici
3	21	DATA 2	vstup	
4	22	DATA 3	vstup	
5	23	DATA 4	vstup	
6	24	DATA 5	vstup	
7	25	DATA 6	vstup	
8	26	DATA 7	vstup	
9	27	DATA 8	vstup	
10	28	ACKNLG	výstup	impuls trvání cca 12 µs, aktivní v nule, oznamující, že data byla převzata a že tiskárna je připravena pro další příjem
11	29	BUSY	výstup	pokud je BUSY aktivní (= log.1), není tiskárna připravena k převzetí další slabiky Busy = 1 pokud jsou přebírána data, během tisku, v modu OFF-LINE, popř. při chybě
12	30	PE	výstup	signál indikující ukončení papíru (paper-end)
13	-	-	-	jedničkový (+5 V přes rezistor 3,3 kΩ)
14	-	AUTO FEED XT	vstup	při 0 = automaticky nová řádka při 1 = aut. řádkování vyřazeno
15	-	NC	-	nepoužit (no connection)
16	-	OV	-	nulová úroveň logiky (zemnění)
17	-	GND	-	zemnění zařízení, (isolované od vývodu 16 - zemnění logiky)
18	-	NC	-	nepoužit
19až30	-	GND	-	zemnění párů 1 až 12
31	-	INIT	vstup	aktivní v nule, při trvání minimálně 50 µs iniciuje tiskárnu
32	-	ERROR	výstup	aktivní v nule, je-li tiskárna 1) OFF-LINE 2) při ukončení papíru 3) při ev. chybě
33	-	GND	-	zemnění
34	-	NC	-	nepoužit
35	-	-	-	jako 13
36	-	SLCT IN	vstup	volba tiskárny (DC1 až DC3 jsou funkční jen při 36 = "1")

V.7. Paralelně-sériový převodník

Pro převod osmibitového paralelního slova (slabiky) do sériového formátu dle standardu RS-232 C/V.24 existují speciální integrované obvody, jako je m.j. UART MHB 1012 (universální asynchronní přijímač - vysílač) a další. Takovýto obvod, umožňující nejen asynchronní převod paralelního formátu do sériového, ale současně i sériového do paralelního, nebývá vždy dostupný a dokonce v některých aplikacích i využitý v obou druzích převodu. Diskrétní zapojení je nejen levnější, ale mnohdy lépe využitelné.

Následující obr. 66 přináší zapojení jednoduchého paralelně-sériového převodníku, osazeného třemi integrovanými obvody a jedním tranzistorem, s možností volby úrovně výstupního signálu TTL či RS-232 C. Takovýto převodník může najít použití např. u stávající klávesnice s paralelním výstupem, již je třeba v daném případě připojit k mikropočítači se sériovým vstupem, u tiskárny se sériovým vstupm V.24 apod.

Zapojení pracuje následujícím způsobem: synchronizační puls "strobe" z klávesnice budí klopný obvod $IO_1/2$ - 74LS74. Nulová úroveň jeho výstupu N (č. 8) přivedená na vstup "shift/load" (č. 1) registru 74LS165 způsobí převzetí paralelní informace ze vstupů A až H, tj. D0 až D6, přičemž D7 - vzhledem k uzemnění H - je vždy nulový. To odpovídá - v sériovém formátu po převodu - předepsané nulové úrovni startovacího bitu. Vlastní konverse se realisuje v registru, a to v rytmu hodinového kmitočtu přiváděného na vstup "clock" (vývod č. 2). Kmitočet - odpovídající požadované přenosové rychlosti b/s - je dán součinem hodnot P_1 a C_2, přičemž změnou P_1 (tvořeného potenciometrovým trimrem víceotáčkového cermetového provedení) lze jej přesně nastavit. Kmitočtová stálost je ovšem dána jednak vynikajícími vlastnostmi časovače typu 555, jednak teplotními koeficienty kondenzátorů C_2 a C_3. Pokud je požadováno, lze změnit přenosovou rychlost změnou kapacity kondenzátoru C_2; v daném případě odpovídá jeho hodnota - spolu s hodnotou P_1 - přenosové rychlosti 300 bitů/s = 300 Bd.

Hodinový výstup z časovače je též připojen na vstup druhého klopného obvodu (vývod č. 3 obvodu 1/2 74LS74). Ten přijímá sled transponovaných bitů, jež se též sériově objeví na jeho výstupu Q (č. 5), odkud budí tranzistor T_1. K mazacímu vstupu č. 1 klopného obvodu je připojen RC člen složený z odporu R_1 a kondenzátoru C_1, jenž zajišťuje definovaný stav klopného obvodu po připojení napájecího napětí +5 V. Tento klopný obvod zajišťuje dále, že startovací bit má vždy stejnou šířku jako ostatní bity signálu.

Paralelně-sériový převod začíná při nástupní (kladné) hraně strobovacího impulsu. Tím je generován nahrávací impuls "load" z výstupu N (č. 8), jehož trvání je vlivem zpětné vazby z výstupu posuvného registru Q (č. 9) do mazacího vstupu klopného obvodu R (č. 13) velmi krátké. (Tato skutečnost klade vysoké nároky na jakost strobovacího signálu, neboť jakékoliv zakmitání či hazardní špičky by vyvolaly komplikace převodu.)

Z kolektoru tranzistoru T_1 se snímá sériový signál s jedním startovacím bitem, sedmi datovými bity a jednotkovou úrovní stop-bitů pro každý znak. Vzhledem k tomu, že hodinový signál je stále přítomen, je na výstupu úroveň stop-bitu potud, pokud není vysílán další (nový) znak. Pomocí vstupu "break" je konečně možné připojit i shodně označené tlačítko klávesnice. Na obr. 67 je vyznačen průběh signálů ve vyznačených bodech.

Obdobné zapojení, umožňující však navíc změnu polarity synchronisačního signálu "strobe" a zadání i osmého bitu datového slova, je nakresleno na obr. 68. Na rozdíl od předcházejícího je zde použito dvou posuvných registrů, zapojených za sebou. Generátor taktu je zde opět realisován stabilním časovačem typu 555; členy, určující kmitočet, jsou opět kondenzátor C_2 a potenciometrový víceotáčkový trimr P_1. V daném případě odpovídají jejich hodnoty přenosové rychlosti 9600 bitů.s^{-1}; pochopitelně lze použít i jinou přenosovou rychlost.

Posuvné registry - 2x 74LS165 - se nacházejí normálně v provozním stavu "posun" (shift); ten opouští pouze v okamžiku ukládání paralelních dat (load), vyvolaného změnou úrovně na vstupech č. 1 obou registrů, a to synchronisačním impulsem "strobe". Tento impuls je přiváděn přes hradlo XOR, čímž je možno snadno dosáhnout změny jeho polarity v případě, že zdroj dat (např. klávesnice) generuje synchronisační impulsy v invertované formě. Pak stačí - pro dosažení požadované polarity - jen přepnout spínač S_1 do druhé polohy.

Transponovaný signál je odebírán z výstupu N (vývod č. 7 druhého registru) a přes oddělovací rezistor R_1 přiváděn na dvojici tranzistorů T_1 a T_2, která upravuje signál do úrovně RS-232C. Sériový signál začíná startovacím bitem nulové úrovně, po něm následuje sedm bitů pro vyjádření znaků ASCII. Osmý bit (paritní) může být v případě potřeby vložen přes vstup "break"; jinak má nulovou úroveň. Následující stop-bity jsou všechny jedničkové.

Pro eventuální aplikaci je uveřejněn i obrazec plošných spojů tohoto užitečného převodníku, jakož i pohled na rozložení součástí, obr. 69. Obvod IO_3 je osazen typem 74LS86, což je čtyřnásobný dvouvstupový člen pro realizaci exklusivního součtu. Jsou z něj však využity jen dva členy pro výše zmíněné funkce.

V.8. CMOS-RAM jako simulátor paměti EPROM

Zřejmě nebude třeba zvlášť zdůrazňovat, že jakýkoli mikropočítač či mikroprocesorový systém - byť i jen jednoúčelový - nemůže pracovat bez programu, ať již provozního (= monitor) či uživatelského (sestaveného jen pro řešení dané úlohy nebo pro řízení daného procesu). Program takovéhoto druhu bývá obvykle uložen v pevné paměti EPROM; jeho vývoj se děje přes návrh blokového schématu až do podrobného rozepsání algoritmu v assembleru či přímo ve strojovém kódu. Pokud je k dispozici tzv. vývojový systém, lze takto rozepsaný program "ladit", tzn. testovat spolu s odstraňováním případných syntak-

tických, popř. i sémantických chyb. Není-li však vývojový systém disponibilní, následuje přehrání rozepsaného programu - pomocí programátoru - do pevné paměti EPROM. Ukáže-li se však, že nahraný program nepracuje bezvadně nebo dle představ uživatele (nebo nepracuje vůbec, což je též možné), následuje vymazání paměti EPROM ultrafialovým světlem ve speciálním zařízení k tomu účelu koncipovaným - a po opravách algoritmu - nové nahrání v programátoru.

Obě etapy, tj. mazání i nahrávání pamětí EPROM - nemusí být vždy pro potenciálního uživatele nejschůdnější, a to z celé řady důvodů: programátor vývojově starších typů EPROM (2708 a 2704) nahrává daný obsah 50x se opakujícím pulsem o trvání max 1 ms, zatímco novější vícekapacitní paměti EPROM 2758 a 2716 vyžadují pro každý bit pouze jeden impuls, avšak o trvání 50 ms, paměti 2732 pak navíc impuls opačné polarity; programátory pro odlišné typy pamětí musí tedy respektovat zpravidla odlišné údaje výrobce. Navíc ne každý programátor je univerzální a je schopen nahrávat i nejnovější (velkokapacitní) typy (2764, 27128) - třeba jen proto, že v době jeho vzniku tyto ještě nebyly na trhu. Rovněž tak doba mazání (osvitu ultrafialovým světlem) jakož i intenzita osvitu se liší i u stejných typů různých výrobců; zpravidla se pohybuje kolem 20 minut, u těch "tvrdších jnějších" činí doba ozáření (pro dosažení obsahu FFH u každého paměťového místa) až dvojnásobek.

Níže popsané zapojení poskytuje možnost odstranění případných komplikací, a to nahrazením pevné paměti EPROM úspornou (příkonově) pamětí RAM; tato - vzhledem k zanedbatelnému odběru (0,5 μA) - může být dlouhodobě napájena z vlastního chemického zdroje. Tím je zajištěna proti ztrátě obsahu při vypnutí mikropočítače - pracuje tedy jako simulátor EPROM. Nicméně lze ji pochopitelně používat též jako RAM.

Navržené zapojení je uvažováno pro realizaci samostatného přípravku, který se připojuje k mikropočítači prostřednictvím 24pólové zástrčky, a to do patice určené pro paměť EPROM, včetně nezbytného vícežilového kablíku, obr. 70. Vlastní paměť

C-MOS, typ HM 6116-LP3 se nachází se svou paticí, diodou D, napájecí baterií, přepínačem a nezbytnými rezistory na malé destičce opatřené příslušnými plošnými spoji. Přepínač umožňuje - ve své poloze 1 - ochranu zápisu, takže program z paměti může být pouze čten, nikoli však přepsán. V poloze 2 - propojením vývodu "21" paměti s vedením signálu R/$\overline{W}$ - lze pochopitelně jak číst, tak i zapisovat.

Vzhledem k tomu, že použitá paměť má kapacitu 2 KB, může nahradit EPROM typu 2716 o téže kapacitě. Též je výhodné, že nabíjení zdroje - je-li použito např. niklkadmiových miniaturních článků - obstarává mikropočítač, do jehož patice EPROM je přípravek vložen. Aby však se akumulátor při vypnutí mikropočítače nevybíjel, je mu předřazena dioda D. Rezistor R23 omezuje nabíjecí proud akumulátoru na cca 5 mA. Zabezpečení proti statickým nábojům, jež by mohly ohrozit statickou paměť C-MOS, mají za úkol rezistory R1 až R22. Aby bylo dosaženo kompatibility s mnohdy odlišným zapojením špiček č. "18" a "20" patice EPROM daného mikropočítače, je u těchto dvou přípojů v přípravku pamatováno na možnost obměny prostřednictvím propojek. (Někdy je jeden z těchto vývodů spojen přímo se zemí, někdy jsou propojeny paralelně.)

Simulátor lze výhodně používat např. při plánovaném rozšiřování provozního monitoru, např. o diagnostické rutiny apod., kdy lze p o s t u p n ě , po nemnoha bytech odlaďovat jednotlivé podprogramy. A v tom tkví právě jeho neocenitelná přednost.

Nahrávat lze buď v jiném mikropočítači pouhým umístěním přípravku do odpovídající patice s 24 vývody - nebo přímo ve vyvíjeném systému, a to způsobem přímého přístupu do paměti (DMA) /45/.

V.9. Programátor pamětí

V následujícím bude pospán jednoduchý programátor pamětí, použitelný jak pro běžné typy 2758 a 2716, tak i pro velkokapacitní 2732 (4 KB). Na obr. 71 je zakresleno jeho zapojení:

spínači SA0 až SA10 (SA11) nastavuje se binárně adresa jednoho každého paměťového místa. Na svítivých diodách LD 0 až LD 7, umístěných za oddělujícími invertory (7404), lze kontrolovat binárně obsah jednotlivých buněk před a po nahrání.

Před vlastním nahráváním žádaného programu umístí se daná EPROM do příslušné patice (v zahraničí existují patice speciálně vyvinuté pro tento účel, které se vyznačují "nulovou silou" nutnou pro vložení či vyjmutí; to proto, že pákovým mechanismem lze kontaktní svírky rozevřít). Nyní se přepínačem Př. 1 nastaví použitý typ a pak již následuje nastavení osmibitové slabiky spínači SD0 až SD7. Po nastavení adresy a data následuje stisknutí tlačítka PROG, umístěného ve vstupu klopného obvodu RS, jehož výstupním signálem je spuštěn monostabilní klopný obvod 74121. Ten na svém výstupu generuje impuls definované délky, tj. doporučených 50 ms. (Jeho délka je dána hodnotami Cp a Rp. Ještě před uvedením do provozu programátoru je nutné tuto prodlevu prověřit - nejlépe pomocí osciloskopu. Příliš dlouhý impuls by totiž mohl vést ke zničení paměti EPROM, příliš krátký pak k nenahrání či brzké ztrátě požadované informace.) Tímto impulsem se tedy do paměti EPROM nahraje vždy jedna slabika (= 1 byte). Ještě je nutno připomenout, že otevřená poloha spínačů dat a adresy odpovídá signálu "log 1", uzavřená pak "log 0". Tak je možné postupně nahrát na požadované adresy paměti předepsaná data programu.

Přepnutím přepínače Př. 2 z polohy "pgm" pro programování do polohy "rd" lze naprogramovaná data po adresách číst - ovšem nyní za nezbytného předpokladu, že všechny spínače SDi jsou otevřené. Pokud bylo použito paměti EPROM nové nebo s vymazaným obsahem (osvícením ultrafialovým světlem v mazacím zařízení), musí být obsah každé buňky roven "log.1", tzn., že každá slabika má obsah FFH. Pokud by tomu tak nebylo - což je možné namátkově kontrolovat popisovaným programátorem při poloze přepínače Př. 2 v poloze "rd" - pak nebylo mazání dostačující, nebo je paměť vadná. Takovou pochopitelně nemá cenu

nahrávat novým programem; je proto vždy užitečné zkontrolovat alespoň část obsahu disponibilní paměti ještě před zahájením časově náročného ručního programování.

Na dalším vyobrazení jsou znázorněny časovací průběhy impulsů při programování dvoukilobyteových (2716) a čtyřkilobyteových (2732) pamětí; průběhy obou druhů se odlišují, obr. 72. U paměti 2716 - sdružený přepínač Př. 1 v dolní poloze - je před zahájením programovacího průběhu uložena úroveň "log. 1" na vstup $\overline{OE}$, čímž jsou výstupní budiče paměti uzavřeny (vývod č. 20). Na vstup V_{pp}, na němž se normálně nachází +5 V, je přiloženo současně programovací napětí cca +25 V, které za žádných okolností nesmí překročit hodnotu 26 V - vývod č. 21. Uvolňující vstup $\overline{CE}$/PGM (vývod č. 18) leží zatím na nule, a teprve po stisku programovacího tlačítka obdrží úroveň +5 V na dobu 50 ms, čímž je ukončeno nahrání jedné slabiky.

U paměti 2732 - přepínač Př. 2 v horní poloze - je programování odlišné. Vzhledem k tomu, že vývod č. 21 je vstupem pro adresní vedení A11, je zde V_{pp} sloučen s $\overline{OE}$ (č. 20). Je-li tedy $\overline{OE}$ na nule, jsou uvolněny datové výstupy pro čtení; je-li však na tento vývod přiloženo napětí +25 V, je obvod připraven k programování. Současně však musí vstup $\overline{CE}$ nést jako klidový signál úroveň +5 V, která - po stisku programovacího tlačítka PROG - je na dobu 50 ms změněna na 0 V. Z uvedeného je patrné, že polarita programovacího pulsu 50 ms je pro 2732 právě opačná než pro 2716! Čtyřkilobyteové paměti se tedy programují záporným pulsem, odebíraným z výstupu "1" obvodu 74121.

V.10. Školní mikropočítač

Závěrem této části uvádíme zapojení jednoduchého jednodeskového mikropočítače, určeného pro školící účely, jehož sestava je na obr. 73. Mimo mikroprocesor Z80-- IO_1 je osazen ještě deseti integrovanými obvody, nutnými k jeho funkci (IO_2 až IO_{11}), sedmi tranzistory sedmisegmentového osmimístného displeje (T_1 až T_7) včetně nezbytné 24 tlačítkové klávesnice.

Indikace adresy, dat a případných sdělení obsluze je prováděna na zmíněném displeji, kde jednotlivé segmenty jsou prostřednictvím kanálu PA stykového obvodu 8255 (IO_{10}) propojovány s datovou sběrnicí. Tak je možno vytvářet na displeji i stilisovaná písmena sdělení, jako např.: Error, run apod. Rovněž tlačítka jsou propojována přes kanál PC a klíčovací obvod IO_{11} s datovou sběrnicí; jsou tedy spolu s ukazateli (8x TIL 702) periodicky ošetřovány mikroprocesorem, pochopitelně na základě vhodného programového vybavení. Kanál PB je volný - k dispozici uživateli. Kapacita programové paměti EPROM (pětivoltového typu 2758) činí 1 KB, stejnou kapacitu má i paměť dat RAM (2x 2114). Selekci čipů zajišťuje IO_5 (3205 = 74LS138) při neúplném dekódování. Řídicí signály $\overline{IOWR}$, $\overline{IORD}$, $\overline{MEMW}$ a $\overline{MEMR}$ jsou získávány z CPJ prostřednictvím jednoduché logiky obvodem IO_8.

Generátor taktu je běžného typu a je osazen obvodem MH74S04, dva invertory tohoto obvodu slouží též k úpravě resetovacího signálu pro CPU a - v opačné polaritě - i pro stykový obvod 8255. Pomocí IO_7 je možné - vzhledem k použitému krystalu a typu CPJ - volit poloviční či čtvrtinový kmitočet pro takt. Paměť dat je možné bezproblematicky rozšířit o 6 KB, přičemž výběr čipů pak zajišťují signály $\overline{CS}_1$ až $\overline{CS}_6$. Pro rozšíření je vhodné použít pamětí C-MOS, zálohovaných z vlastního chemického zdroje, a to pro uchování vložených dat zkušebních programů. Pak odpadne někdy nejisté nahrávání a zpětné čtení z magnetofonu.

Příloha 1

ROZDĚLENÍ ŠESTNÁCTIKILOBYTOVÉ PAMĚTI PO STRÁNKÁCH++

nultá 0000-00FF	první 0100-01FF	druhá 0200-02FF	třetí 0300-03FF	1. KB
čtvrtá 0400-04FF	pátá 0500-05FF	šestá 0600-06FF	sedmá 0700-07FF	2. KB
osmá 00800-08FF	devátá 0900-09FF	desátá 0A00-0AFF	jedenáctá 0B00-0BFF	3. KB
dvanáctá 0C00-0CFF	třináctá 0D00-0DFF	čtrnáctá 0E00-0EFF	patnáctá 0F00-0FFF	4. KB
šestnáctá 1000-10FF	sedmnáctá 1100-11FF	osmnáctá 1200-L2FF	devatenáctá 1300-13FF	5, KB
dvacátá 1400-14FF	jednadvacátá 22. 1500-15FF	1600-16FF	třiadvacátá 1700-17FF	6. KB
24. 1800-18FF	25. 1900-19FF	26. 1A00-1AFF	27. 1B00-1BFF	7. KB
28. 1C00-1CFF	29. 1D00-1DFF	30. 1E00-1EFF	31. 1F00-1FFF	8. KB
32. 2000-20FF	33. 2100-21FF	34. 2200-22FF	35. 2300-23FF	9. KB
36. 2400-24FF	37. 2500-25FF	38. 2600-26FF	39. 2700-27FF	10. KB
40. 2800-28FF	41. 2900-29FF	42. 2A00-2AFF	43. 2B00-2BFF	11. KB
44. 2C00-2CFF	45. 2D00-2DFF	46. 2E00-2EFF	47. 2F00-2FFF	12. KB
48. 3000-30FF	49. 3100-31FF	50. 3200-32FF	51. 3300-33FF	13. KB
52. 3400-34FF	53. 3500-35FF	54. 3600-36FF	55. 3700-37FF	14. KB
56. 3800-38FF	57. 3900-39FF	58. 3A00-3AFF	59. 3B00-3BFF	15. KB
60. 3C00-3CFF	61. 3D00-3DFF	62. 3E00-3EFF	63. 3F00-3FFF	16. KB

++1 stránka = 256 bytů, 4 stránky = 1024 bytů = 1 KB

Příloha 2

ROZDĚLENÍ ŠEDESÁTČTYŘKILOBYTOVÉ PAMĚTI PO KILOBYTECH
V ŠESTNÁCTKOVÉM (HEX) A DESÍTKOVÉM VYJÁDŘENÍ

1	0000-03FF 0-1023	2	0400-07FF 1023-2047	3	0800-0BFF 2048-3071	4	0C00-0FFF 3072-4095
5	1000-13FF 4096-5119	6	1400-17FF 5120-6143	7	1800-1BFF 6144-7167	8	1C00-1FFF 7168-8191
9	2000-23FF 8192-9215	10	2400-27FF 9216-10239	11	2800-2BFF 10240-11263	12	2C00-2FFF 11264-12287
13	3000-33FF 12288-13311	14	3400-37FF 13312-14335	15	3800-3BFF 14336-15359	16	3C00-3FFF 15360-16383
17	4000-43FF 16384-17407	18	4400-47FF 17408-18431	19	4800-4BFF 18432-19455	20	4C00-4FFF 19456-20479
21	5000-53FF 20480-21503	22	5400-57FF 21504-22527	23	5800-58FF 22528-23551	24	5C00-5FFF 23552-24575
25	6000-63FF 24576-25599	26	6400-67FF 25600-26623	27	6800-6BFF 26624-27647	28	6C00-6FFF 27648-28671
29	7000-73FF 28672-29695	30	7400-77FF 29696-30719	31	7800-7BFF 30720-31743	32	7C00-7FFF 31744-32767
33	8000-83FF 32768-33791	34	8400-87FF 33792-34815	35	8800-8BFF 34816-35839	36	8C00-8FFF 35840-36863
37	9000-93FF 36864-37887	38	9400-97FF 37888-38911	39	9800-9BFF 38912-39935	40	9C00-9FFF 39936-40959
41	A000-A3FF 40960-41983	42	A400-A7FF 41984-43007	43	A800-ABFF 43008-44031	44	AC00-AFFF 44032-45055
45	B000-B3FF 45056-46079	46	B400-B7FF 46080-47103	47	B800-BBFF 47104-48127	48	BC00-BFFF 48128-49151
49	C000-C3FF 49152-50175	50	C400-C7FF 50176-51199	51	C800-CBFF 51200-52223	52	CC00-CFFF 52224-53247
53	D000-D3FF 53248-54271	54	D400-D7FF 54272-55295	55	D800-DBFF 55296-56319	56	DC00-DFFF 56320-57343
57	E000-E3FF 57344-58367	58	E400-E7FF 58368-59391	59	E800-EBFF 59392-60415	60	EC00-EFFF 60416-61439
61	F000-F3FF 61440-62463	62	F400-F7FF 62464-63487	63	F800-FBFF 63488-64511	64	FC00-FFFF 64512-65535

Pozn.: v prvním řádku je vždy šestnáctkové vyjádření,
v druhém řádku je vždy desítkové vyjádření.

Literatura

/1/ Klein, R.D.: Mikrocomputer Hard - und Softwarepraxis, Franzis´ Verlag, München - 1981.

/2/ Klein, R.D.: Mikrocomputer selbstgebaut und programmiert, Franzis´ Verlag, München - 1984.

/3/ Wiesemann, R.: Minimal-Computer mit Z80-CPU, mc 1/82, str. 71 až 75.

/4/ Ciarcia, St.: Comm-80 Adapted to ZX80, BYTE June 1983, str. 462.

/5/ Ciarcia, St.: Build Your Own Z80 Computer, BYTE Books/ McGraw-Hill, Peterborough New Hampshire - 1981.

/5/ Brinkhus, H.B. - Schneider, W.: Einfaches Interface zwischen dynamischen RAMs und Z80-Systemen, Elektronik 11/1981, str. 62 až 64.

/6/ Blank, H. - Schmidt, H.J.: Z80 ersetzt CPUs der 808/8085 -Familie, Elektronik 19/81, str. 71 až 74.

/7/ Lorig, K.D.: 64k on the 16k Dynamic RAM card, Elektor, september 1983, str. 29 až 31.

/8/ H. B.: Parallele Schnitstellen /IEC - oder IEEE - Bus?/, CHIP 3/1983, str. 221.
Was ist eine RS 232 C?, CHIP 3/1983, str. 233

/9/ Sergel, K.H.: CP/M - eine Sache mit Zukunft, mc 1/1982, str. 76 až 79.

/10/ Klein, R.D.: CP/M - ein Betriebssystem für jedermann, mc 1/1983, str. 42 až 42, mc 2/83, str. 35 až 36, mc 3/83, str. 37 až 38, mc 4/83, str. 38 až 40, mc 5/83, str. 52 až 53, mc 6/83, str. 68 až 70, mc 7/83, str. 42 až 46, mc 8/82, str. 29 až 31.

/11/ Joepgen, H.G.: Microshell - Ergänzung zu CP/M, mc 5/83, str. 66 až 67.

/12/ Schwenk, Kl.: Kopierprogramm für CP/M, mc 9/83, str. 69.

/13/ Klein, R.D.: Der mc-CP/M Computer, mc 9/82, str. 22 až 27, mc 10/82, str. 44 až 48,
Der mc-Monitor zum mc-Computer, mc 10/82, str. 74 až 80,

Der mc-Floppy-Controller für CP/M, mc 11/82, str. 73 až 77,
CP/M-Anpassung und Floppy-Routinen, mc 11/82, str. 48 až 54,
Nachträge zum mc-CP/M-Computer, mc 2/83, str. 55 až 56.

/14/ Heidenreich, U.: Z80-Code - schwerer Code?, mc 10/83, str. 96.

/15/ Hasselberg, M.: 422 neue Z80-Befehle, mc 1/82, str. 27.

/16/ Rybák, Vl.: IMS-2 - význam a ovládání základních spojovacích funkcí, Sdělovací technika 11/83, str. 407 až 409.

/17/ Kanis, W.: Der Z80-EMUF, Preiswerter Einplatinen-Computer, mc 4/83, str. 112 až 115.

/18/ Schwarz, W. - Meyer, G. - Eckhardt, D.: Mikrorechner, Wirkungsweise, Programmierung, Applikation, VEB Verlag Technik, Berlin 1981 (2. vydání).

/19/ Barden, W.Jr.: Z-80 Microcomputer Design Project, Howard W. Sams § Co., Inc., Indianapolis, Indiana - 1980.

/20/ Seul, A.: RAM/EPROM card for the Z80, Elektor, may 1982, str. 5-51 až 5-53.

/21/ Mester, R. - Götz, U.: Z80-A CPU card ...for the Polyformant, Elektor, may 1982, str. 5-28 až 5-31.

/22/ Riepe, J.: Einzelschrittschaltung für den Mikroprozessor Z80, Elektronik 24/79, str. 100.

/23/ Kol.: Einführung in die Mikroprozessortechnik, Texas Instruments Learning Center, Freising - 1977.

/24/ Osborne, A.: Einführung in die Mikrocomputertechnik, te-wi Verlag G mbH, München - 1978.

/25/ Hyan, J.T.: Mikroprocesor Z80 - Ročenka sdělovací techniky '84, kapitola 7: mikroprocesory a mikropočítače, str. 155 až 158, SNTL Praha - 1984.

/26/ Hyan, J.T.: Úvod do mikroprocesorové techniky - problematika uplatňování mikropočítačů v uživatelské sféře ASŘTP, Dům techniky ČSVTS Praha, 2. vyd.1984.

/27/ Hyan, J.T.: Akustický výstup mikropočítačů, Automatizace 7/81, str. 187 až 191.
/28/ Hyan, J.T.: Videointerface mikropočítačů, Automatizace 2/80, str. 53 až 56.
/29/ Hyan, J.T. a kol.: Mikroprocesory a mikropočítače, příloha časopisu Amatérské radio, 1/82 až 10/82.
/30/ Blatný, J. - Krištoufek, K. - Pokorný, Z. - Kolenička, J.: Číslicové počítače, SNTL/Alfa, Praha-1980.
/31/ Fadrhons, J.: Mikroprocesory Z80, 8085 a NSC 800, 1. část, Sdělovací technika 2/82, str. 259 až 261, 2. část: Sdělovací technika 8/82, str. 301 až 304.
/32/ Z80 - Nanocomputer Training System, firemní literatura fy SGS/ATES, Agrate Brianza - 1979.
/33/ Pelka, H.: Praxis mit Mikroprozessoren, 2. Aufl., München, Franzis´ Verlag GmbH - 1980.
/34/ Nichols, A.E. - Nichols, J.C. - Rony, P.E.: Z-80, Einführung und Programmierung, Elektor Verlag GmbH, Gangelt - 1979.
/35/ Kubař, Z.: Mikropočítačový systém z NDR /U880D = Z80/, Sdělovací technika 8/82, str. 299 až 300.
/36/ Dědina, B. - Valášek, P.: Mikroprocesory a mikropočítače, 2. vyd. SNTL Praha - 1983.
/37/ Z80 Microcomputer Device Family, Z80 Micro Applications, Microcomputer Components Data Book, str. 7 až 254, firemní literatura fy MOSTEK, MK79778, Carroltown - 1979.
/38/ MK 3880 Control Processing Unit, Z80 Microcomputer devices, Technical Manual fy MOSTEK, No.MK78505, Carroltown - 1977, Texas.
/39/ Z80-KIT Einfach-Computer und Lernsystem, firemní literatura fy KONTRON Elektronik GmbH, Eching b.München - 1979.
/40/ Klein, Michael: Z80 Applikationen, Frazis´ Verlag GmbH, München - 1980.
/41/ Carr, J.J.: Z80 Users Manual, Reston Publishing Company Inc., Reston, Virginia - 1980.

/42/ Hyan, J.T.: Krokování mikroprocesoru Z80, Ročenka sdělovací techniky '85, kap. 7, SNTL - Praha /v tisku/.

/43/ Zaks, R.: Programmierung des Z80, SYBEX-Verlag GmbH, 4. vydání, Düsseldorf - 1983.

/44/ Klein, R.D.: Mini-Floppy-Anschluss für den mc-CP/M Computer, mc 2/1983, str. 66 až 73.

/45/ Pelka, H.: Praxis mit Mikroprozessoren, kapt.: Modul für direkten Speicherzugriff, str. 112 až 119, Franzis' Verlag GmbH, München 1980.

/46/ Schurig, E.: Parallele Ein-/Ausgabe mit dem Mikroprozessor Z80, Elektronik 3/1980, str. 81 až 83.

/47/ Zaks, R.: Mikroprozessor Interface Techniken, 3. vydání, SYBEX-Verlag GmbH, Düsseldorf - 1983.

/48/ Zaks, R.: CP/M Handbuch mit MP/M, Sybex-Verlag GmbH, Düsseldorf 1981.

/49/ Zaks, R.: Chip und System, Einführung in die Mikroprozessoren-Technik, SYBEX-Verlag GmbH, Düsseldorf 1983.

/50/ Brendle, M.: Ein-/Ausgabe-Bus für Mikrocomputer, Elektronik 3/1980, str. 84 až 85.

/51/ Paolo, H.: Z80-EMUF steuert Selbstbau-Plotter, mc2/1284, str. 88 až 92.

/52/ Gaulke, E.: Z80-EMUF als Spooler, mc 10/1983.

/53/ AN.: Technical manual for serial I/O controller, MK 3884/MK 3885, Mostek Corp. Carrolltown - 1979, firemní literatura, No MK 78583.

/54/ AN.: Mikroprocesorový systém 8080, ZP-ČSVTS Tesla Piešťany, září 1978 (překlad příručky fy Siemens)

/55/ AN.: Mikrorechnerschaltkreise der II. Leistungsklasse, VEB Funkwerk Erfurt, katalog č. Ag 05-87-80 W-V-2-1.

/56/ Rybák, V.: IMS-2 - Sběrnice a přenos zpráv, Sdělovací technika 9/1983, str. 327 až 329.

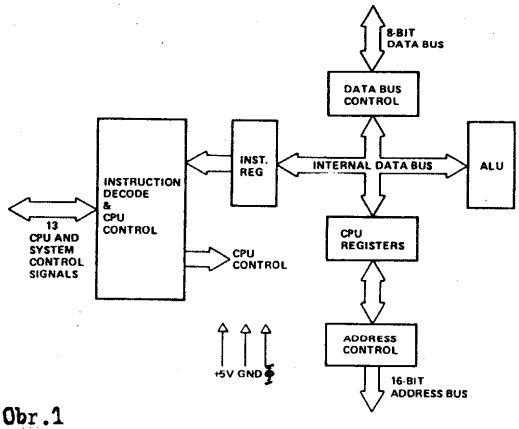
/57/ Klein, M.: Ein einfacher IEC-Bus-Monitor, Elektronik 2/1980.

/58/ Unger, G.: IEC-Bus-Kopplungsgerät für Digitalvoltmeter, Elektronik 17/1789, str. 39 až 46.
/59/ Richert, U.: IEC-Bus-Interface für Prozessrechner, Elektronik 12/1976, str. 58 až 62.
/60/ Uher, L.: Programovatelná přepojovací jednotka pro IMS-2 - přijímačová část, Sdělovací technika 5:1980, str. 173 až 174.
/61/ Math, I.: Math´s Notes, CQ October 1979, str. 73 až 74.
/62/ IN.: Interface für uP-Systeme, Schnittstelle zwischen Mikrocomputer und Elekterminal, Elektor Mai 1979, str. 5-47 až5-49.
/63/ AN.: Elekterminal: Video-Interface. 1024 Zeichen auf dem TV-Bildschirm, Elektor Dezember 1978, str. 12-42 až 12-50.
/64/ Plate, J.: Schnittstellen, Bindeglied zur Peripherie, mc 7/83, str. 30 až 33.
/65/ Stiny, L.: V.24 - ganz einfach, mc 7/1983, str. 34 až 35.
/66/ Langer, S.: Die Schnittstelle RS-232 - Beschreibung und Anwendung, mc 9/1982, str. 28.
/67/ AN.: Z80 CPU simulator, Elektor April 1984, str. 4-45 až 4-48.
/68/ Jeschke, H.: Sprache aus dem MZ-80, mc 7/1983, str. 71.
/69/ Tireford, H.: Die Adressierungsarten bei Mikroprozessoren, Elektronik 12/1976, str. 49 až 53.
/70/ Müller, R.: Pozor na rozdílný význam příznaku parity u mikroprocesorů 8080 a Z80, Sdělovací technika 5/84, str. 163 až 164.
/71/ Hyan, J.T.: Provozní systém CP/M, AR A 9/1984 - vložka mikroelektroniky.
/72/ Hyan, J.T.: Trendy rozvoje osobních mikropočítačů, sborník I ze semináře "Mini - mikro ´84", pořádaného ČSVTS Praha a DT - Praha, 1984.
/73/ Hyan, J.T.: Zesilovače sběrnic, AR A 9/1984 - vložka mikroelektroniky.
/74/ Hyan, J.T.: Stykové obvody PIO a SIO, Ročenka sdělovací techniky ´86, kapitola 7: mikroprocesory a mikropočítače, SNTL - Praha (v tisku).

/75/ Šrotýř, M.: Optoelektronické vazební členy ve vstupních a výstupních obvodech mikropočítačů. Automatizace 12/1980, str. 325 až 328.

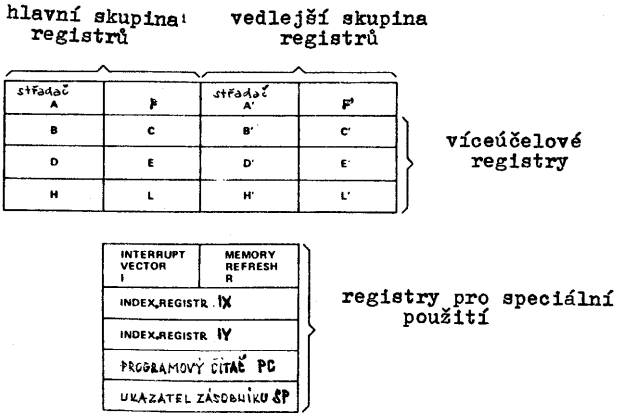
OBRÁZKY A TABULKY

ARCHITEKTURA MIKROPROCESORU Z80



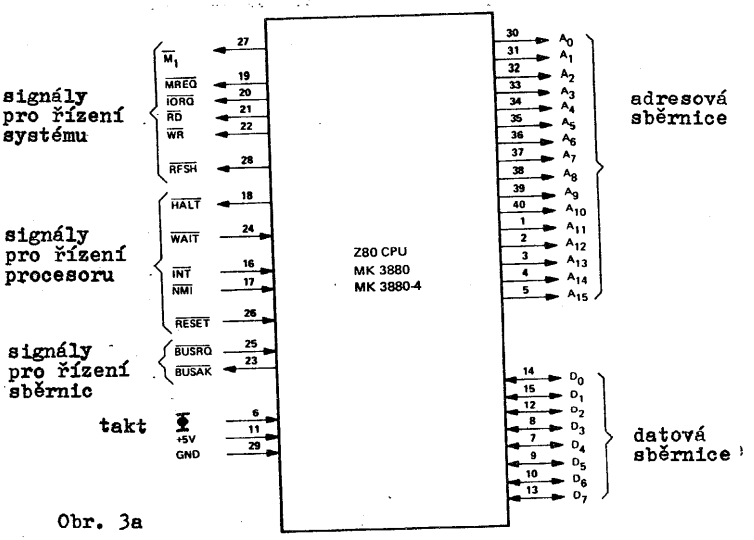
Obr.1

USPOŘÁDÁNÍ REGISTRŮ MIKROPROCESORU Z80



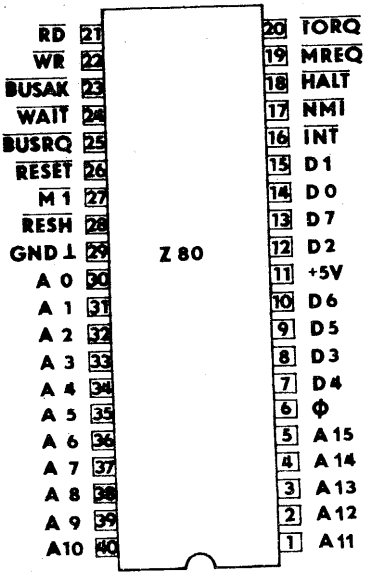
Obr. 2

FUNKCE VÝVODŮ MIKROPROCESORU Z80



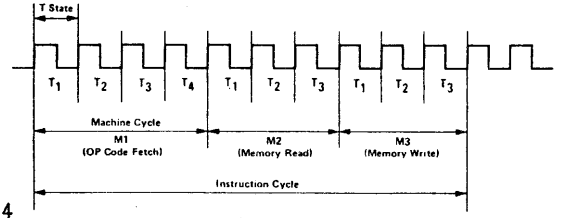
Obr. 3a

POUZDRO MIKROPROCESORU Z80 A OZNAČENÍ JEHO VÝVODŮ



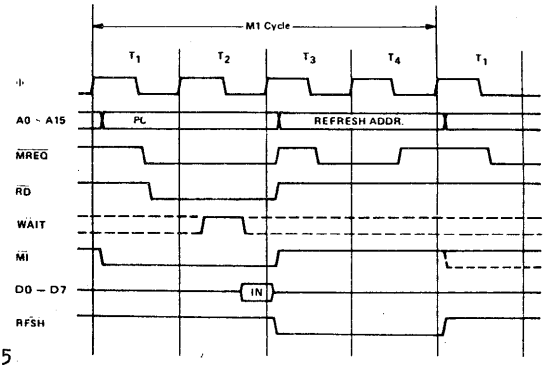
Obr. 3b

ČASOVÁNÍ INSTRUKČNÍHO CYKLU



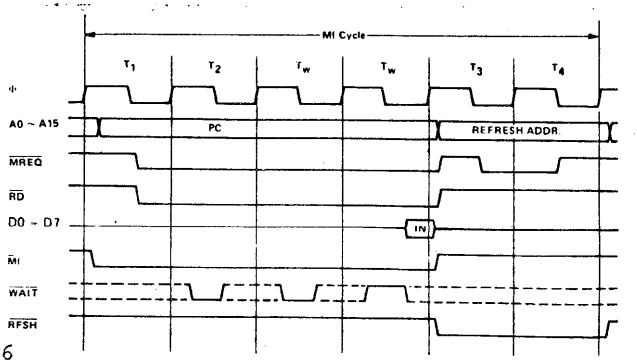
Obr. 4

PRŮBĚH STROJNÍHO CYKLU M1



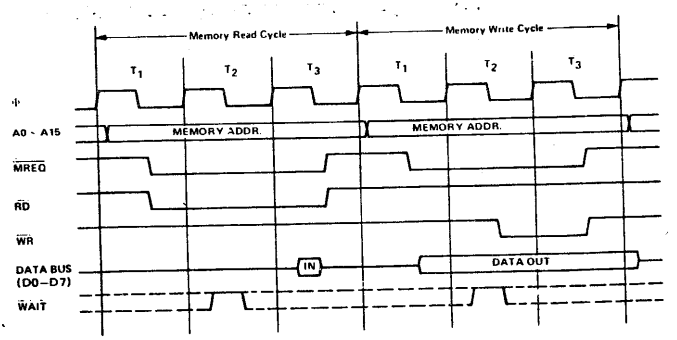
Obr. 5

PRŮBĚH STROJNÍHO CYKLU M1 S VYČKÁVACÍMI STAVY Tw



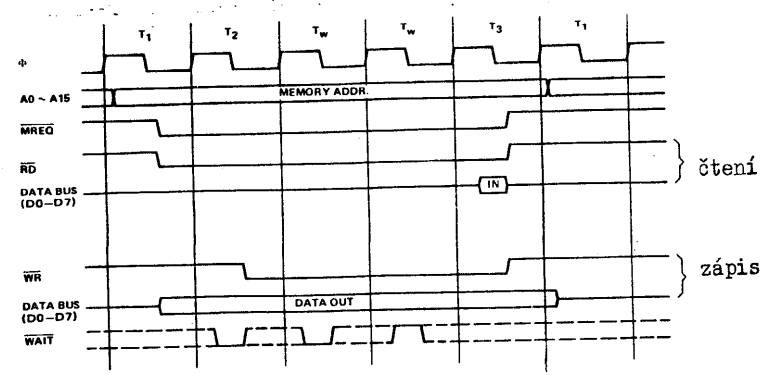
Obr. 6

PRŮBĚH CYKLU ČTENÍ A ZÁPISU DO PAMĚTI



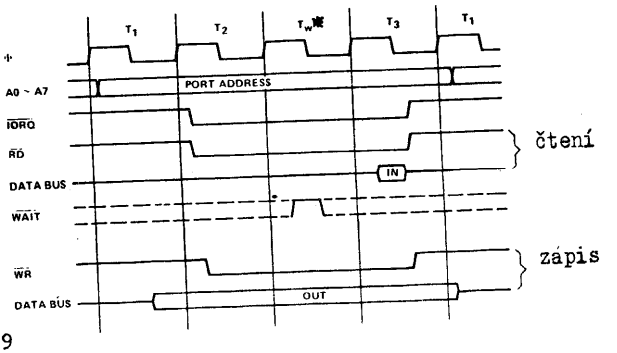
Obr. 7

PRODLOUŽENÉ CYKLY ČTENÍ A ZÁPISU O VYČKÁVACÍ STAVY Tw



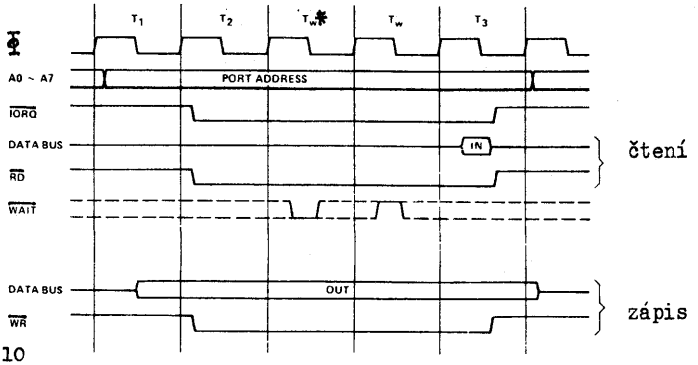
Obr. 8

PRŮBĚH CYKLU KOMUNIKACE S V/V ZAŘÍZENÍMI



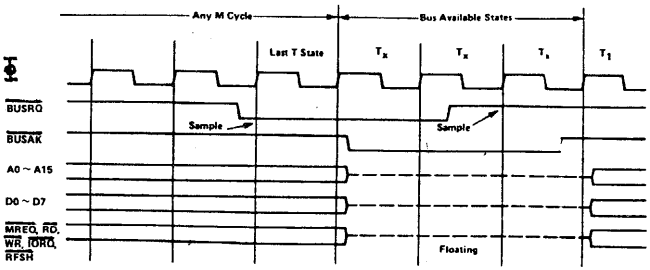
Obr. 9

PRODLOUŽENÉ CYKLY KOMUNIKACE S V/V ZAŘÍZENÍM
O VYČKÁVACÍ STAVY TV



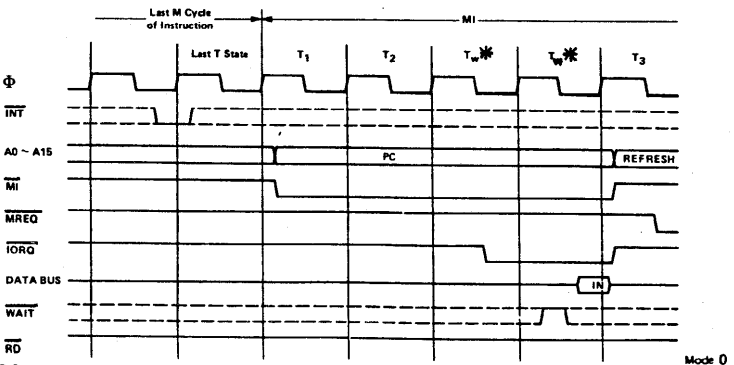
Obr. 10

PRŮBĚH CYKLU UVOLNĚNÍ SBĚRNIC



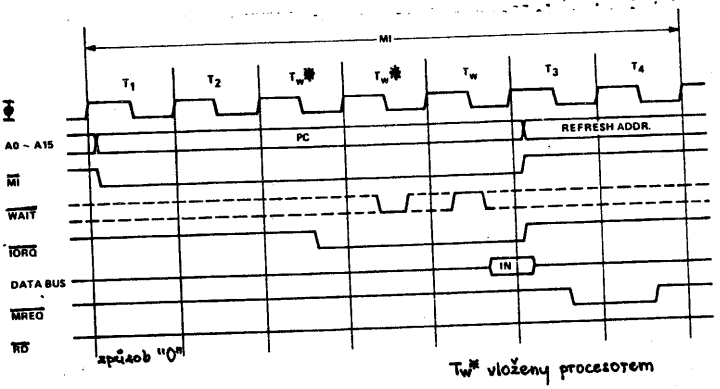
Obr. 11

PRŮBĚH CYKLU ŽÁDOSTI O PŘERUŠENÍ



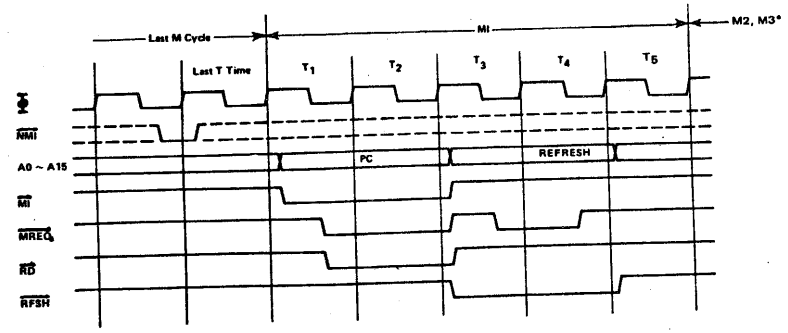
Obr. 12

PRŮBĚH CYKLU ŽÁDOSTI O PŘERUŠENÍ S PRODLOUŽENÍM O Tw



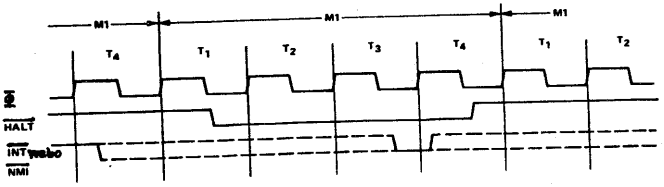
Obr. 13

PRŮBĚH CYKLU NEMASKOVATELNÉHO PŘERUŠENÍ



Obr. 14

PRŮBĚH CYKLU ZA STAVU "HALT"



Obr. 15

Tabulka č. 1 - Operační kódy osmibitových instrukcí ukládání, zdroj

místo určení

		IMPLIED		REGISTER							REG INDIRECT			INDEXED		EXT. ADDR.	IMME.
		I	R	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(IX + d)	(IY + d)	(nn)	n
REGISTER	A	ED 57	ED 5F	7F	78	79	7A	7B	7C	7D	7E	0A	1A	DD 7E d.	FD 7E d	3A n n	3E n
	B			47	40	41	42	43	44	45	46			DD 46 d	FD 46 d		06 n
	C			4F	48	49	4A	4B	4C	4D	4E			DD 4E d	FD 4E d		0E n
	D			57	50	51	52	53	54	55	56			DD 56 d	FD 56 d		16 n
	E			5F	58	59	5A	5B	5C	5D	5E			DD 5E d	FD 5E d		1E n
	H			67	60	61	62	63	64	65	66			DD 66 d	FD 66 d		26 n
	L			6F	68	69	6A	6B	6C	6D	6E			DD 6E d	FD 6E d		2E n
REG INDIRECT	(HL)			77	70	71	72	73	74	75							36 n
	(BC)			02													
	(DE)			12													
INDEXED	(IX+d)			DD 77 d	DD 70 d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d							DD 36 d n
	(IY+d)			FD 77 d	FD 70 d	FD 71 d	FD 72 d	FD 73 d	FD 74 d	FD 75 d							FD 36 d n
EXT. ADDR	(nn)			32 n n													
IMPLIED	I			ED 47													
	R			ED 4F													

Tabulka č. 2 - Operační kódy šestnáctibitových instrukcí ukládání a instrukcí PUSH a POP

místo určení

zdroj

		REGISTER							IMM. EXT.	EXT. ADDR.	REG. INDIR.
		AF	BC	DE	HL	SP	IX	IY	nn	(nn)	(SP)
REGISTER	AF										F1
	BC								01 n n	ED 4B n n	C1
	DE								11 n n	ED 5B n n	D1
	HL								21 n n	2A n n	E1
	SP				F9		DD F9	FD F9	31 n n	ED 7B n n	
	IX								DD 21 n n	DD 2A n n	DD E1
	IY								FD 21 n n	FD 2A n n	FD E1
EXT. ADDR.	(nn)		ED 43 n n	ED 53 n n	22 n n	ED 73 n n	DD 22 n n	FD 22 n n			
REG. IND.	(SP)	F5	C5	D5	E5		DD E5	FD E5			

PUSH INSTRUCTIONS →

↑ POP INSTRUCTIONS

Tabulka č. 3 - Operační kódy instrukcí výměny

		IMPLIED ADDRESSING				
		AF'	BC', DE' & HL'	HL	IX	IY
IMPLIED	AF	08				
	BC, DE & HL		D9			
	DE			EB		
REG. INDIR.	(SP)			E3	DD E3	FD E3

Tabulka č. 4 - Operační kódy instrukcí blokového přenosu

místo určení		zdroj: REG. INDIR. (HL)	
REG. INDIR.	(DE)	ED A0	'LDI' – Load (DE)←(HL), Inc HL & DE, Dec BC
		ED B0	'LDIR' – Load (DE)←(HL) Inc HL & DE, Dec BC, Repeat until BC = 0
		ED A8	'LDD' – Load (DE)←(HL) Dec HL & DE, Dec BC
		ED B8	'LDDR' – Load (DE)←(HL) Dec HL & DE, Dec BC, Repeat until BC = 0

Reg HL points to source
Reg DE points to destination
Reg BC is byte counter

Tabulka č. 5 - Operační kódy instrukcí vyhledávání

SEARCH LOCATION

REG. INDIR. (HL)	
ED A1	'CPI' Inc HL, Dec BC
ED B1	'CPIR', Inc HL, Dec BC repeat until BC = 0 or find match
ED A9	'CPD' Dec HL & BC
ED B9	'CPDR' Dec HL & BC Repeat until BC = 0 or find match

HL points to location in memory to be compared with accumulator contents
BC is byte counter

Tabulka č. 6 - Operační kódy osmibitových aritmetických a logických instrukcí

	zdroj										
	REGISTER ADDRESSING							REG. INDIR.	INDEXED		IMMED.
	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	n
'ADD'	87	80	81	82	83	84	85	86	DD 86 d	FD 86 d	C6 n
ADD w CARRY 'ADC'	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n
SUBTRACT 'SUB'	97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n
SUB w CARRY 'SBC'	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n
'AND'	A7	A0	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n
'XOR'	AF	A8	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n
'OR'	B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	F6 n
COMPARE 'CP'	BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n
INCREMENT 'INC'	3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d	
DECREMENT 'DEC'	3D	05	0D	15	1D	25	2D	35	DD 35 d	FD 35 d	

Tabulka č. 7 - Operační kódy instrukcí registrů A a F

Decimal Adjust Acc, 'DAA'	27
Complement Acc, 'CPL'	2F
Negate Acc, 'NEG' (2's complement)	ED 44
Complement Carry Flag, 'CCF'	3F
Set Carry Flag, 'SCF'	37

Tabulka č. 8 - Operační kódy šestnáctibitových aritmetických instrukcí

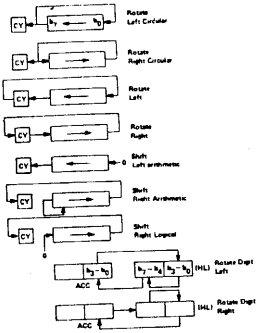
místo určení		zdroj BC	DE	HL	SP	IX	IY
'ADD'	HL	09	19	29	39		
	IX	DD 09	DD 19		DD 39	DD 29	
	IY	FD 09	FD 19		FD 39		FD 29
ADD WITH CARRY AND SET FLAGS 'ADC'	HL	ED 4A	ED 5A	ED 6A	ED 7A		
SUB WITH CARRY AND SET FLAGS 'SBC'	HL	ED 42	ED 52	ED 62	ED 72		
INCREMENT 'INC'		03	13	23	33	DD 23	FD 23
DECREMENT 'DEC'		0B	1B	2B	3B	DD 2B	FD 2B

zdroj a umístění

typ rotace či posuvu

	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)
'RLC'	CB 07	CB 00	CB 01	CB 02	CB 03	CB 04	CB 05	CB 06	DD CB d 06	FD CB d 06
'RRC'	CB 0F	CB 08	CB 09	CB 0A	CB 0B	CB 0C	CB 0D	CB 0E	DD CB d 0E	FD CB d 0E
'RL'	CB 17	CB 10	CB 11	CB 12	CB 13	CB 14	CB 15	CB 16	DD CB d 16	FD CB d 16
'RR'	CB 1F	CB 18	CB 19	CB 1A	CB 1B	CB 1C	CB 1D	CB 1E	DD CB d 1E	FD CB d 1E
'SLA'	CB 27	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	CB 26	DD CB d 26	FD CB d 26
'SRA'	CB 2F	CB 28	CB 29	CB 2A	CB 2B	CB 2C	CB 2D	CB 2E	DD CB d 2E	FD CB d 2E
'SRL'	CB 3F	CB 38	CB 39	CB 3A	CB 3B	CB 3C	CB 3D	CB 3E	DD CB d 3E	FD CB d 3E
'RLD'								ED 6F		
'RRD'								ED 67		

	A
RLCA	07
RRCA	0F
RLA	17
RRA	1F



Tabulka č. 9 - Operační kódy příkazů rotace a posunu

Tabulka č. 10 - Operační kódy manipulace s bity

	BIT	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)
TEST BIT 'BIT'	0	CB 47	CB 40	CB 41	CB 42	CB 43	CB 44	CB 45	CB 46	DD CB d 46	FD CB d 46
	1	CB 4F	CB 48	CB 49	CB 4A	CB 4B	CB 4C	CB 4D	CB 4E	DD CB d 4E	FD CB d 4E
	2	CB 57	CB 50	CB 51	CB 52	CB 53	CB 54	CB 55	CB 56	DD CB d 56	FD CB d 56
	3	CB 5F	CB 58	CB 59	CB 5A	CB 5B	CB 5C	CB 5D	CB 5E	DD CB d 5E	FD CB d 5E
	4	CB 67	CB 60	CB 61	CB 62	CB 63	CB 64	CB 65	CB 66	DD CB d 66	FD CB d 66
	5	CB 6F	CB 68	CB 69	CB 6A	CB 6B	CB 6C	CB 6D	CB 6E	DD CB d 6E	FD CB d 6E
	6	CB 77	CB 70	CB 71	CB 72	CB 73	CB 74	CB 75	CB 76	DD CB d 76	FD CB d 76
	7	CB 7F	CB 78	CB 79	CB 7A	CB 7B	CB 7C	CB 7D	CB 7E	DD CB d 7E	FD CB d 7E
RESET BIT 'RES'	0	CB 87	CB 80	CB 81	CB 82	CB 83	CB 84	CB 85	CB 86	DD CB d 86	FD CB d 86
	1	CB 8F	CB 88	CB 89	CB 8A	CB 8B	CB 8C	CB 8D	CB 8E	DD CB d 8E	FD CB d 8E
	2	CB 97	CB 90	CB 91	CB 92	CB 93	CB 94	CB 95	CB 96	DD CB d 96	FD CB d 96
	3	CB 9F	CB 98	CB 99	CB 9A	CB 9B	CB 9C	CB 9D	CB 9E	DD CB d 9E	FD CB d 9E
	4	CB A7	CB A0	CB A1	CB A2	CB A3	CB A4	CB A5	CB A6	DD CB d A6	FD CB d A6
	5	CB AF	CB A8	CB A9	CB AA	CB AB	CB AC	CB AD	CB AE	DD CB d AE	FD CB d AE
	6	CB B7	CB B0	CB B1	CB B2	CB B3	CB B4	CB B5	CB B6	DD CB d B6	FD CB d B6
	7	CB BF	CB B8	CB B9	CB BA	CB BB	CB BC	CB BD	CB BE	DD CB d BE	FD CB d BE
SET BIT 'SET'	0	CB C7	CB C0	CB C1	CB C2	CB C3	CB C4	CB C5	CB C6	DD CB d C6	FD CB d C6
	1	CB CF	CB C8	CB C9	CB CA	CB CB	CB CC	CB CD	CB CE	DD CB d CE	FD CB d CE
	2	CB D7	CB D0	CB D1	CB D2	CB D3	CB D4	CB D5	CB D6	DD CB d D6	FD CB d D6
	3	CB DF	CB D8	CB D9	CB DA	CB DB	CB DC	CB DD	CB DE	DD CB d DE	FD CB d DE
	4	CB E7	CB E0	CB E1	CB E2	CB E3	CB E4	CB E5	CB E6	DD CB d E6	FD CB d E6
	5	CB EF	CB E8	CB E9	CB EA	CB EB	CB EC	CB ED	CB EE	DD CB d EE	FD CB d EE
	6	CB F7	CB F0	CB F1	CB F2	CB F3	CB F4	CB F5	CB F6	DD CB d F6	FD CB d F6
	7	CB FF	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	CB FE	DD CB d FE	FD CB d FE

podmínka

			UN-COND.	CARRY	NON CARRY	ZERO	NON ZERO	PARITY EVEN	PARITY ODD	SIGN NEG	SIGN POS	REG B≠0
JUMP 'JP'	IMMED. EXT.	nn	C3 n n	DA n n	D2 n n	CA n n	C2 n n	EA n n	E2 n n	FA n n	F2 n n	
JUMP 'JR'	RELATIVE	PC+e	18 e-2	38 e-2	30 e-2	28 e-2	20 e-2					
JUMP 'JP'	REG. INDIR.	(HL)	E9									
JUMP 'JP'		(IX)	DD E9									
JUMP 'JP'		(IY)	FD E9									
'CALL'	IMMED. EXT.	nn	CD n n	DC n n	D4 n n	CC n n	C4 n n	EC n n	E4 n n	FC n n	F4 n n	
DECREMENT B, JUMP IF NON ZERO 'DJNZ'	RELATIVE	PC+e										10 e-2
RETURN 'RET'	REGISTER INDIR.	(SP) (SP+1)	C9	D8	D0	C8	C0	E8	E0	F8	F0	
RETURN FROM INT 'RETI'	REG. INDIR.	(SP) (SP+1)	ED 4D									
RETURN FROM NON MASKABLE INT 'RETN'	REG. INDIR.	(SP) (SP+1)	ED 45									

NOTE–CERTAIN FLAGS HAVE MORE THAN ONE PURPOSE

Tabulka č. 11 - Operační kódy skoků, volání a návratů

OP kód

CALL ADDRESS		
0000_H	C7	'RST 0'
0008_H	CF	'RST 8'
0010_H	D7	'RST 16'
0018_H	DF	'RST 24'
0020_H	E7	'RST 32'
0028_H	EF	'RST 40'
0030_H	F7	'RST 48'
0038_H	FF	'RST 56'

Tabulka č. 12 - OP kódy restartu

adresa portu

určení vstupu

		IMMED. n	REG. INDIR. (C)
INPUT 'IN'	A	DB	ED 78
	B		ED 40
	C		ED 48
	D		ED 50
	E		ED 58
	H		ED 60
	L		ED 68
'INI' – INPUT & Inc HL, Dec B	REG. INDIR (HL)		ED A2
'INIR' – INP, Inc HL, Dec B, REPEAT IF B≠0			ED B2
'IND' – INPUT & Dec HL, Dec B			ED AA
'INDR' – INPUT, Dec HL, Dec B, REPEAT IF B≠0			ED BA

REG ADDRESSING

příkazy blokového vstupu

Tabulka č. 13 - OP kódy vstupu

zdroj

			REGISTER							REG. IND.
			A	B	C	D	E	H	L	(HL)
'OUT'	IMMED.	n	D3 n							
	REG. IND.	(C)	ED 79	ED 41	ED 49	ED 51	ED 59	ED 61	ED 69	
'OUTI' – OUTPUT Inc HL, Dec b	REG. IND.	(C)								ED A3
'OTIR' – OUTPUT, Inc HL, Dec B, REPEAT IF B≠0	REG. IND.	(C)								ED B3
'OUTD' – OUTPUT Dec HL & B	REG. IND.	(C)								ED AB
'OTDR' – OUTPUT, Dec HL & B, REPEAT IF B≠0	REG. IND.	(C)								ED BB

příkazy blokového výstupu

adresa určeného portu

Tabulka č. 14 - Operační kódy výstupu

'NOP'	00
'HALT'	76
DISABLE INT '(DI)'	F3
ENABLE INT '(EI)'	FB
SET INT MODE 0 'IM0'	ED 46
SET INT MODE 1 'IM1'	ED 56
SET INT MODE 2 'IM2'	ED 5E

8080A MODE

CALL TO LOCATION 0038_H

INDIRECT CALL USING REGISTER I AND 8 BITS FROM INTERRUPTING DEVICE AS A POINTER.

Tabulka č. 15 - Operační kódy řízení

Tabulka č. 16 - Souhrn operací ovlivňujících příznakové bity

příkaz	D7 S	Z		H		P/V	N	D0 C	komentář
ADD A,s; ADC A,s	↕	↕	X	↕	X	V	0	↕	8-bit add or add with carry
SUB,s; SBCA,s; CP,s; NEG	↕	↕	X	↕	X	V	1	↕	8-bit subtract, subtract with carry, compare and negate accumulator
AND s	↕	↕	X	1	X	P	0	0	} Logical operations
OR s; XOR s	↕	↕	X	0	X	P	0	0	} Logical operations
INC s	↕	↕	X	↕	X	V	0	•	8-bit increment
DEC s	↕	↕	X	↕	X	V	1	•	8-bit decrement
ADD DD, SS	•	•	X	X	X	•	0	↕	16-bit add
ADC HL, SS	↕	↕	X	X	X	V	0	↕	16-bit add with carry
SBC HL, SS	↕	↕	X	X	X	V	1	↕	16-bit subtract with carry
RLA; RLCA; RRA; RRCA	•	•	X	0	X	•	0	↕	Rotate accumulator
RL s; RLC s; RR s; RRC s; SLA s; SRA s; SRL s	↕	↕	X	0	X	P	0	↕	Rotate and shift locations
RLD; RRD	↕	↕	X	0	X	P	0	•	Rotate digit left and right
DAA	↕	↕	X	↕	X	P	•	↕	Decimal adjust accumulator
CPL	•	•	X	1	X	•	1	•	Complement accumulator
SCF	•	•	X	0	X	•	0	1	Set carry
CCF	•	•	X	X	X	•	0	↕	Complement carry
IN r, (C)	↕	↕	X	0	X	P	0	•	Input register indirect
INI; IND; OUTI; OUTD	X	↕	X	X	X	X	1	•	} Block input and output
INIR; INDR; OTIR; OTDR	X	1	X	X	X	X	1	•	} Z = 0 if B ≠ 0 otherwise Z = 1
LDI; LDD	X	X	X	0	X	↕	0	•	} Block transfer instructions
LDIR; LDDR	X	X	X	0	X	0	0	•	} P/V = 1 if BC ≠ 0, otherwise P/V = 0
CPI; CPIR; CPD; CPDR	X	↕	X	X	X	↕	1	•	Block search instructions Z = 1 if A = (HL), otherwise Z = 0 P/V = 1 if BC ≠ 0, otherwise P/V = 0
LD A, I; LD A, R	↕	↕	X	0	X	IFF	0	•	The content of the interrupt enable flip-flop (IFF) is copied into the P/V flag
BIT b, s	X	↕	X	1	X	X	0	•	The state of bit b of location s is copied into the Z flag

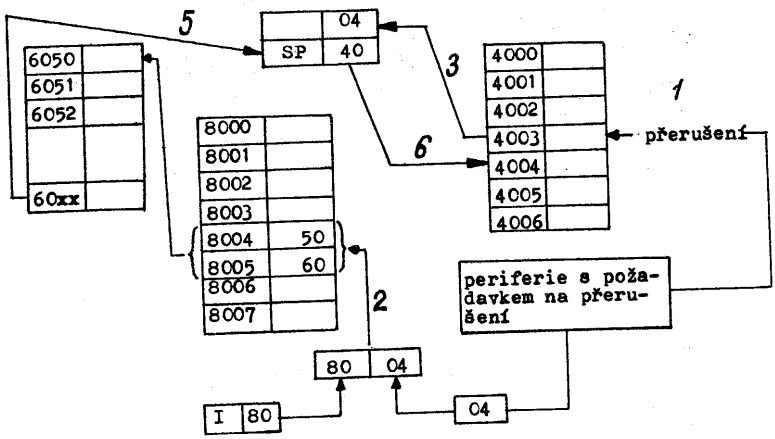
Legenda:

C - bit přenosu, C = 1, jestliže nejvýznačnější bit operandu či výsledku generuje přenos
Z - bit nuly, Z = 1, jestliže výsledek operace je nula
S - bit znaménka, S = 1, jestliže nejvýznačnější bit výsledku je jedničkový
P/V - bit parity/přeplnění, P/V = 1, jestliže výsledek operace má sudou paritu, P/V = 0 při liché paritě, P/V = 1, jestliže výsledek operace generuje přeplnění
H - bit polovičního přenosu, H = 1, jestliže operace odčítání či přičítání generuje výpůjčku či přenos do D4 střadače
N - bit příznaku sčítání/odčítání. N = 1, jestliže předcházející bylo odčítání
  - bit je ovlivněn výsledkem operace
  - bit je nezměněn operací
0 - příznakový bit je operací nulován
1 - příznakový bit je operací nahozen
X - bit je nedefinován
V - bit P/V ve funkci indikátoru přeplnění
P - bit P/V ve funkci indikátoru parity
r - některý z registrů A, B, C, D, E, H, L
s - osmibitové umístění
ss - šestnáctibitové umístění
ii - některý z dvou indexových registrů IX či IY
R - osvěžovací registr
n - osmibitová hodnota v rozmezí 0 až 255
nn - šestnáctibitová hodnota v rozmezí 0 až 65 535

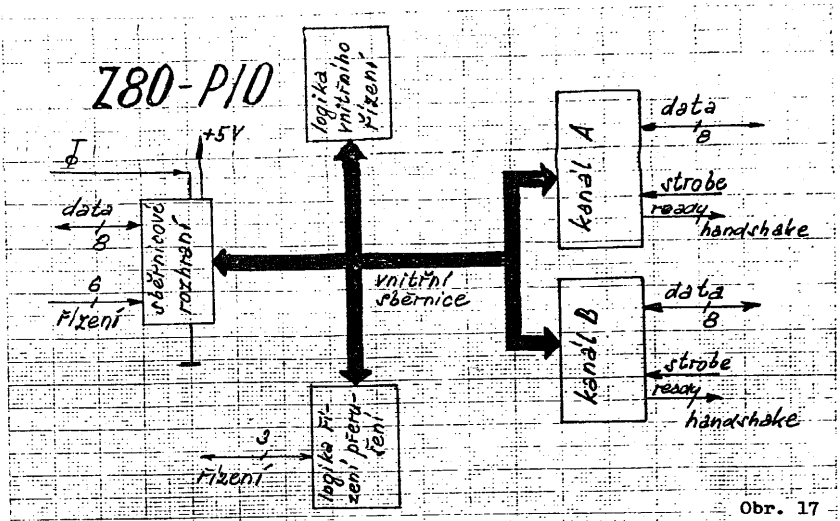
Tabulka č. 17 - Účinek různých instrukcí na dvojici uvolňovacích klopných obvodů přerušení

činnost-příkaz	IFF_1	IFF_2	pozn.
nulování CPJ (reset)	0	0	
"DI"	0	0	
"EI"	1	1	
"LDA,I"	.	.	IFF_2 P/V
"LDA,R"	.	.	IFF_2 P/V
příjem signálu NMI	0	.	
"RETN"	IFF_2	.	IFF_2 IFF_1
příjem signálu INT	0	0	
"RETI"	.	.	

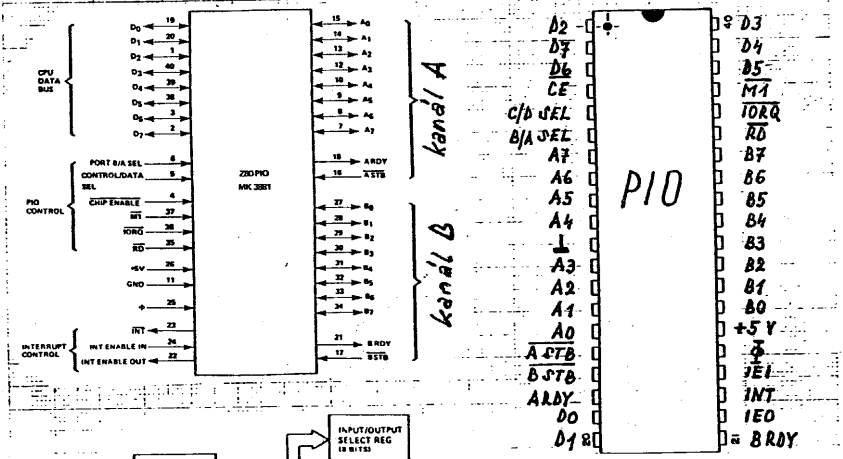
"." vyjadřuje stav beze změny

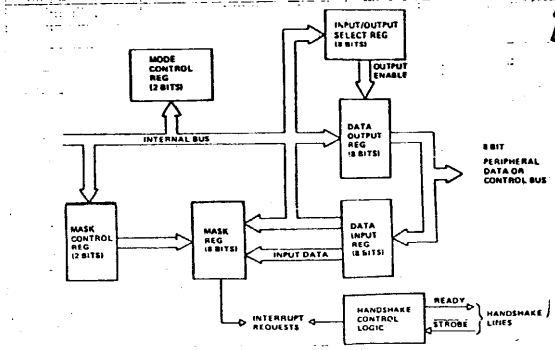


Obr. 16

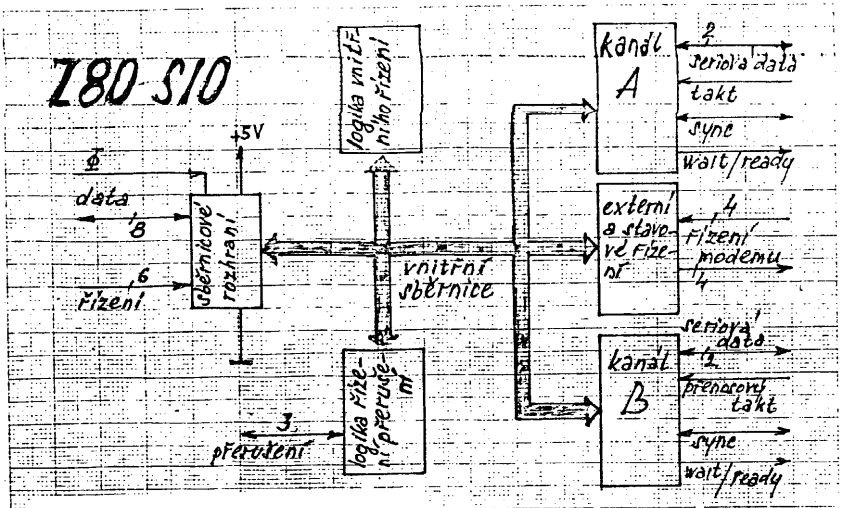

Z80-PIO
logika vnitřního řízení
+5V
Φ
data 8
sběrnicové rozhraní
8
řízení
vnitřní sběrnice
kanál A
data 8
strobe
ready
handshake
kanál B
data 8
strobe
ready
handshake
logika řízení přerušení
3
řízení

Obr. 17

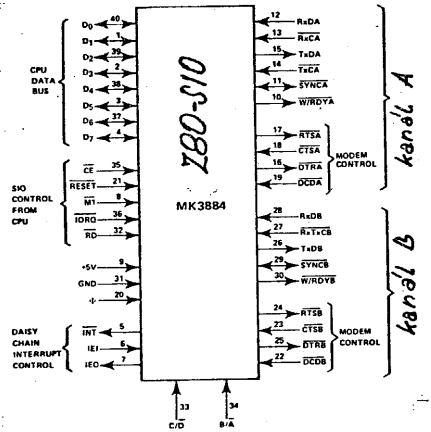

CPU DATA BUS
PIO CONTROL
INTERRUPT CONTROL
Z80 PIO MK 3881
kanál A
kanál B
PIO
+5V
GND

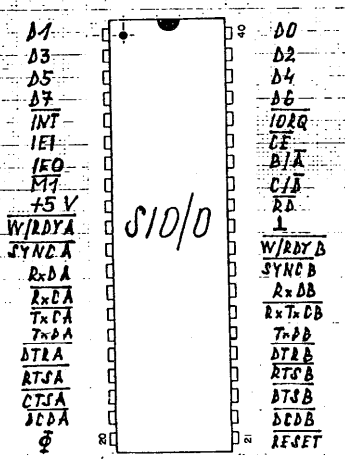

INPUT/OUTPUT SELECT REG (8 BITS)
OUTPUT ENABLE
MODE CONTROL REG (2 BITS)
INTERNAL BUS
DATA OUTPUT REG (8 BITS)
8 BIT PERIPHERAL DATA OR CONTROL BUS
MASK CONTROL REG (2 BITS)
MASK REG (8 BITS)
INPUT DATA
DATA INPUT REG (8 BITS)
INTERRUPT REQUESTS
HANDSHAKE CONTROL LOGIC
READY
STROBE
HANDSHAKE LINES

Obr. 18


Z80 SIO
+5V
Φ
data
8
6
řízení
sběrnicové rozhraní
logika vnitř. řízení
vnitřní sběrnice
kanál A
2
sériová data
takt
sync
wait/ready
externí a stavové řízení
4
řízení modemu
4
kanál B
sériová data
2
přenosový takt
sync
wait/ready
3
přerušení
logika řízení přerušení

Obr. 19

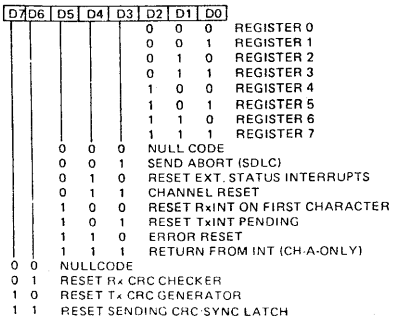

Z80-SIO
MK3884
CPU DATA BUS
SIO CONTROL FROM CPU
DAISY CHAIN INTERRUPT CONTROL
MODEM CONTROL
kanál A
kanál B
C/D
B/A


SIO/0

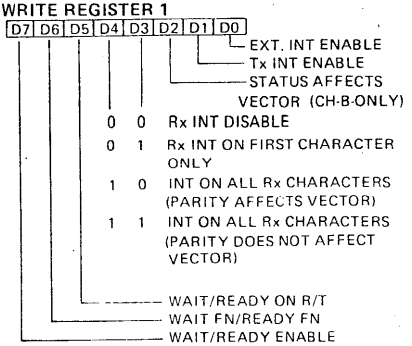
Obr. 20

Tab. č. 18 - Registry zápisu

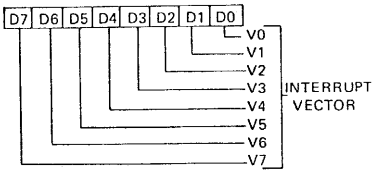
WRITE REGISTER 0



WRITE REGISTER 1

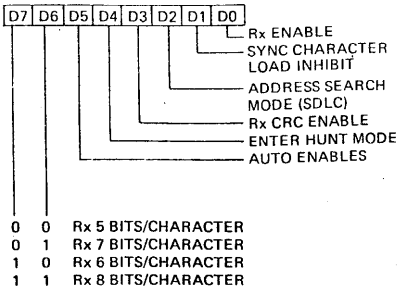


WRITE REGISTER 2*

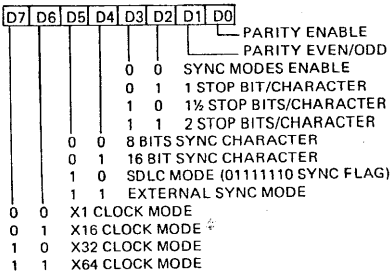


*CAN ONLY BE WRITTEN INTO CHANNEL B

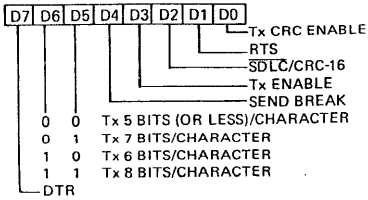
WRITE REGISTER 3



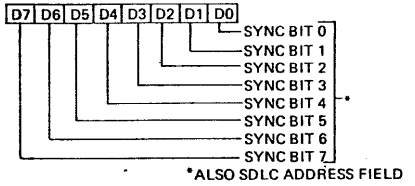
WRITE REGISTER 4



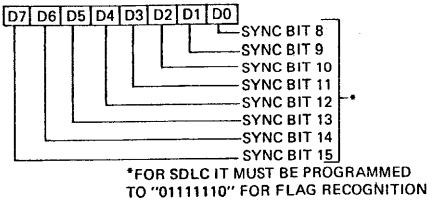
WRITE REGISTER 5



WRITE REGISTER 6

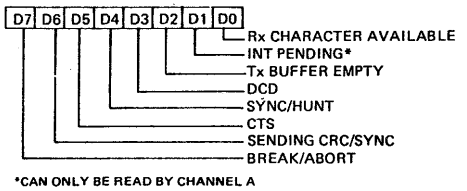


WRITE REGISTER 7

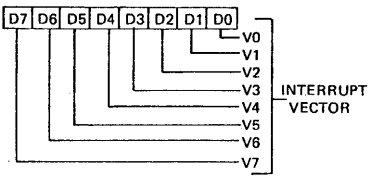


Tab. č. 19 – Čtecí registry

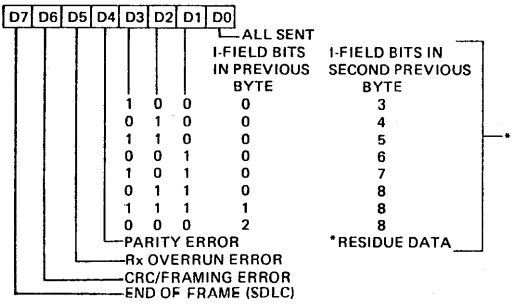
READ REGISTER 0

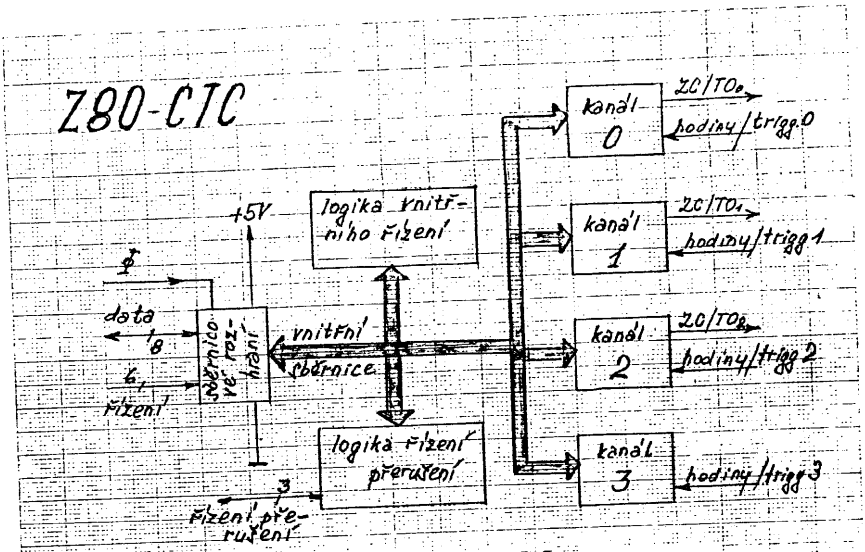


READ REGISTER 2 (Channel B Only)

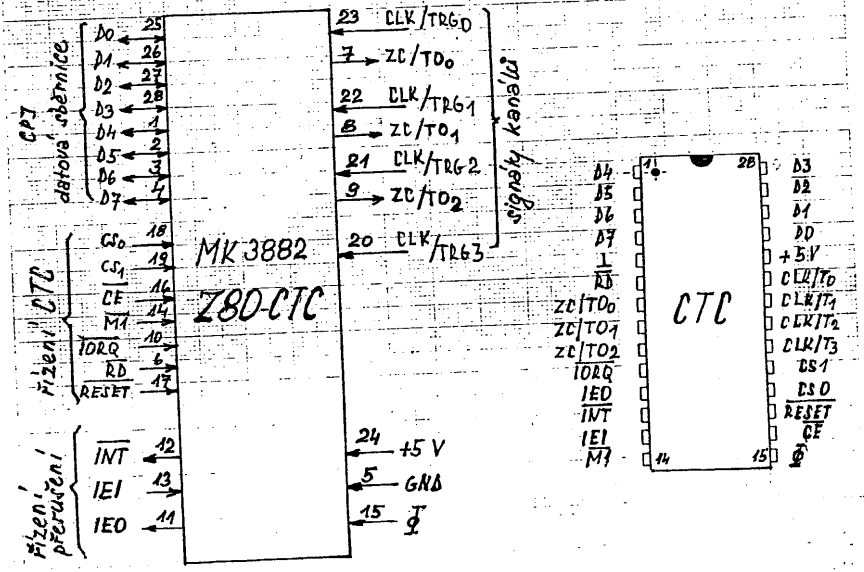


READ REGISTER 1

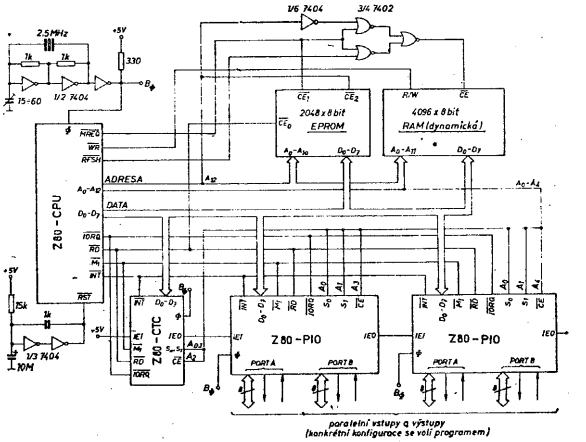



Z80-CIC
kanál 0
kanál 1
kanál 2
kanál 3
ZC/TO0
hodiny/trigg 0
ZC/TO1
hodiny/trigg 1
ZC/TO2
hodiny/trigg 2
hodiny/trigg 3
logika vnitřního řízení
logika řízení přerušení
+5V
Φ
data 8
řízení 6
3
řízení přerušení
vnitřní sběrnice
sběrnice řízení
vnitřní sběrnice

Obr. 22

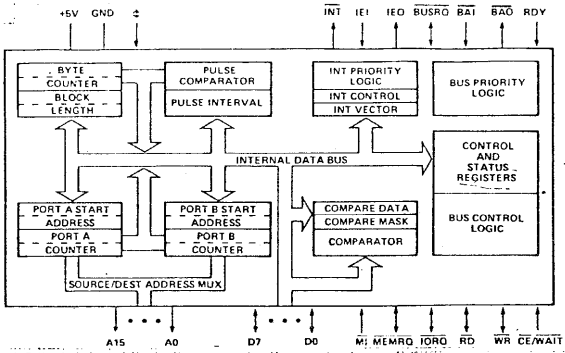

MK 3882
Z80-CIC
CTC
datová sběrnice CPU
řízení CTC
řízení přerušení
signály kanálů
D0 D1 D2 D3 D4 D5 D6 D7
CS0 CS1 CE M1 IORQ RD RESET
INT IEI IEO
CLK/TRG0
ZC/TO0
CLK/TRG1
ZC/TO1
CLK/TRG2
ZC/TO2
CLK/TRG3
+5 V
GND
Φ
D4 D5 D6 D7 CE RD ZC/TO0 ZC/TO1 ZC/TO2 IORQ IEO INT IEI M1
D3 D2 D1 D0 +5V CLK/T0 CLK/T1 CLK/T2 CLK/T3 CS1 CS0 RESET CE Φ

Obr. 21

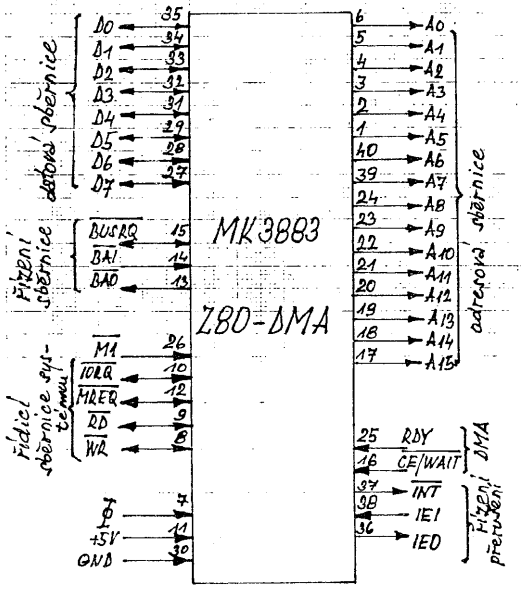


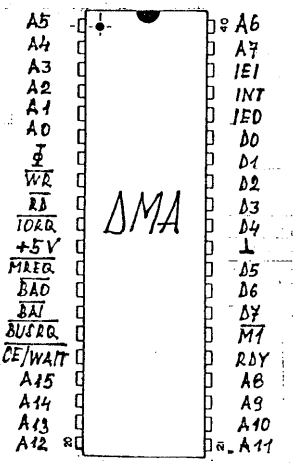
Obr. 23 Zapojení malého systému s obvody série Z80

Z80-DMA



Obr. 24

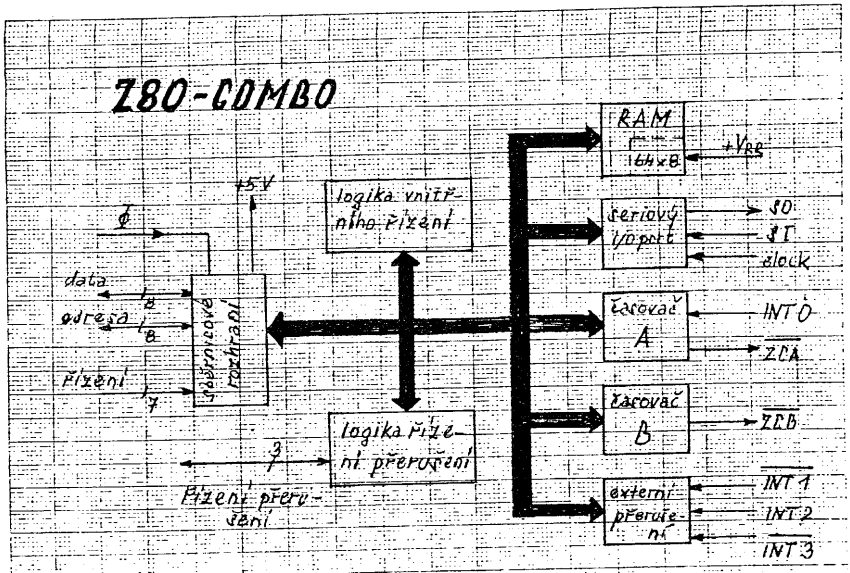




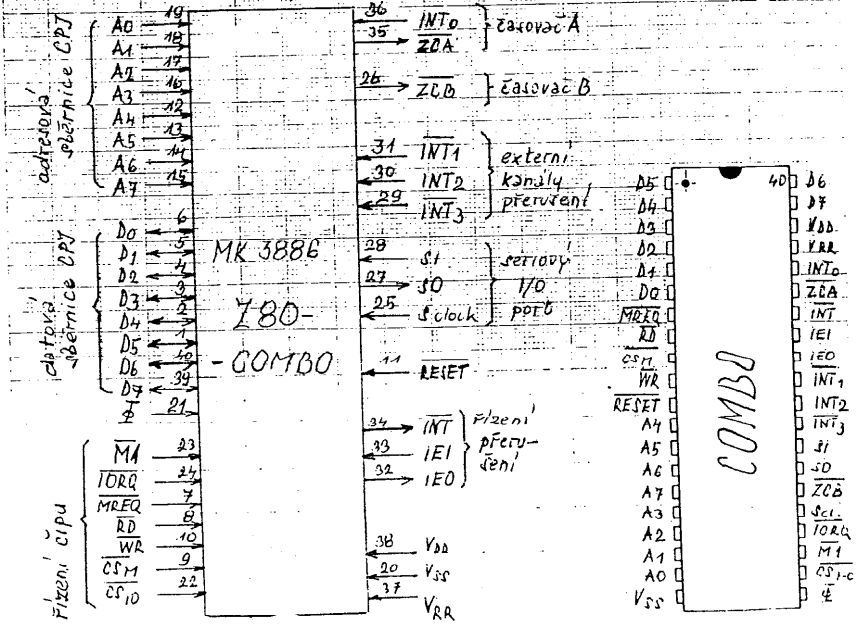
Obr. 25

	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1 Command Byte 1A Sets the DMA to receive Block length and Port A address and sets direction of transfer	0 Group 1	1 Blk Length Upper Follows	1 Blk Length Lower Follows	0 No Port A Upper Addr Follows	1 Port A Lower Addr Follows	1 A"B	0 1 1A Transfer No Search		6D
2 Port A Address Lower 8-bits	0	0	0	0	0	1	0	1	01
3 Block Length Lower 8-bits	0	0	0	0	0	0	0	0	00
4 Block Length Upper 8-bits	0	0	0	1	0	0	0	0	10
5 Command Byte 1B Defines Port A as peripheral with fixed addresses	0 Group 1	0 No Timing Follows	1 Fixed Addresses	X	1 Port is IO	0 This is Port "A"	0 0 1B		
6 Command Byte 1B Defines Port B as a memory with incrementing addresses	0 Group 1	0 No Timing Follows	0 Address Changes	1 Address Increments	0 Port is Memory	1 This is Port "B"	0 0 1B		14
7 Command Byte 2B Sets mode to burst, sets DMA to expect Port B starting address	1 Group 2	1 Burst Mode	0	0 No Int Cont Byte Follows	1 Port B Upper Addr Follows	1 Port B Lower Addr Follows	0 1 2B		CD
8 Port B Address Lower 8-bits	0	1	0	1	0	0	0	0	50
9 Port B Address Upper 8-bits	0	0	0	1	0	0	0	0	10
10 Command Byte 2C Sets Ready Active High	1 Group 2	X	0 No Auto Restart	0 No wait States	1 Rdy Active High	X	1 0 2C		
11 Command Byte 2D loads starting addresses and resets block counter	1 Group 2	1	0	0 Load	1	1	1 1 2A		CF
12 Command Byte 2D Enables DMA to start operation	1 Group 2	0	0	0 ENABLE DMA	0	1	1 1 2D		87

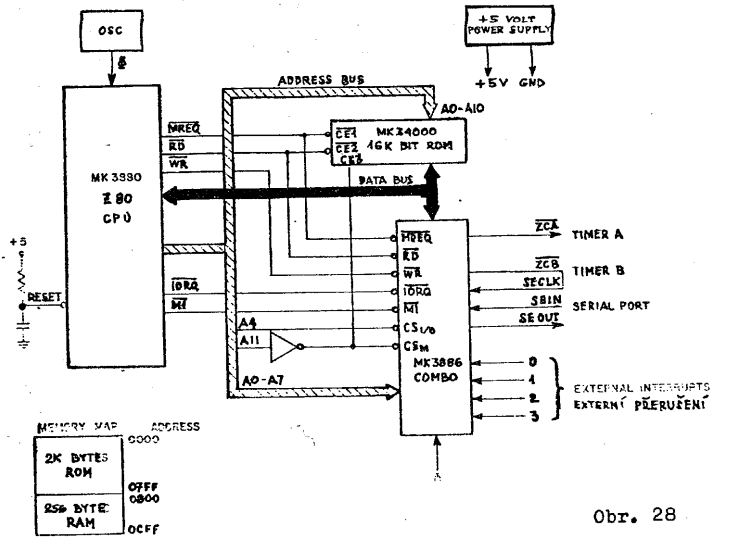
Tab. č. 20 - Příklad programování obratu DMA


Z80-COMBO
RAM
16k×8
+VBB
+5V
logika vnitřního řízení
sériový I/O port
SO
SI
clock
data
adresa
řízení
sběrnicové rozhraní
časovač A
INT 0
ZCA
časovač B
ZCB
logika řízení přerušení
externí přerušení
INT 1
INT 2
INT 3
řízení přerušení

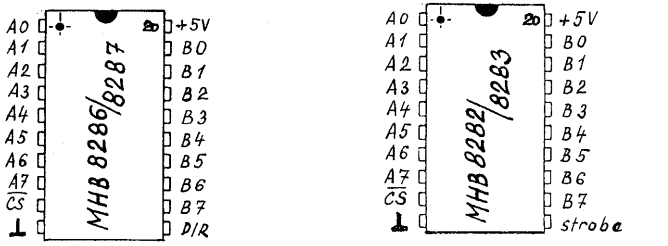
Obr. 26


adresová sběrnice CPU
datová sběrnice CPU
řízení CPU
MK 3886
Z80-
-COMBO
časovač A
časovač B
externí kanály přerušení
sériový I/O port
řízení přerušení
COMBO

Obr. 27



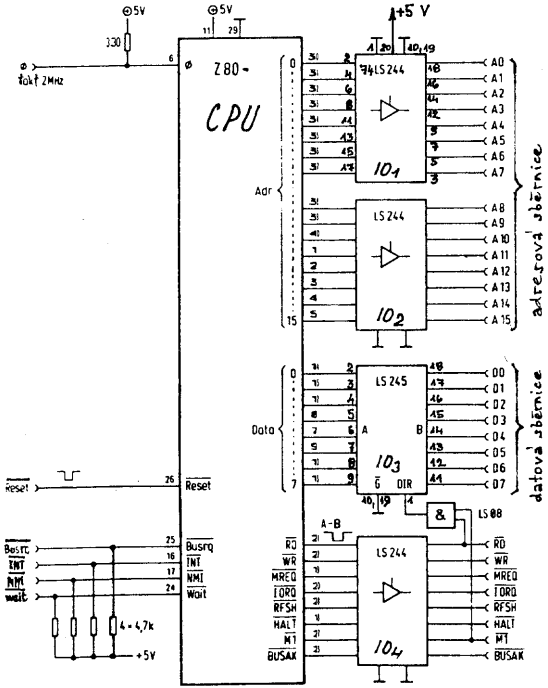
Obr. 28



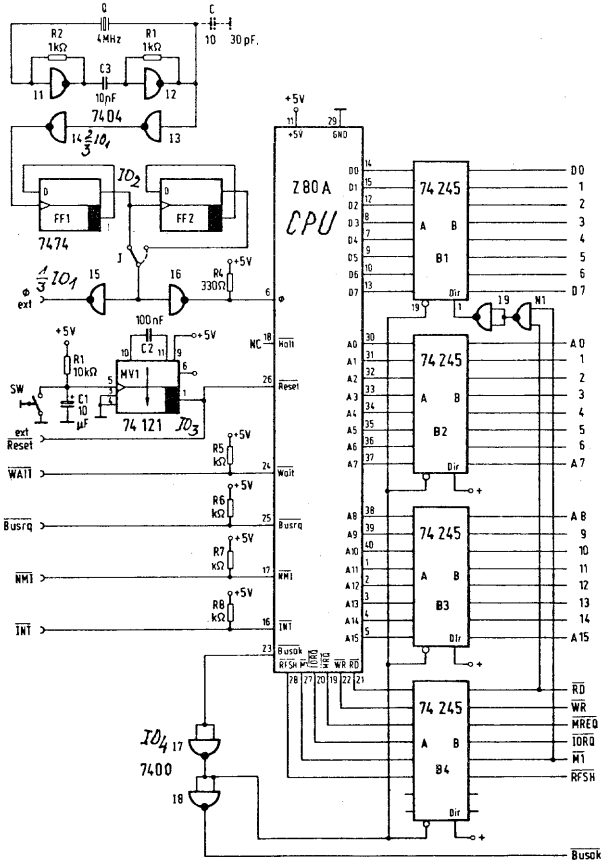
D/R = H ⇒ A → B
L ⇒ B → A

strobe = H ⇒ A → B
L ⇒ B = latch

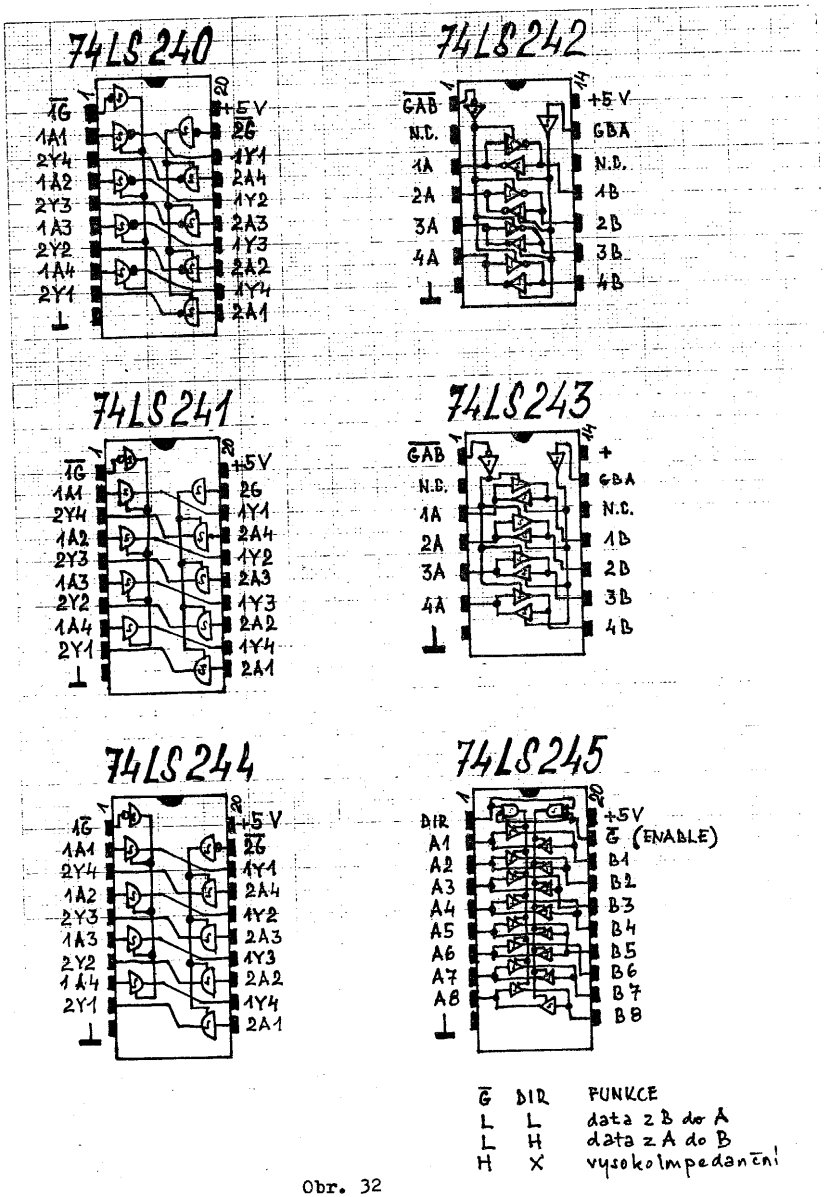
Obr. 35


+5V
330
Takt 2MHz
Z80-CPU
Adr
Data
74LS244
LS244
LS245
LS08
adresová sběrnice
datová sběrnice
Reset
Busrq
INT
NMI
wait
4 - 4,7k
RD
WR
MREQ
IORQ
RFSH
HALT
M1
BUSAK

Obr. 29


Q
4MHz
C
10 - 30 pF
R2 1kΩ
R1 1kΩ
C3 10nF
7404
7474
FF1
FF2
R4 330Ω
100nF
C2
R1 10kΩ
MV1
74 121
C1 10 µF
SW
ext Reset
R5 kΩ
R6 kΩ
R7 kΩ
R8 kΩ
WAIT
Busrq
NMI
INT
Busak
7400
Z80A CPU
74 245
B1
B2
B3
B4
Dir
RD
WR
MREQ
IORQ
M1
RFSH
Busak

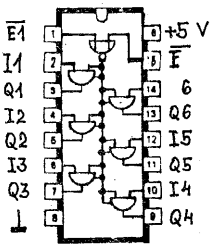
Obr. 30



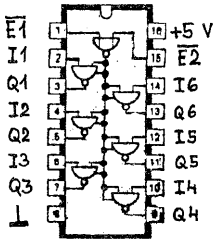
G	DIR	FUNKCE
L	L	data z B do A
L	H	data z A do B
H	X	vysokoimpedanční

Obr. 32

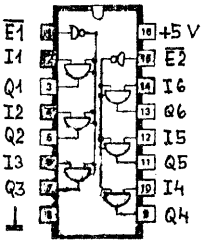
74LS365



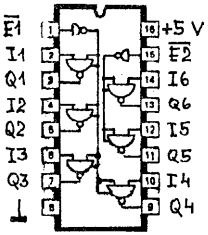
74LS366

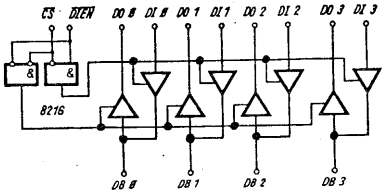


74LS367



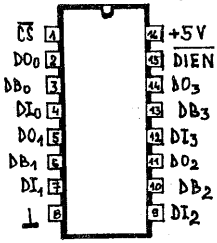
74LS368





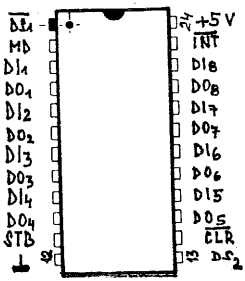
Obr. 31

MH 3216/26

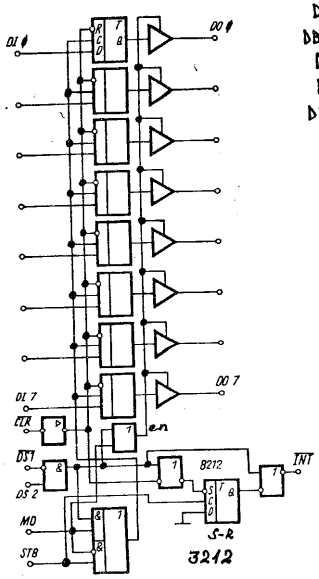


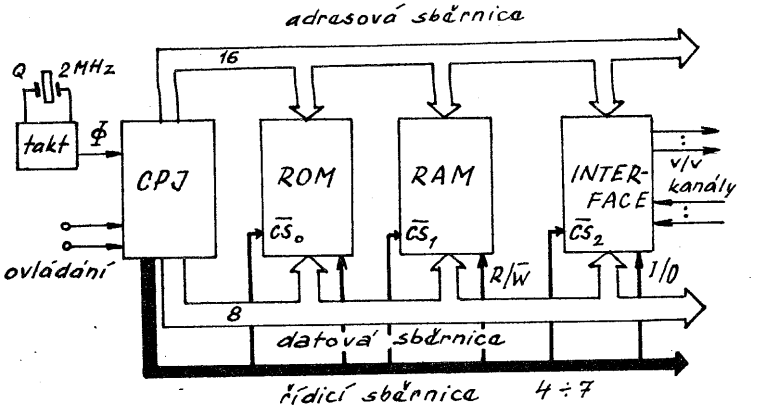
Obr. 33

MH3212



Obr. 34

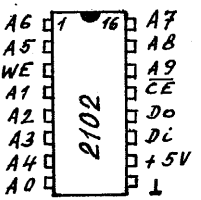




Obr. 36

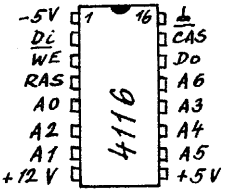
RAM's

statická

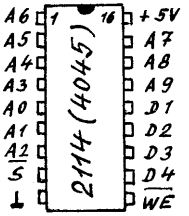


1024 x 1

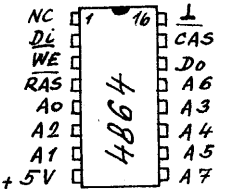
dynamická



16384 x 1



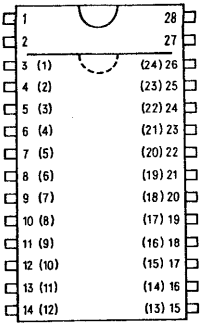
2114 ≡ 4045
1024 x 4



4864 ≡ 4164
65536 x 1

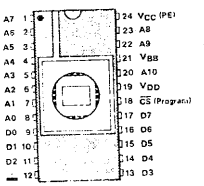
Obr. 38

EPROM 27256	EPROM 27128	CRAM 5564	PRAM 4864	EPROM 2764	EPROM 2732	PRAM 4816	EEPROM 2816	SRAM 4802	CRAM 6116	EPROM 2716	SRAM 4118
V_{PP}	V_{PP}	NC	$\overline{RFS}$	V_{PP}		$\overline{RFS}$					
A12	A12	A12	A12	A12		NC					
A7	A7	A7	A7	A7	A7	A7	A7	A7	A7	A7	A7
A6	A6	A6	A6	A6	A6	A6	A6	A6	A6	A6	A6
A5	A5	A5	A5	A5	A5	A5	A5	A5	A5	A5	A5
A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4
A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3
A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2
A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
A0	A0	A0	A0	A0	A0	A0	A0	A0	A0	A0	A0
D0	D0	D0	D0	D0	D0	D0	D0	D0	D0	D0	D0
D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1
D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND

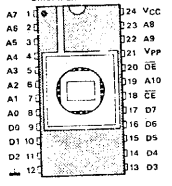


SRAM 4118	EPROM 2716	CRAM 6116	SRAM 4802	EEPROM 2816	PRAM 4816	EPROM 2732	EPROM 2764	PRAM 4864	CRAM 5564	EPROM 27128	EPROM 27256
					V_{CC}		V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
					$\overline{WE}$		PGM	$\overline{WE}$	R/W	PGM	A14
V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	CS	V_{CC}	NC	CS	CE2	A13	A13
A8	A8	A8	A8	A8	A8	A8	A8	A8	A8	A8	A8
A9	A9	A9	A9	A9	A9	A9	A9	A9	A9	A9	A9
$\overline{WE}$	U_{PP}	$\overline{WE}$	$\overline{WE}$	V_{PP}	NC	A11	A11	A11	A11	A11	A11
$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$
$\overline{L}$	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7
D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6
D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5
D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4
D3	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3

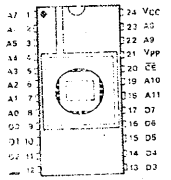
2716
Texas Instruments



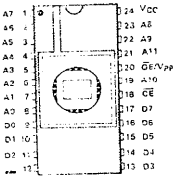
2516
Texas Instruments
2716
other manufacturers



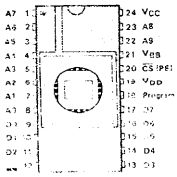
2532



2732



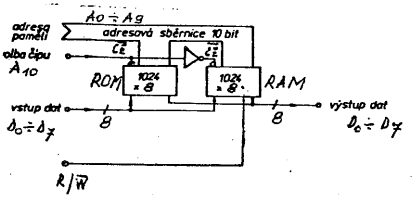
2708



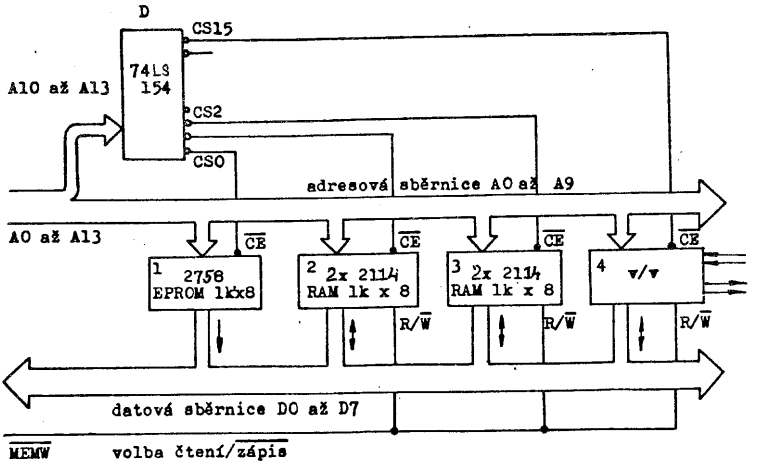
Pin-arrangement

V_{CC} = + 5 V
V_{BB} = − 5 V
V_{DD} = +12 V
V_{PP} = + 5 V in stand-by mode
+25 V in programming mode
$A_0 \ldots A_n$ = Address inputs
$D_0 \ldots D_7$ = Data in- and outputs
CS = Chip Select
CE = Chip Enable
OE = Output Enable
PD = Power Down

Obr. 37

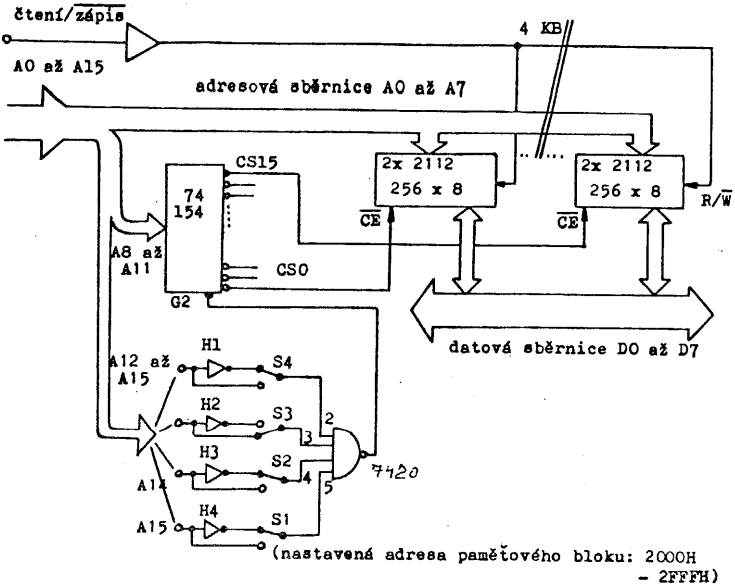


Obr. 39

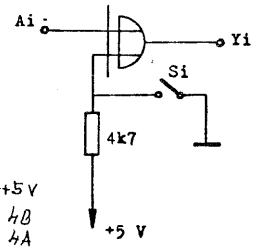


adresový rozsah: 1 - 0000H až 03FFH
2 - 0400H až 07FFH
3 - 0800H až 0BFFH
4 - 3C00H

Obr. 40

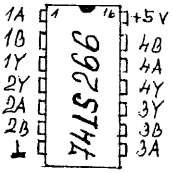


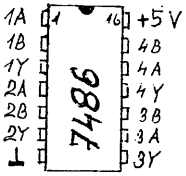
7486 1/4



$Y = A \oplus B = A\bar{B} + \bar{A}B$

Ai	Bi	Yi	Yi - 74LS266
L	L	L	H
L	H	H	L
H	L	H	L
H	H	L	H





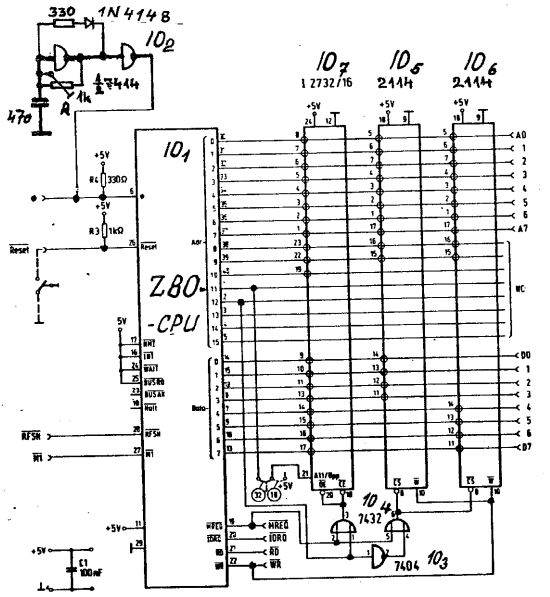
Obr. 41

open collector

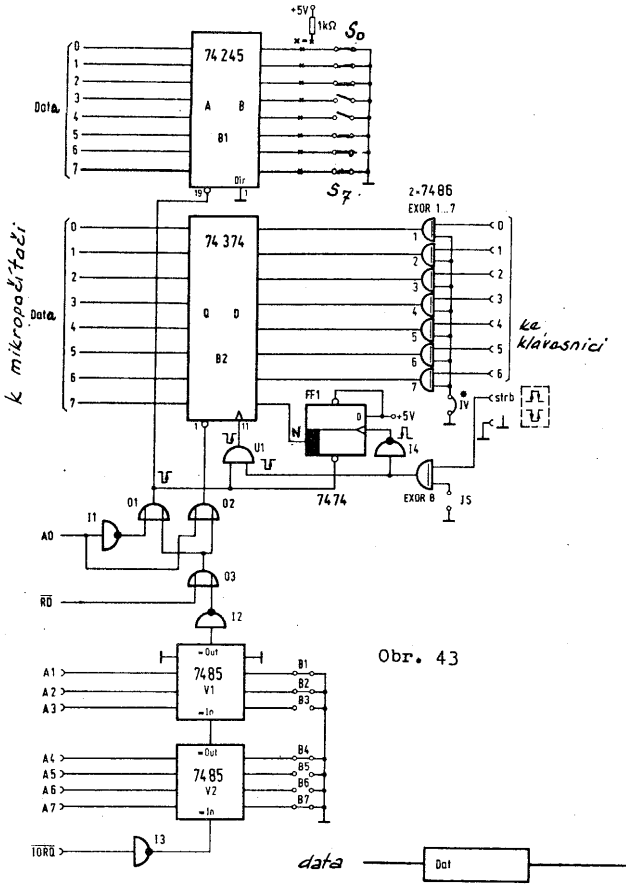
$Y = \overline{A \oplus B} = \overline{A\bar{B} + \bar{A}B}$

7486 ≡ 74LS136 *

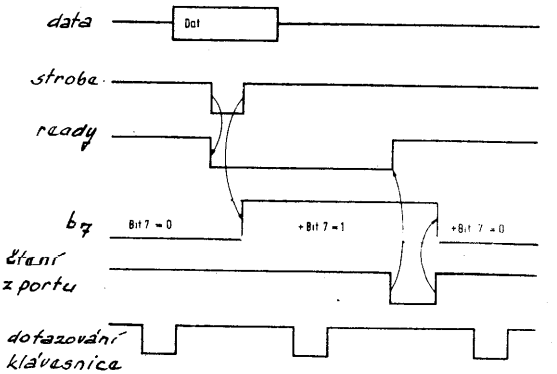
* open


330
1N4148
10_2
74414
1k
A
470
10_7
L 2732/16
10_5
2114
10_6
2114
10_1
Z80-
-CPU
+5V
R4 330Ω
R3 1kΩ
Reset
RFSH
WI
WAIT
BUSRQ
BUSAK
MREQ
IORQ
RD
WR
A0
A7
WC
D0
D7
C1
100nF
10_4
7432
7404
10_3

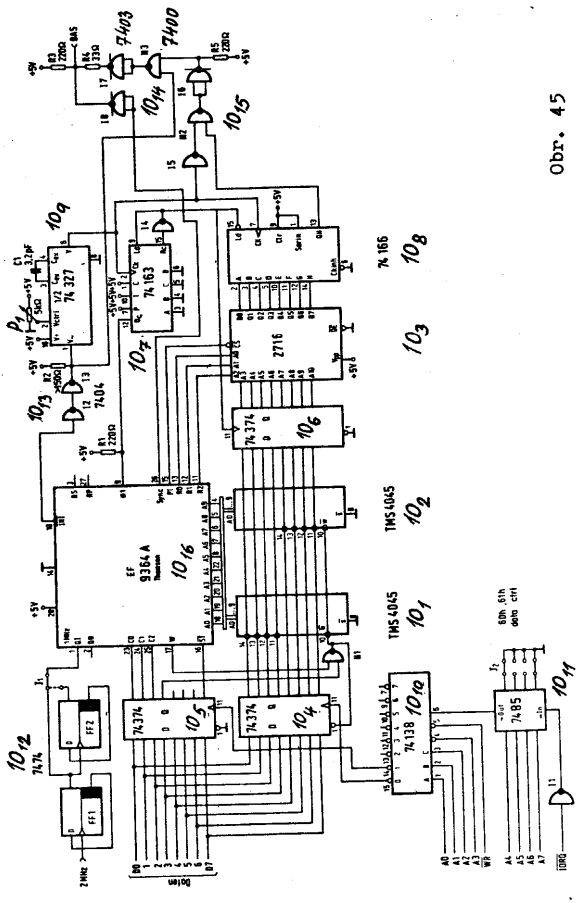
Obr. 42


k mikropočítači
Data
74 245
A B
B1
Dir
+5V
1kΩ
S0
S7
2×7486
EXOR 1...7
74 374
Q D
B2
ke klávesnici
FF1
+5V
strb
JV
U1
7474
EXOR 8
J5
I1
I2
I3
I4
Q1
Q2
Q3
A0
RD
IORQ
7485
V1
V2
A1
A2
A3
A4
A5
A6
A7
B1
B2
B3
B4
B5
B6
B7

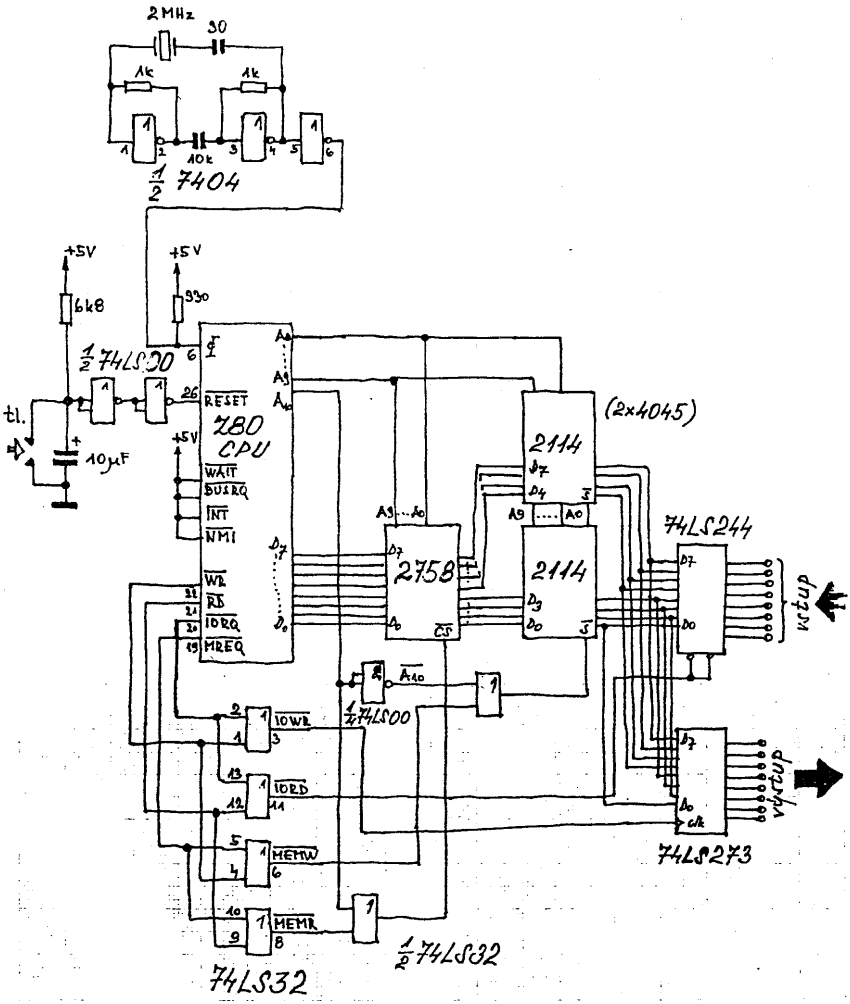
Obr. 43


data
Dat
stroba
ready
b7
Bit 7 = 0
+Bit 7 = 1
+Bit 7 = 0
čtení z portu
dotazování klávesnica

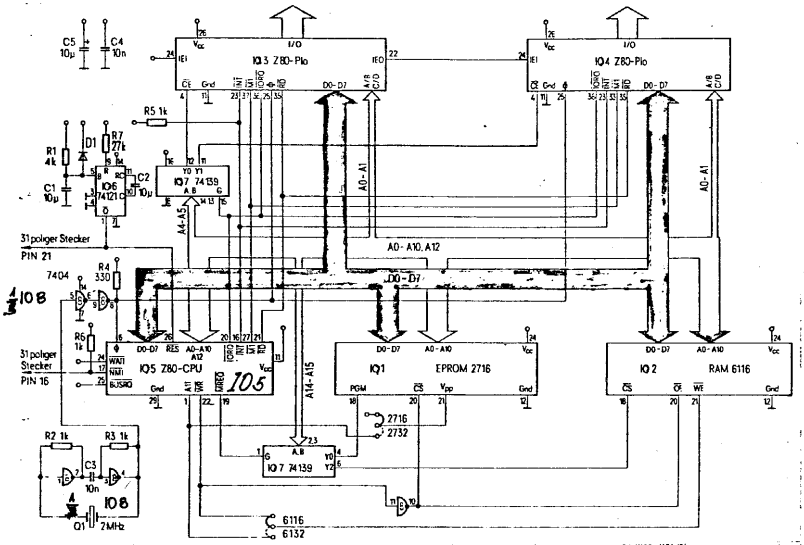
Obr. 44



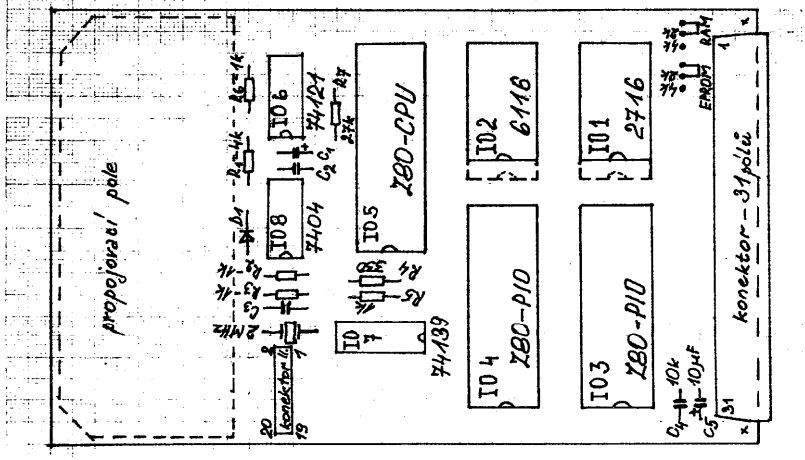
Obr. 45


2MHz
30
1k
1k
10k
1/2 7404
+5V
+5V
648
330
1/2 74LS00
tl.
10µF
RESET
Z80 CPU
WAIT
BUSRQ
INT
NMI
WR
RD
IORQ
MREQ
2758
2114
2114
(2x4045)
74LS244
vstup
výstup
74LS273
1/4 74LS00
IOWR
IORD
MEMW
MEMR
74LS32
1/2 74LS32

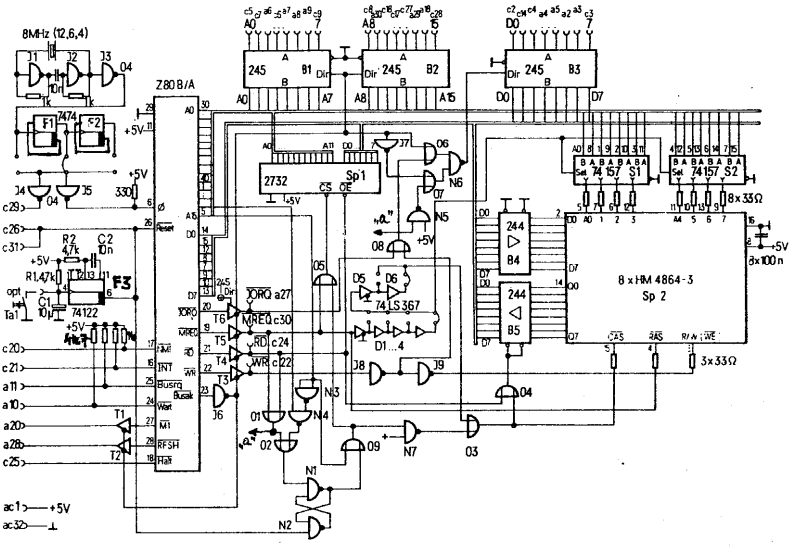
Obr. 46


IO3 Z80-Pio
IO4 Z80-Pio
IO5 Z80-CPU
IO1 EPROM 2716
IO2 RAM 6116
IO7 74139
31 poliger Stecker
PIN 21
PIN 16
D0-D7
A0-A10, A12
A14-A15
2716
2732
6116
6132
7MHz

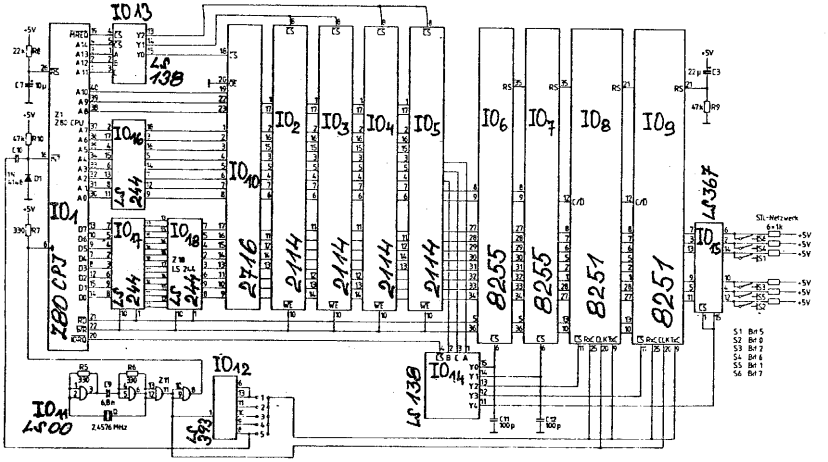
Obr. 47


propojovací pole
IO6
IO8
IO5 Z80-CPU
IO2 6116
IO1 2716
IO4 Z80-PIO
IO3 Z80-PIO
IO7 74139
konektor - 31 polí
konektor

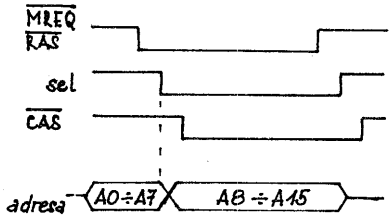
Obr. 48


8MHz (12,6,4)
Z80 B/A
245 B1
245 B2
245 B3
2732
Sp1
74 157 S1
74 157 S2
244 B4
244 B5
8 x HM 4864-3
Sp 2
74 LS 367
74122
8x33Ω
3x33Ω
8x100n
+5V
F1
F2
F3
MREQ c30
IORQ a27
RD c24
WR c22
M1
RFSH
INT
Busrq
Busak
Wait
Reset
opt
Ta1
R1 4,7k
R2 4,7k
C1 10µ
C2 10n

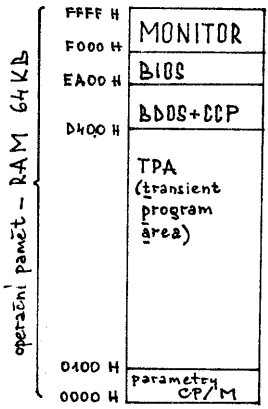
Obr. 50



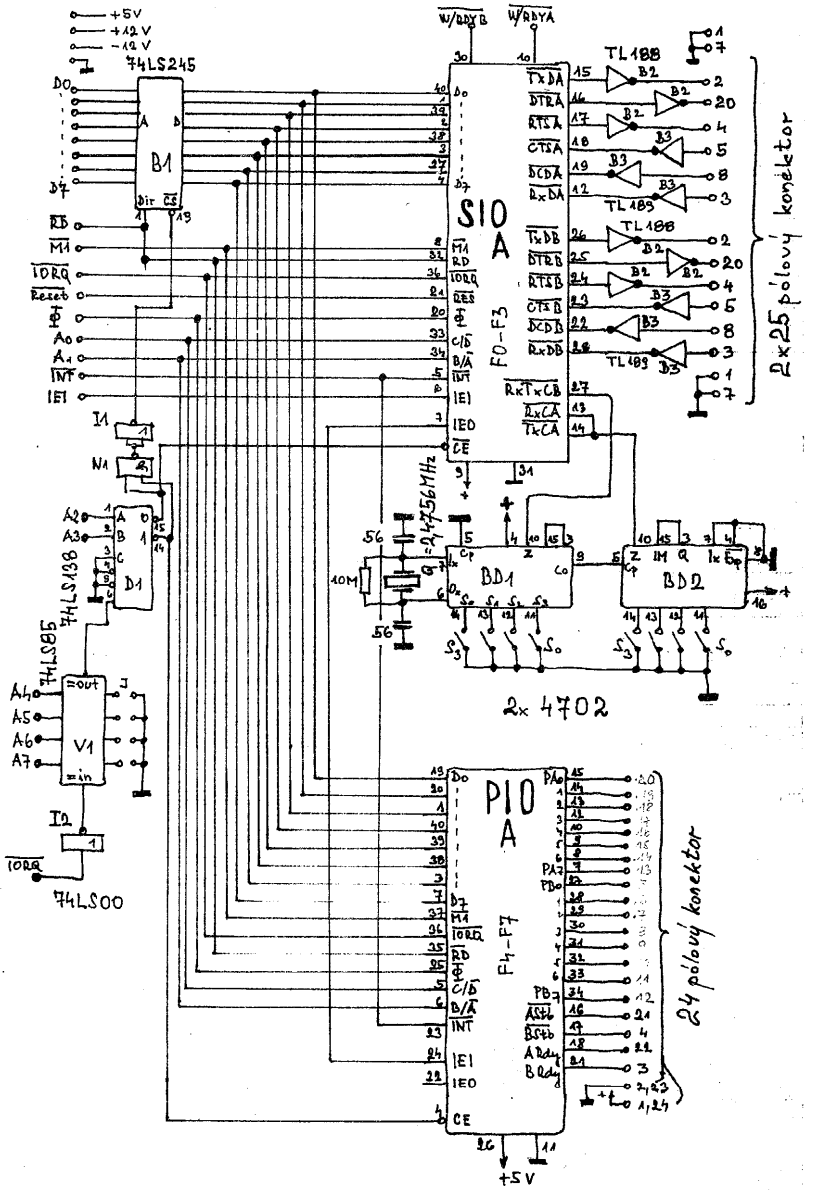
Obr. 49



Obr. 51

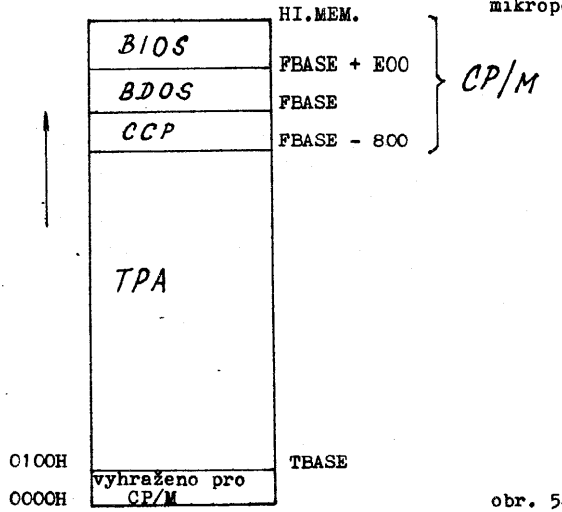


Obr. 53


+5V
+12V
-12V
74LS245
B1
D0
D7
RD
M1
IORQ
Reset
Φ
A0
A1
INT
IEI
I1
N1
A2
A3
74LS138
D1
74LS85
A4
A5
A6
A7
V1
=out
=in
I2
74LS00
W/RDYB
W/RDYA
SIO
A
F0-F3
TxDA
DTRA
RTSA
CTSA
DCDA
RxDA
TxDB
DTRB
RTSB
CTSB
DCDB
RxDB
RxTxCB
RxCA
TxCA
TL188
TL189
2x25 pólový konektor
56
10M
Q=2,4756MHz
BD1
BD2
2x 4702
PIO
A
F4-F7
PA0
PA7
PB0
PB7
24 pólový konektor
+5V

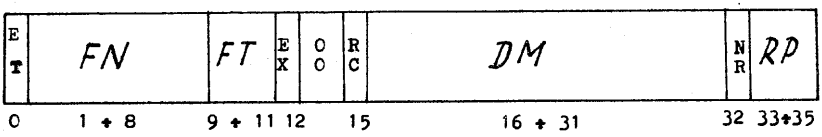
Obr. 52

Obecné umístění provozního systému CP/M v operační paměti mikropočítače

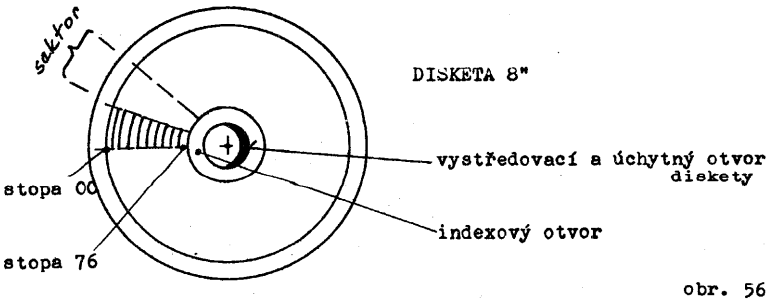


ii obr. 54

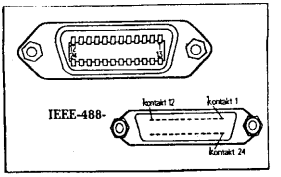
Formát řídicího bloku souboru FCB - file control block



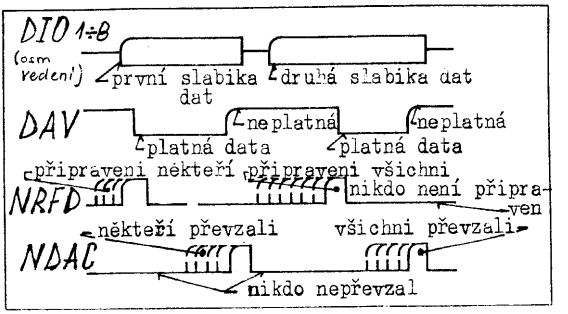
ij obr. 55



ik obr. 56



Obr. 57

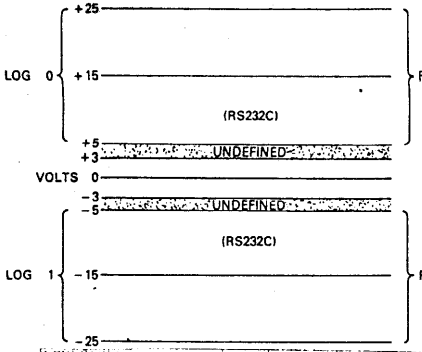


Obr. 58

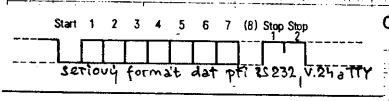
IEC 66.22		IEEE	IEC 66.22		IEEE
kontakt	signál	kontakt	kontakt	signál	kontakt
1	DIO 1	1	14	DIO 5	13
2	DIO 2	2	15	DIO 6	14
3	DIO 3	3	16	DIO 7	15
4	DIO 4	4	17	DIO 8	16
5	REN	17	18	GND	–
6	EOI	5	19	(6)	18
7	DAV	6	20	(7)	19
8	NRFD	7	21	(8)	20
9	NDAC	8	22	(9)	21
10	IFC	9	23	GND	–
11	SRQ	10	24	(11)	23
12	ATN	11	25	(12)	–
13	stínění	12	–	(10)	22

zemnění

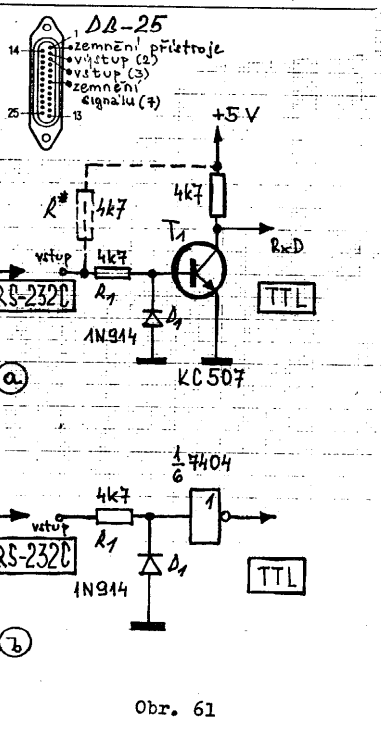
Tab. č. 23



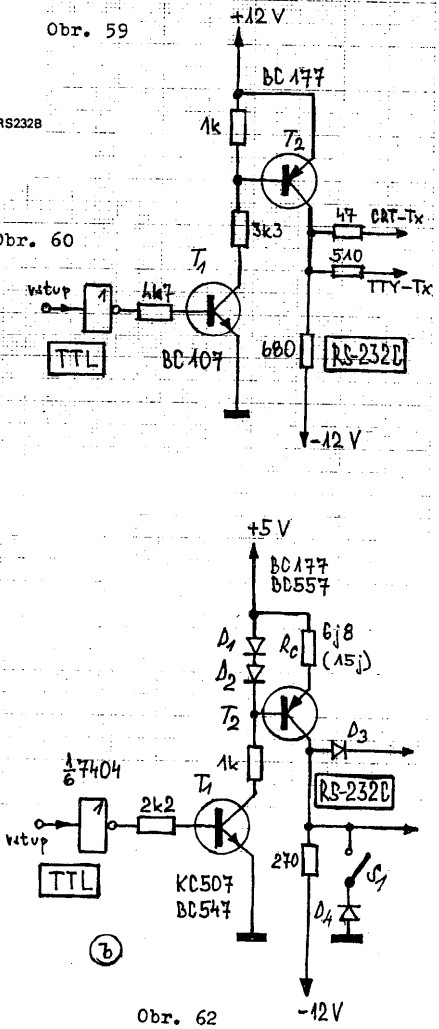
Obr. 59



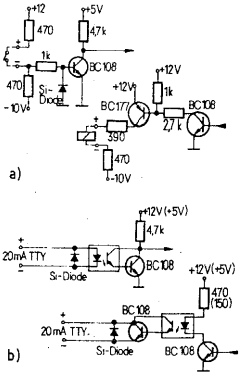
Obr. 60



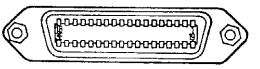
Obr. 61



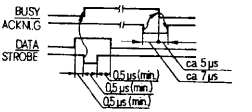
Obr. 62



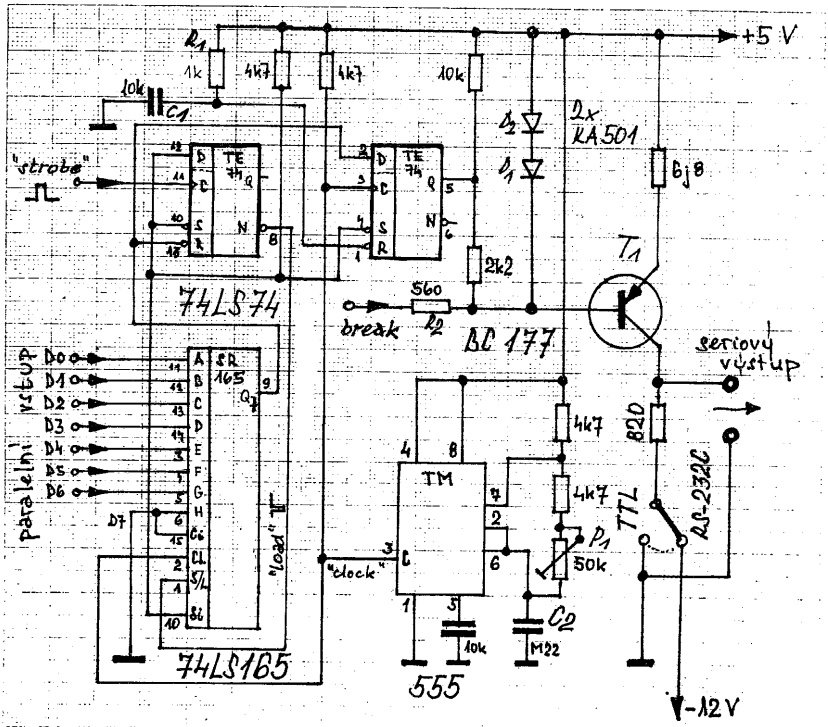
Obr. 63



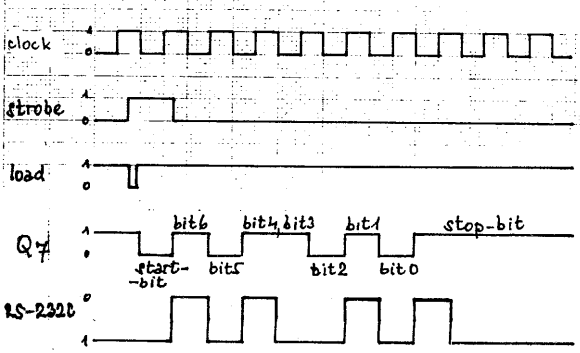
Obr. 64



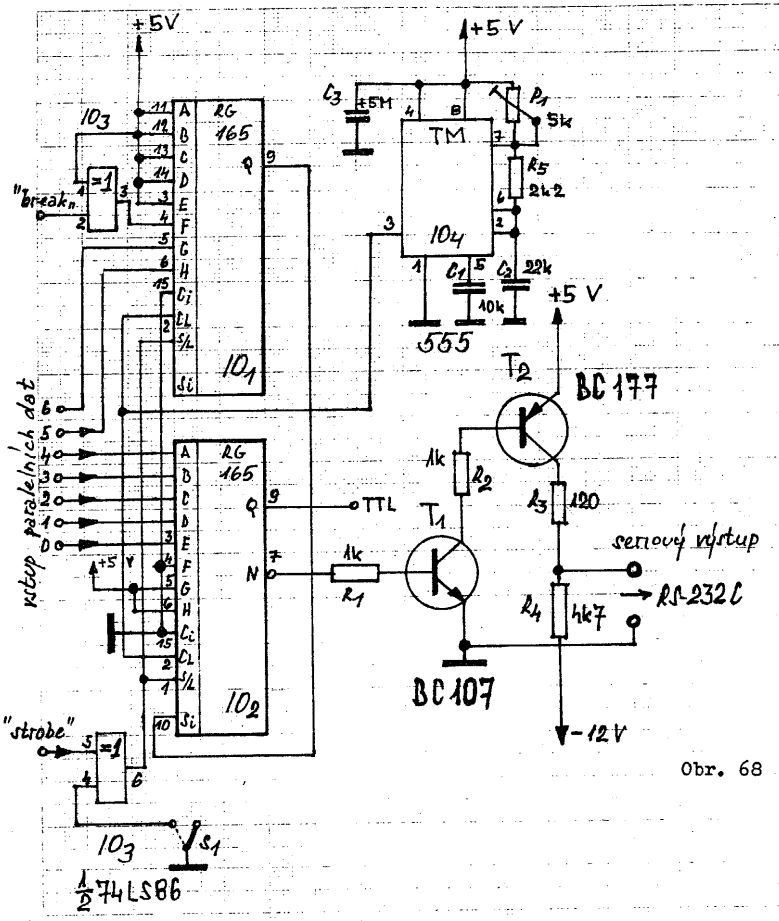
Obr. 65


+5 V
R1
1k
4k7
4k7
10k
10k
C1
"strobe"
74LS74
2x KA501
D2
D1
Bj8
T1
2k2
560
"break"
R2
BC 177
sériový výstup
paralelní vstup
D0
D1
D2
D3
D4
D5
D6
D7
SR 165
Q7
"load"
"clock"
4k7
820
TTL
RS-232C
TM
4k7
P1
50k
C2
10k
M22
555
74LS165
-12 V

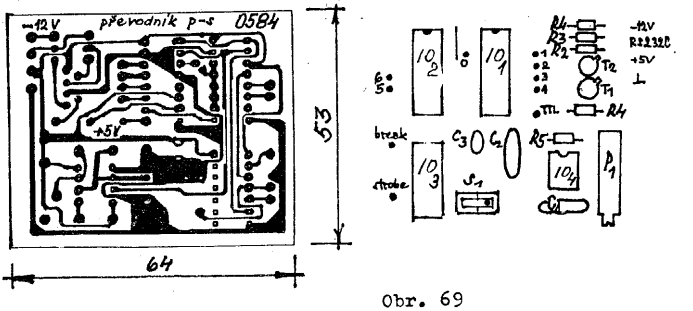
Obr. 66


clock
strobe
load
Q7
bit6
bit4, bit3
bit1
stop-bit
start-bit
bit5
bit2
bit0
RS-232C

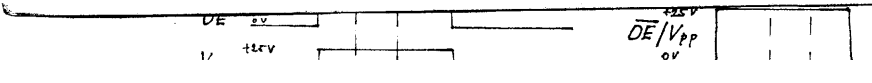
Obr. 67

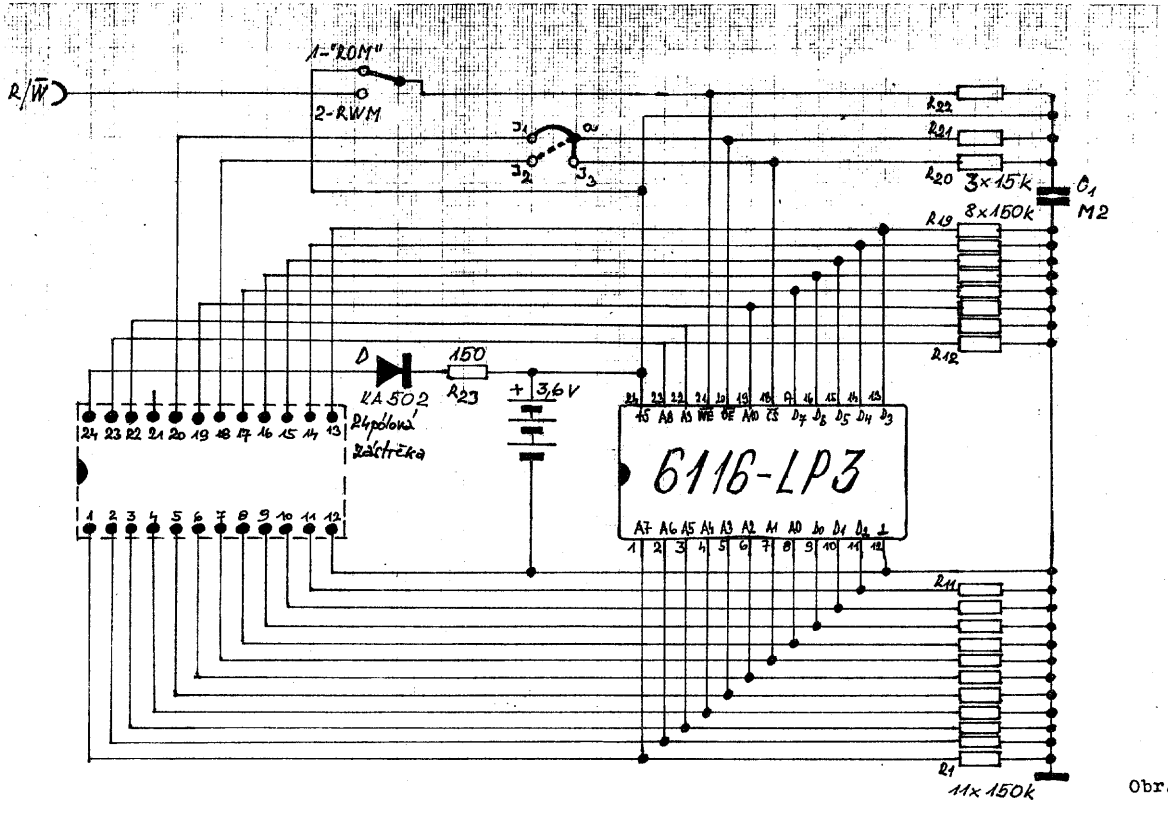

+5V
+5 V
C_3 ±5M
P_1 5k
TM
R_5 2k2
10_4
C_1 10k
C_2 22k
555
10_3
"break"
=1
RG 165
10_1
10_2
vstup paralelních dat
"strobe"
S_1
$\frac{1}{2}$74LS86
TTL
1k
R_1
T_1
T_2
BC 177
BC 107
R_2 1k
R_3 120
R_4 4k7
sériový výstup
RS-232C
-12 V

Obr. 68

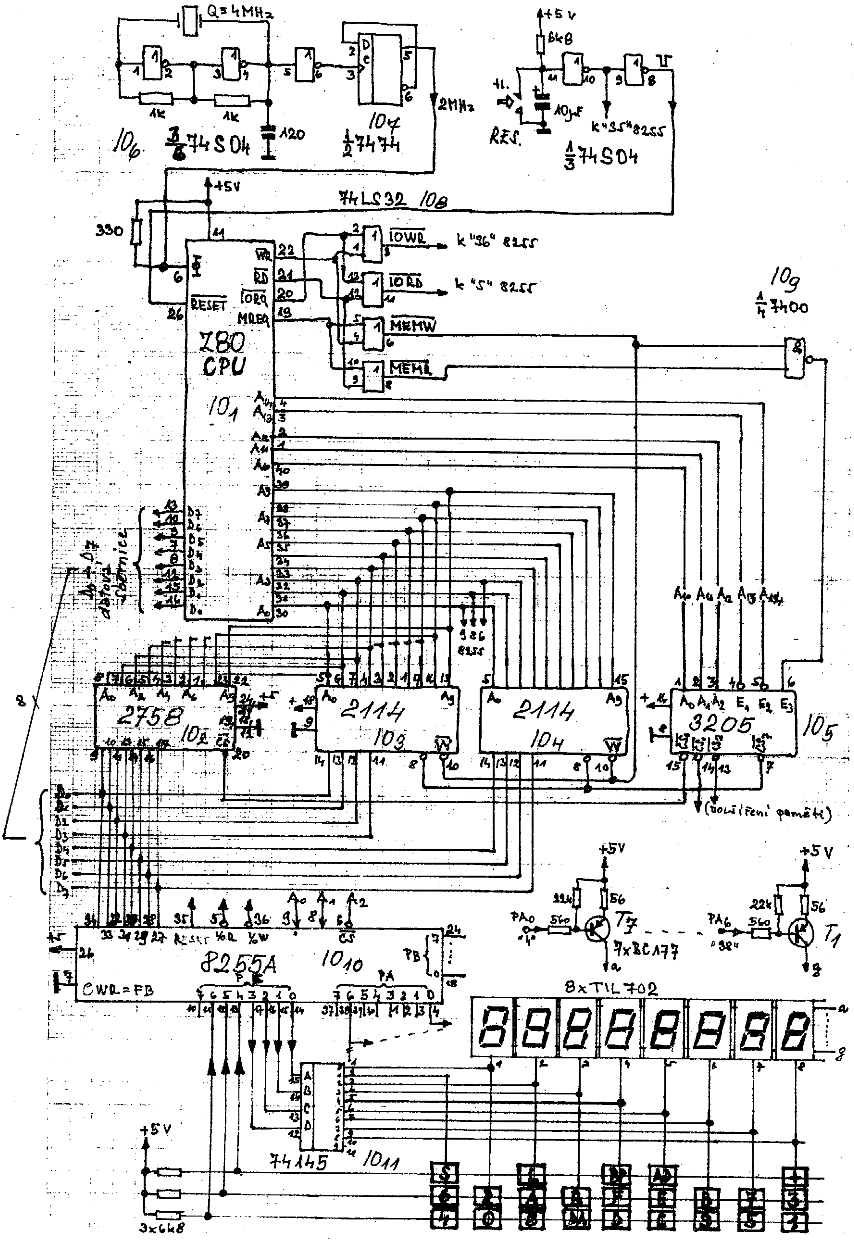

-12V
převodník p-s 0584
+5V
53
64
break
strobe
S_1
C_3 C_4
R_5
P_1
T_1 T_2
R_1 R_2 R_3 R_4
RS232C
+5V
TTL

Obr. 69


OE
0V
+25V
$\overline{OE}/V_{PP}$
0V


1-"ROM"
2-RWM
R/W
J1
J2
J3
R22
R21
R20
3x15k
C1
M2
R19
8x150k
R12
D
KA502
150
R23
+ 3,6V
Objímka
6116-LP3
R11
R1
11x150k

Obr. 70


Q = 4MHz
1k
1k
120
10₆ 3/6 74S04
10₇ 1/2 7474
2MHz
+5V
6k8
10μF
RES.
k "35" 8255
1/3 74S04
74LS32 10₈
330
IOWR
k "36" 8255
IORD
k "5" 8255
MEMW
MEMR
10₉ 1/4 7400
Z80 CPU
10₁
RESET
2758
10₂
2114
10₃
2114
10₄
3205
10₅
(doplnění paměti)
8255A
10₁₀
CWR = FB
8xTIL702
7xBC177
T₇
T₁
74145
10₁₁
3x6k8
+5V

Obr. 73

Název publikace: MIKROPROCESOR Z80 A JEHO APLIKACE

Zpracoval: Kolektiv autorů

Stran: 234 Náklad: 95 výtisků Formát: A 5
Číslo publikace: DT 2823
Vydal a vytiskl: Dům techniky ČSVTS Praha,
Gorkého nám. 23, Praha 1 112 82
Rok vydání: 1984 DT 01 - 540/84

Bez jazykové úpravy DT