KNIŽNICE ČSVTS
MIKROPROCESOROVÁ TECHNIKA

VLADIMÍR DVOŘÁK

MIKROPROCESOR Z 80 A PROGRAMOVATELNÉ OBVODY

1988

ÚVOD

Počátek sedmdesátých let přináší kvalitativní změnu do oboru výpočetní techniky. Dosavadní počítače a minipočítače, které byly cenově, rozměrově, spotřebou energie, spolehlivostí atd. nevýhodné, jsou nahrazovány mikropočítači. Mikropočítač je osazen mikroprocesorem a několika dalšími prvky vysoké integrace. Takto vytvořený systém na LSI prvcích vyniká řadou předností např. nízkou spotřebou energie, malou rozměrovostí,zmenšenými náklady na výrobu desek s plošnými spoji, vysokou spolehlivostí, modulovou konstrukcí, rychlou opravitelností. V současné době je ve světě vyráběno několik desítek typů mikroprocesorů řezového i monolitického charakteru, velké množství podpůrných obvodů a stovky typů polovodičových pamětí.

V tuzemsku směřuje mikroelektronika od řezového bipolárního systému řady MH 3000, přes unipolární monolitický mikroprocesor MHB 8080A, k jednočipovým mikropočítačům MHB 8035 a MHB 8048. Zatím největší rozšíření mají v ČSSR mikropočítačové stavebnice na bázi mikroprocesoru MHB 8080A. Omezenou domácí nabídku lze doplnit dovozem ze zemí RVHP. Např. NDR vyrábí osmibitový mikroprocesor UB 880D, jenž je plně ekvivalentní mikroprocesoru firmy Zilog Z80-CPU. Zvládnutí tohoto technicky vyspělejšího mikroprocesoru než MHB 8080A dává reálné předpoklady pro výrobu jednočipových mikropočítačů typu Z-8 nebo šestnáctibitových mikroprocesorů řady Z-8000. Ze SSSR se dováží perspektivní šestnáctibitový mikroprocesor KA1808VM86 , který je ekvivalentní I 8086. BLR dodává systém CM 600, mající analog v osmibitovém systému firmy Motorola MC 6800.

Publikace je zaměřena na podrobný popis mikroprocesoru Z80-CPU včetně jeho podpůrných obvodů, které maximálně rozšiřují využití centrální mikroprocesorové jednotky, patřící mezi nejlepší osmibitové mikroprocesory s mnoha moderními koncepčními prvky. Tuto skutečnost potvrzuje i převzetí celé stavebnice firmou Mostek a SGS-ATES.

Cílem rozvoje a výroby mikroelektronických prvků je zabezpečení automatizace národního hospodářství ve všech sférách. Dnes se již nejedná o jednotlivé aplikace mikroelektroniky, ale o jejich hromadný průnik, který přinese změnu kvality výroby, technologie a služeb v mnoha odvětvích. Automatizace v průmyslu přináší úspory ekonomické a sociální. Typickým příkladem automatizačních prostředků jsou NC (NUMERICAL CONTROL) a CNC (COMPUTERIZED NUMERICAL CONTROL) systémy pro řízení obráběcích strojů. Aplikační proces zasáhl i oblast robotizace. Tímto způsobem lze vyloučit pracovní síly z výrobního procesu tam, kde jde o prostředí obtížné nebo zdraví škodlivé.

1. ZÁKLADNÍ POJMY Z MIKROPOČÍTAČOVÉ TECHNIKY

1.1 Mikropočítačový systém

Technologický pokrok v bipolárních a hlavně v unipolárních technikách umožnil koncem šedesátých let konstrukce prvních kalkulátorových čipů a polovodičových pamětí v LSI hustotách. Hustota LSI (LARGE SCALE INTEGRATION), tj. technika s vysokou hustotou integrace pohybující se od 1 000 ÷ 10 000 aktivních prvků na čipu, dovolila v sedmdesátých letech realizovat v jediném pouzdře integrované obvody s velmi složitou centrální jednotkou a nejenom ji, ale také řadu dalších komplikovaných obvodových struktur, jež tvoří stavební prvky počítačového systému. Vznikl tak nový směr ve vývoji výpočetní techniky a také nová terminologie, která tento vývoj odráží.

Vedle různých centrálních jednotek v jednom pouzdře integrovaného obvodu vysoké integrace se začal tvořit i druhý směr v konstrukci centrálních jednotek, a to z důvodů rychlosti jejich práce. Jedná se o stavebnicové centrální jednotky, které lze skládat ze stavebnicových prvků tzv. řezů. Od zmíněného začátku sedmdesátých let oba tyto směry vývoje technologie a techniky integrovaných obvodů podstupují intenzivní rozvoj a mají své specifické rysy, jimiž se podstatně odlišují.

Termínem monolitický mikroprocesor se označuje určitá forma centrální jednotky mikropočítače, umístěná v jediném pouzdře integrovaného obvodu, zatímco termínem řezový mikroprocesor se označuje soustava integrovaných obvodů tj. řezů, které spolu s pamětí mikroprogramů vytvářejí stavebnicovou centrální jednotku s repertoárem instrukcí, který si definuje pomocí mikroprogramů uživatel. U monolitických mikroprocesorů je repertoár instrukcí jednou provždy dán výrobcem daného pouzdra. Koncernový podnik Tesla Rožnov vyrábí od roku 1980 řezový systém MH 3000 a koncernový podnik Tesla Piešťany od roku 1983 monolitický osmibitový systém MHB 8080A. Ekvivalentní systémy vyráběné firmou Intel jsou označeny I 3000 a I 8080A.

Mezi oběma směry vývoje mikroprocesorů existují styčné body a uvedená vysvětlení by pak bylo nutno určitým způsobem modifikovat. Vzhledem k rozsahu studované oblasti a počtu různých výrobců mikroprocesorů není možné se jimi zabývat. Zatímco mikroprocesor, ať monolitický nebo řezový, tvoří pouze centrální jednotku, je nutno k tomuto mikroprocesoru přiřadit ještě celou řadu podpůrných obvodů a polovodičových pamětí, které jsou opět obvody vysoké integrace, aby vznikl počítačový systém, který by byl schopen zpracovávat program uložený v paměti, řídit periferní zařízení, styk s člověkem, přenos dat apod. Takový systém se nazývá mikropočítač neboli mikropočítačový systém.

Podpůrné obvody řídí hlavně styk mikroprocesoru s periferními jednotkami a prostředím, ale slouží také jako řadiče sběrnic, dekodéry, registry, obvody přerušení, DMA řadiče, časovače. Hrubé členění podpůrných obvodů je možné provést do třech skupin:

a) základní podpůrné obvody - jsou prvky, které dotvářejí jádro mikroprocesorového systému. Jejich funkce navazuje na práci mikroprocesoru, který ne vždy zajistí veškeré nezbytné činnosti. Nejčastěji se jedná o časovací obvody a systémové řadiče např. I 8224 a I 8228 pro mikroprocesor I 8080A.

b) neprogramovatelné podpůrné obvody - tvoří velkou skupinu obvodů, jejichž funkce je dána vnitřní stavbou a je částečně modifikovatelná zapojením vstupních řídicích signálů např. I 8205, I 8212, I 8214, I 8216, I 8226. U nás jsou vyráběny pod označením MH 3205, MH 3212, MH 3214, MH 3216, MH 3226. Prvky slouží jako dekodéry, registry, budiče, řadiče přerušení, komunikační obvody.

c) programovatelné podpůrné obvody - funkci těchto prvků definuje sada řídicích neboli programovacích slov CW (CONTROL WORD), která jsou programově dodána z iniciativy mikroprocesoru a plní řídicí registry obvodů CWR (CONTROL WORD REGISTER). Největší zastoupení mají obvody stykové, přes které jsou připojeny paralelně nebo sériově pracující periferie, akční členy atd. V tuzemsku jsou vyráběny stykové komunikační obvody

MHB 8251, MHB 8255A a MHB 1012 (C). Zahraniční výrobci dodávají širší sortiment programovatelných prvků např. I 8253, I 8257, I 8259 spolu s komunikačními obvody I 8251 a I 8255A.

Program a data jsou uloženy v polovodičových pamětech, které nahradily dříve používané paměti feritové nebo paměti na tenkých magnetických vrstvách. Jedinou nevýhodou operačních polovodičových pamětí proti feritovým je, že ztrácejí svůj obsah po odpojení napájení. Ve všech ostatních parametrech předčí paměti feritové. Polovodičové paměti různých typů a struktur umožňují vytváření širokých matic, které plně pokrývají možnosti mikroprocesorů. Operační charakter mají paměti RWM (RAM), pevné paměti jsou označeny symboly ROM, PROM, EPROM. Podrobný rozbor struktur a typů pamětí podává kapitola 1.11.

U monolitických mikroprocesorů výše uvedené prvky propojuje paralelně rozvedený soubor vodičů tzv. systémová sběrnice (SYSTEM BUS). Část vodičů je určena pro přenos datových informací - datová sběrnice (DATA BUS), většinou má obousměrný charakter. Adresované místo, z kterého jsou data čerpána nebo kam jsou data ukládána, je lokalizováno stavem výstupní adresové sběrnice (ADDRESS BUS). Směr přenosu tj. řízení paměti a podpůrných komunikačních obvodů specifikuje výstupní řídicí sběrnice (CONTROL BUS). Zjednodušené blokové schéma mikropočítačového systému je na obr. 1. Na závěr lze obrazně říci, že mikroprocesor vytváří srdce celého mikropočítače a jeho páteří se stává systémová sběrnice spojující veškeré uzly.

1.2 Struktura mikroprocesoru

Obr. 2 ukazuje obecný blíže nespecifikovaný osmibitový mikroprocesor, který sestává z těchto hlavních bloků:

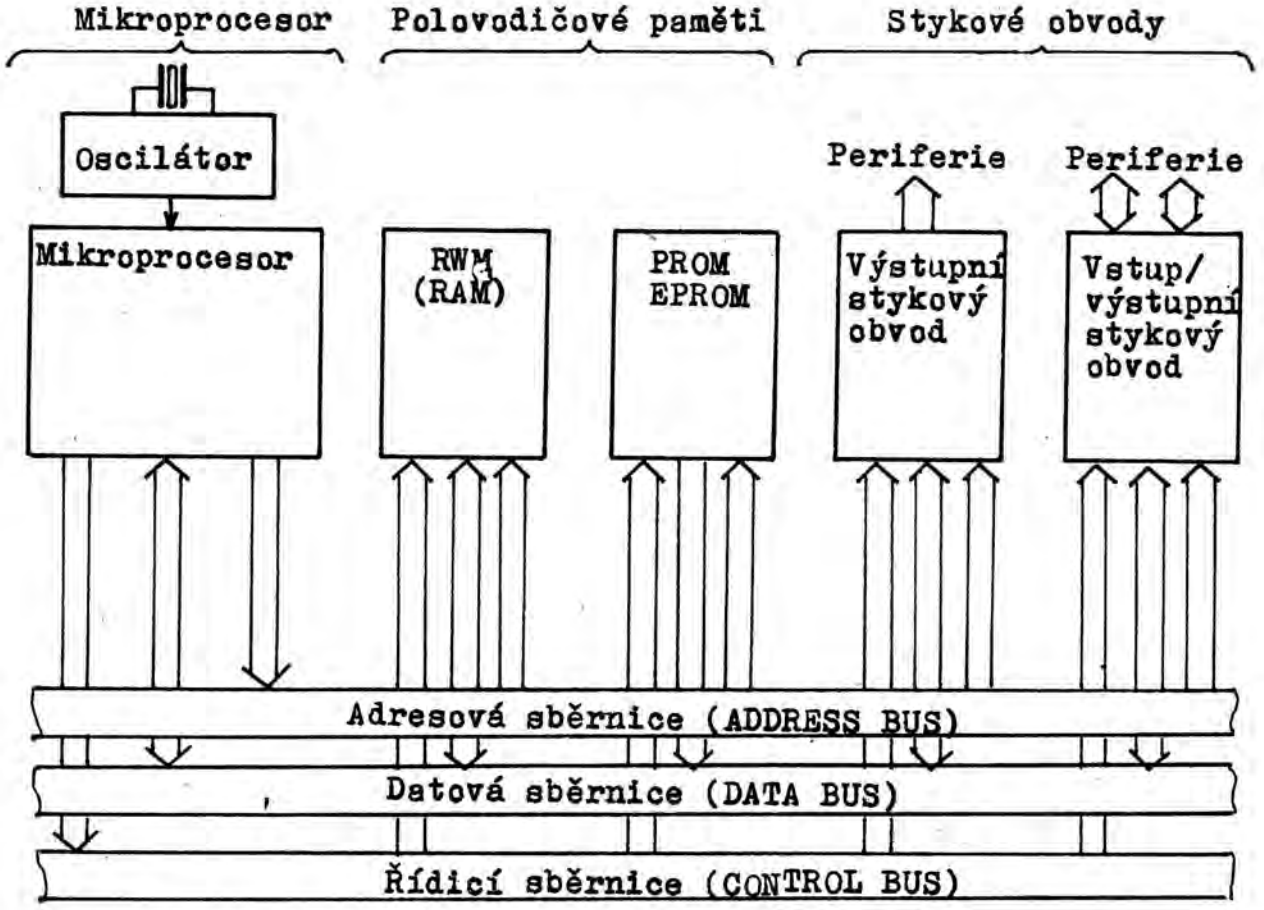
a) pracovní soubor registrů

b) řadič

c) ALU

d) bloku řízení datové, adresové a řídicí sběrnice

e) interní datové sběrnice.

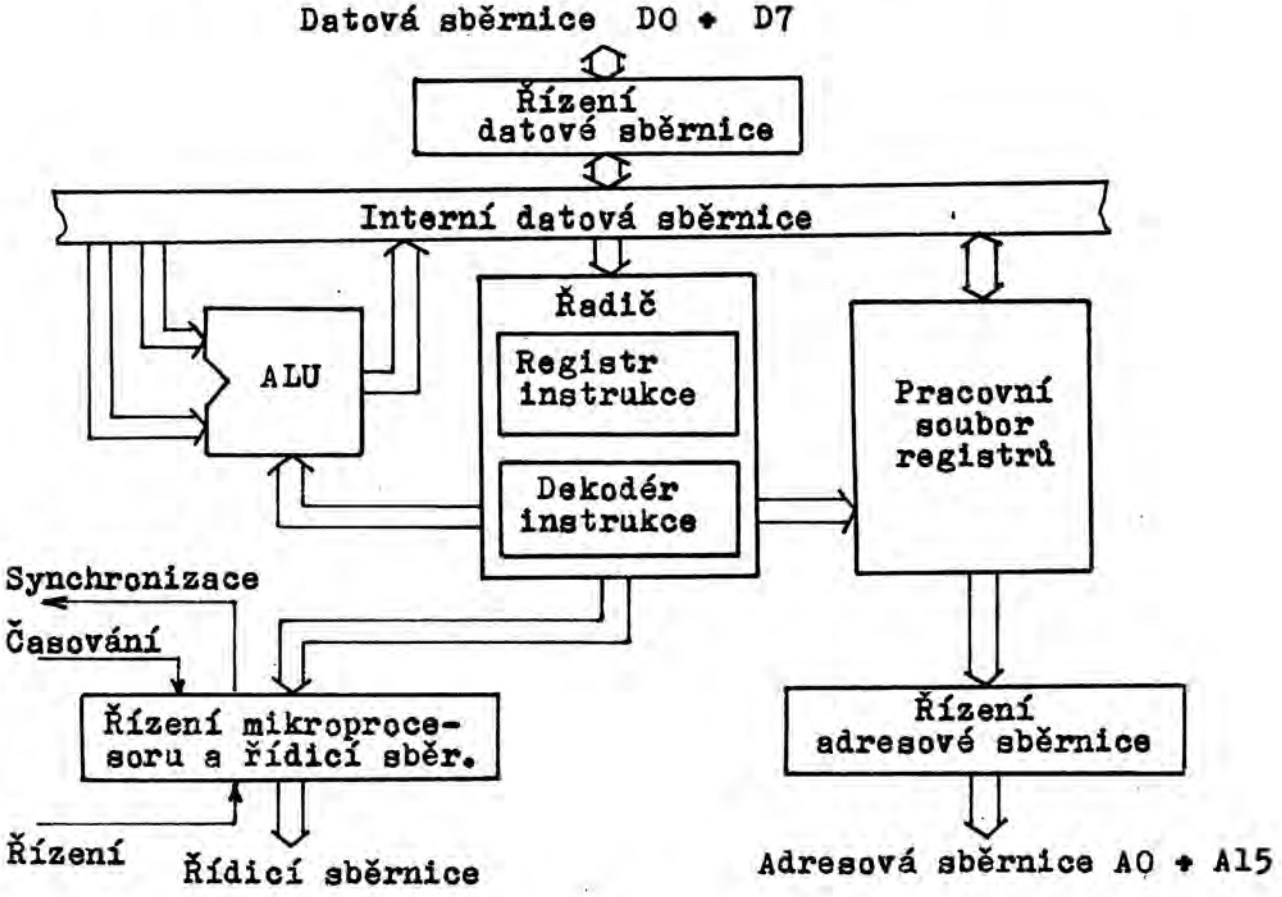


Obr. 1 Zjednodušené blokové schéma mikropočítačového systému

Následuje objasnění jednotlivých bloků mikroprocesoru:

a) pracovní soubor registrů - výklad o funkci pracovních registrů mikroprocesoru bude lépe zaměřit přímo na strukturu mikroprocesoru Z80-CPU (viz obr. 3). Z80-CPU obsahuje soubor osmi základních pracovních registrů A, F, B, C, D, E, H, L a osmi záložních registrů A', F', B', C', D', E', H', L'. Oba soubory jsou identické. Po vynulování systém pracuje se základním souborem. Přechod do souboru záložního se děje programově. Záložní soubor registrů se uplatní ve zrychlení obsluhy přerušení bez nutnosti využití vnější zásobníkové paměti v případě, že obslužný program není znovu přerušován. Využití registrů B, C, D, E, H,L je univerzální a slouží především k přechodnému uložení dat a výsledků operací. Ostatní registry osmibitového nebo šestnáctibitového charakteru mají význam speciální. Jedná se o registry A, F, I, R, IX, IY, SP, PC. Šestnáctibitové registry SP a PC

adresují paměť, a to oblast tzv. zásobníku adresový registr SP (STACK POINTER), zbylou část paměti, kde jsou data a program adresuje registr PC (PROGRAM COUNTER). Způsob práce registrů SP a PC je odlišný. Registr A (ACCUMULATOR) neboli střadač je nejfrekventovanějším registrem a používá se pro zanesení jednoho z operandů do ALU. Výsledek operace je možné uchovat opět ve střadači. V indikátorovém registru F (FLAG REGISTER) jsou uloženy na základě provedení některých instrukcí důležité příznaky, které jiné instrukce potřebují ke svému provedení. Ostatní instrukce vůbec tohoto registru nevyužívají. Indexové registry IX, IY (INDEX REG) se používají při indexovém adresování paměti. Obnovu informace v dynamických pamětech zajišťuje registr R (MEMORY REFRESH), jehož systematicky inkrementovaný stav R0 ÷ R6 je při každém načtení operačního kódu instrukce vyslán na adresovou sběrnici A0 ÷ A6. Registr I (INTERRUPT



Obr. 2 Zjednodušené blokové schéma mikroprocesoru

VECTOR) tvoří vyšší část ukazatele adresy na tabulku počátečních adres obslužných programů přerušení za předpokladu, že Z80-CPU pracuje v přerušení režimu 2 (MODE 2).

b) řadič - funkci bloku řadiče je obtížné specifikovat, poněvadž jeho řídicí obvody zasahují do všech částí mikroprocesoru. Přesto lze v řadiči identifikovat registr instrukce, kde je uchován operační kód instrukce, použitý pro dekódovaní v dekodéru instrukce. Operační kód instrukce je v řadiči přítomen po celou dobu načítání i provedení instrukce. Vygenerované signály z dekodéru instrukce řídí činnost ostatních bloků mikroprocesoru.

c) ALU (ARITHMETIC LOGIC UNIT) aritmeticko-logická jednotka - provádí aritmetické a logické operace s operandy, které zleva do ní vstupují z interní datové sběrnice. Výsledek operace, který může ovlivnit také indikátorový registr, vystupuje vpravo z ALU a je přes interní datovou sběrnici zanesen do střadače nebo jiného pracovního registru. Některé instrukce např. přesunové s ALU nepracují.

d) blok řízení datové, adresové a řídicí sběrnice - blok řízení datové sběrnice řídí vstup/výstup dat na obousměrnou

REG A	FLAG F	REG A´	FLAG F´
B	C	B´	C´
D	E	D´	E´
H	L	H´	L´

ZÁKLADNÍ REGISTRY ZÁLOŽNÍ REGISTRY

INTERRUPT VECTOR I	MEMORY REFRESH R
INDEX REG IX	
INDEX REG IY	
STACK POINTER SP	
PROGRAM COUNTER PC	

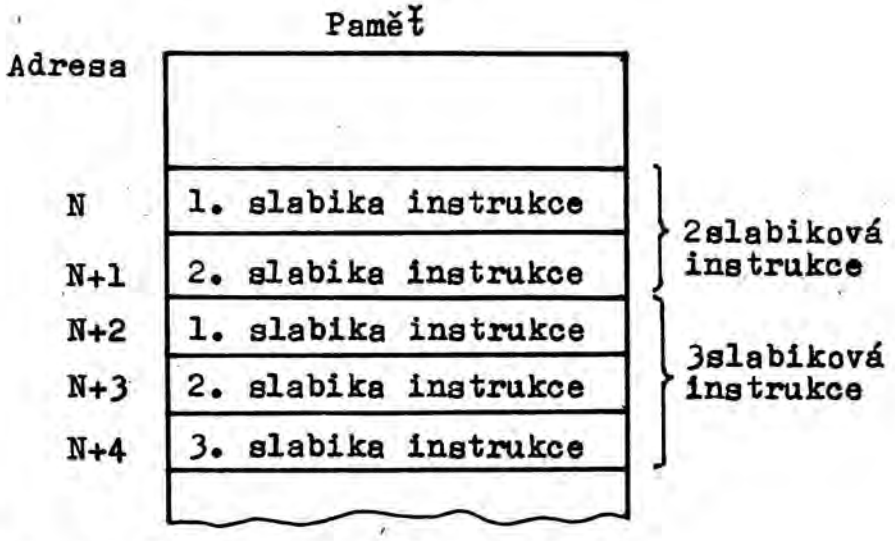
Obr. 3. Pracovní soubor registrů Z80-CPU

osmibitovou datovou sběrnici. Řízení adresové sběrnice řídí výstup adresy na šestnáctibitovou výstupní sběrnici. Konstrukce těchto bloků spolu s blokem řízení řídicí sběrnice je řešena třístavově, aby bylo možné sdílet sběrnice více řídicími prvky (DMA řadič, další mikroprocesor).

e) interní datová sběrnice - hlavní celky mikroprocesoru jsou propojeny interní datovou sběrnicí. Sběrnice je k dispozici pro vnitřní komunikaci.

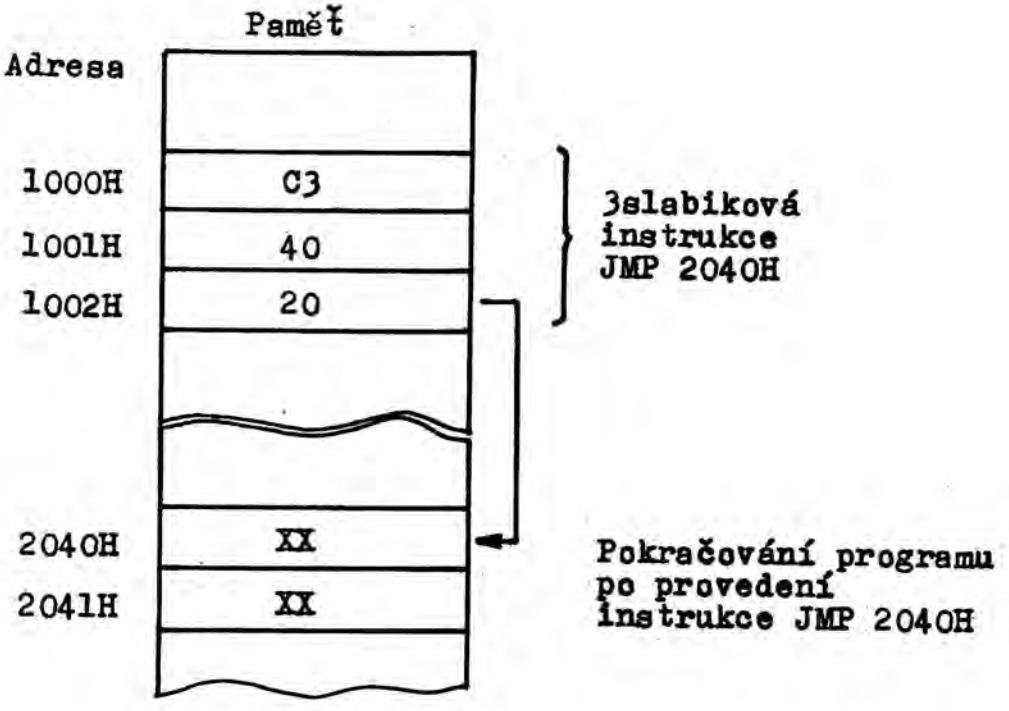
1.3 Načtení a zpracování instrukce

Program, sestavený z jednotlivých instrukcí (příkazů), je uložen v paměti mikropočítače. Každé místo paměti je adresováno z iniciativy mikroprocesoru. Instrukce, které mohou být jednoslabikové i víceslabikové jsou umístěny vzestupně po adresách. Programový čítač PC (PROGRAM COUNTER) zajišťuje tuto adresaci a vždy v předstihu je v něm připravena adresa následující buňky paměti, s kterou se bude pracovat. Přesně řečeno, procesor po přečtení obsahu paměti adresované registrem PC zvýší svůj obsah o jedničku tj. inkrementuje svůj stav, a tak má předem připravenu adresu pro následující buňku, kde leží další slabika instrukce nebo první slabika nové instrukce. Uložení instrukcí v paměti je na obr. 4.



Obr. 4 Uložení instrukcí v paměti

Vzestupná posloupnost provádění instrukcí programu je narušena v případě skokových instrukcí, které ve své operandové části definují adresu paměti, kde bude program tvrdě pokračovat. Např.tříslabiková instrukce JMP 2040H (C3 40 20) způsobí odskok na adresu 2040H paměti. Instrukce je umístěna na adrese 1 000 + 1 002H. Funkce skokové instrukce JMP 2040H je zachycena na obr. 5.

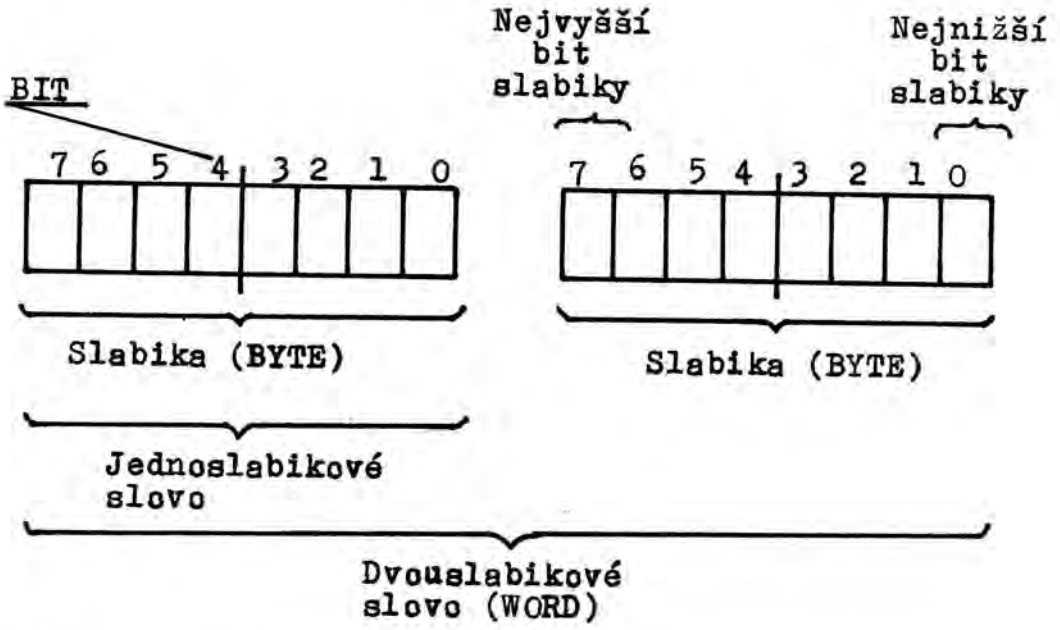


Obr. 5 Provedení skokové instrukce JMP 2040H

Skokové instrukce umožňují odskok na jakékoliv místo paměťové matice a jejich využití se nabízí také při volání podprogramů nebo podprogramů přerušení, při kterých se aplikují speciální instrukce CALL a RST.

Poznámka

Data jsou mikropočítačem zpracována po slovech. Délka slova u mikroprocesorů se nejčastěji pohybuje na 8 či 16 bitech. Mikroprocesor Z80-CPU má délku slova 8 bitů, čemuž odpovídá šířka datové sběrnice i velikost vnitřních paměťových prvků. Osmibitová slova se označují jako slabika (BYTE). Jednotlivé informace v rámci slabiky se nazývají bity (viz obr. 6).



Obr. 6 Vzájemná vazba pojmů slovo, slabika, bit

Instrukce mikroprocesoru Z80-CPU jsou jedno, dvou, tří a čtyř slabikové. Protože DATA BUS je pouze osmibitový, mohou se do vnitřních registrů mikroprocesoru nahrát postupně tj. po slabikách (BYTECH). Instrukce je tedy do mikroprocesoru postupně nahrána po slabikách a vzápětí provedena. První slabika tzv. operační kód obsahuje informaci kolikaslabiková je instrukce.

Příklad:

Instrukční cyklus jednoslabikové instrukce má průběh:

1. Čtení operačního kódu instrukce (FETCH CYCLE)
2. Provedení instrukce (EXECUTE CYCLE)

Příklad:

Instrukční cyklus tříslabikové instrukce má průběh:

1. Čtení operačního kódu instrukce tj. čtení první slabiky (FETCH CYCLE)
2. Čtení operandové části instrukce tj. čtení druhé slabiky (MEMORY READ CYCLE)
3. Čtení operandové části instrukce tj. čtení třetí slabiky (MEMORY READ CYCLE)
4. Provedení instrukce (EXECUTE CYCLE)

Zároveň operační kód definuje způsob provedení instrukce a operandy, s kterými se bude pracovat při vykonání instrukce. Za operačním kódem v případě dvou a tříslabikových instrukcí následuje operandová část, kde je obsažena adresa (periferie, paměti) nebo data (osmibitová/šestnáctibitová), s kterými instrukce bude disponovat při svém vykonání.

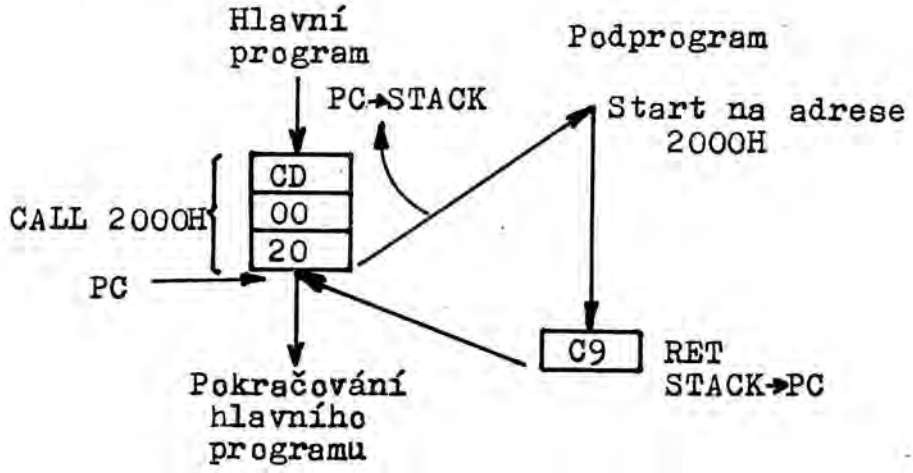
Mikroprocesor Z80-CPU má operační kód jedno až dvoubytový, poněvadž operuje se 693 instrukcemi, tím se rozšiřuje celková délka instrukcí na jedno až čtyřslabikové.

1.4 Zásobníková paměť a podprogramy

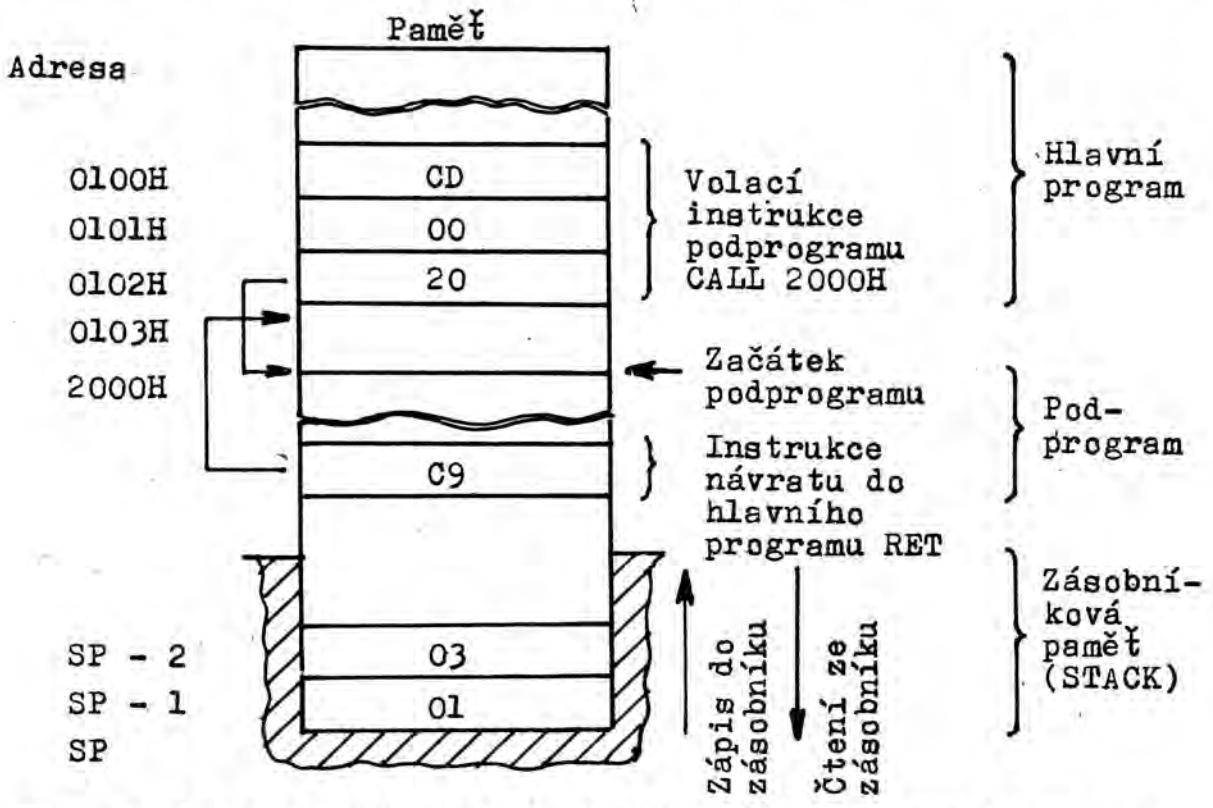
Podprogramem se nazývá část programu, která se opakovaně provádí během hlavního programu. Aby se podprogram nemusel vícenásobně uvádět v paměti, ukládá se v ní jen jednou a v případě potřeby je vždy ve vhodný moment aktivován vyvolávací instrukcí typu CALL nebo RST. Příkladem podprogramu může být operace násobení, dělení, sinus, obsluha periferní jednotky. Jiným příkladem podprogramu je podprogram přerušení, který je vyvolán vnější podmínkou tzv. požadavkem přerušení. Volání a návrat přerušovacího podprogramu probíhá stejným způsobem jako u klasického podprogramu.

Instrukce volání podprogramu CALL je podobná instrukci nepodmíněného skoku JMP. Obsahuje rovněž adresu, která nahradí obsah čítače instrukcí. Mikroprocesor pak pokračuje od této adresy v provádění podprogramu. Na rozdíl od instrukce JMP se však po provedení podprogramu mikroprocesor vrací na místo v programu, odkud byl podprogram volán resp. na místo následující. Z toho vyplývá, že dříve, než se nahradí obsah čítače instrukcí počáteční adresou podprogramu, musí se adresa z čítače instrukcí dočasně uložit na vyhrazené místo paměti, které se označuje jako zásobníková paměť (STACK). Po provedení podprogramu stačí, aby instrukce návratu z podprogramu RET (RETURN), umístěná v podprogramu jako poslední, přemístila zpět do čítače instrukcí dříve uloženou návratovou adresu ze zásobníku. Princip volání

podprogramu a návratu z podprogramu je na obr. 7 a 8. Podprogramy lze vkládat také do sebe tzv. vhnízdění podprogramů (NESTED MODE), což je způsob, kdy podprogram volá jiný podprogram viz obr. 9. Tímto způsobem lze ovšem pracovat jen,má-li procesor k dispozici dostatečnou kapacitu zásobníku pro ukládání adres návratu. Maximální hloubka vložení podprogramů je určena hloubkou zásobníkové paměti.



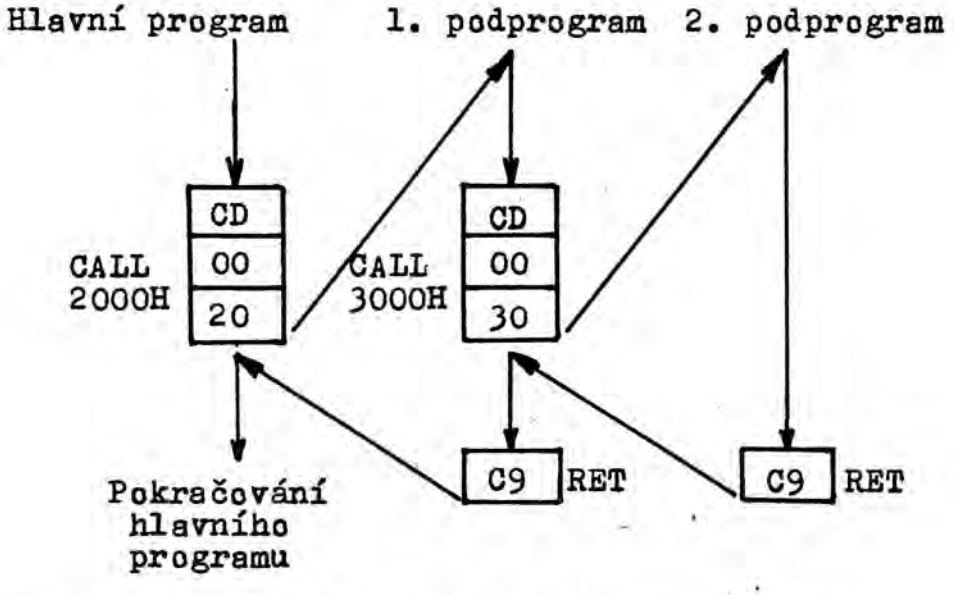
Obr. 7 Princip volání podprogramu a návratu z podprogramu



Obr. 8 Princip práce zásobníkové paměti při volání podprogramu

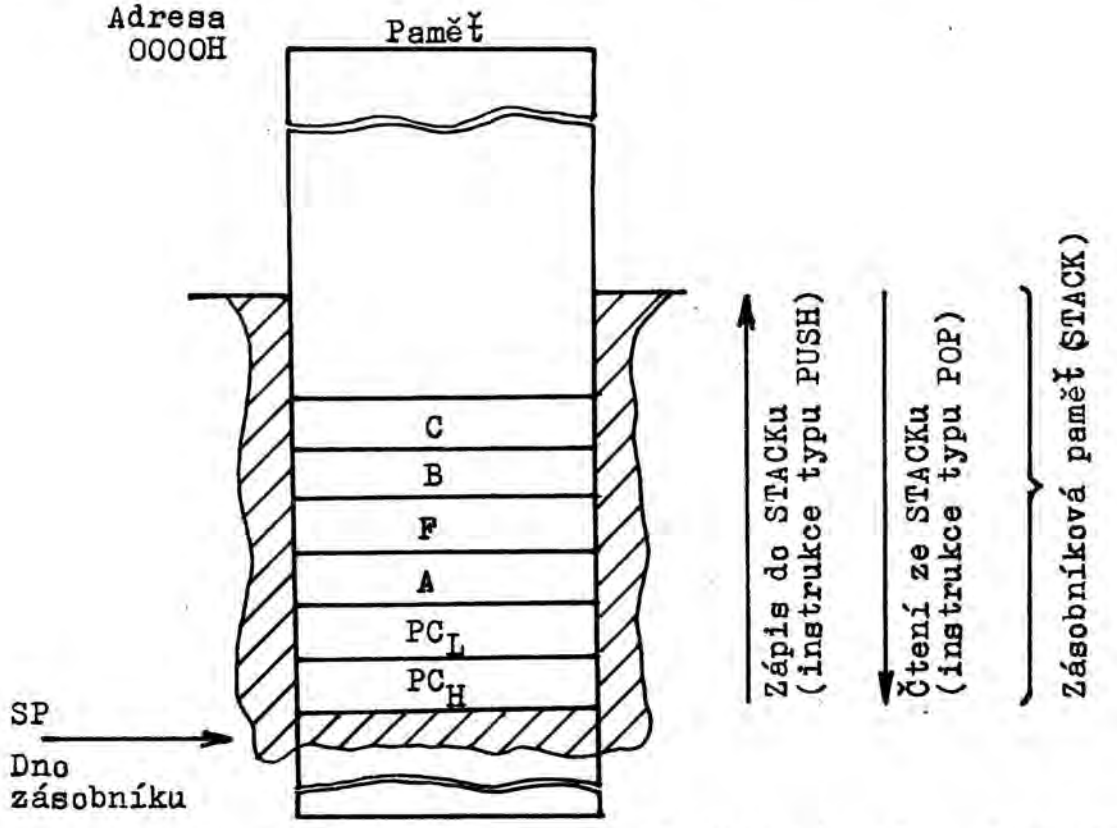
Mikroprocesory první generace měly zásobník přímo uvnitř procesního modulu, proto se označoval jako vnitřní. Byl kapacitně omezen a většinou neumožňoval víceúrovňové obsluhy podprogramů. Procesory druhé generace a vyšší používají jako zásobník vyhrazenou část hlavní paměti a pro její adresování jsou vybaveny speciálním adresovým registrem SP (STACK POINTER) tj. ukazatelem zásobníkové paměti. STACK POINTER se při ukládání adresy návratu automaticky dekrementuje, při zpětném vyčítání návratové adresy se SP automaticky inkrementuje. Zásobník se chová jako paměť typu LIFO (LAST IN-FIRST OUT) tj. poslední dovnitř/první ven a při situování v hlavní paměti umožňuje prakticky neomezené vzájemné vkládání podprogramů a řešení přerušovacích mnohoúrovňových systémů.

Mimo automatického ukládání adres návratu mohou procesory do zásobníku instrukcemi typu PUSH ukládat i obsah akumulátoru, indikátorů (FLAGS) a pracovních registrů typu BC, DE, HL atd. Ke zpětnému čtení těchto informací v přesně obráceném pořadí slouží instrukce typu POP. Instrukce PUSH jsou zápisové, instrukce typu POP čtecí, v obou případech se ale jedná při vykonání instrukce o přesun dvou slabik z mikroprocesoru resp. do mikroprocesoru. Vlastní instrukce typu PUSH a POP jsou jednoslabikové. Princip práce zásobníkové paměti je na obr. 10.



Obr. 9 Vhnízdění podprogramů (NESTED MODE)

První dvě buňky zásobníku jsou obsazeny vyššími a nižšími řády návratové adresy do hlavního programu PC_H a PC_L. Zbylé buňky obsadí instrukce PUSH PSW (ukládá A, F) a PUSH B (ukládá B, C) atd. Dno zásobníku je definováno inicializačním programem a je o jednu adresu výše, než bude první uložená položka v zásobníku.



Obr. 10 Princip práce vnější zásobníkové paměti (STACK)

1.5 Časování mikroprocesoru

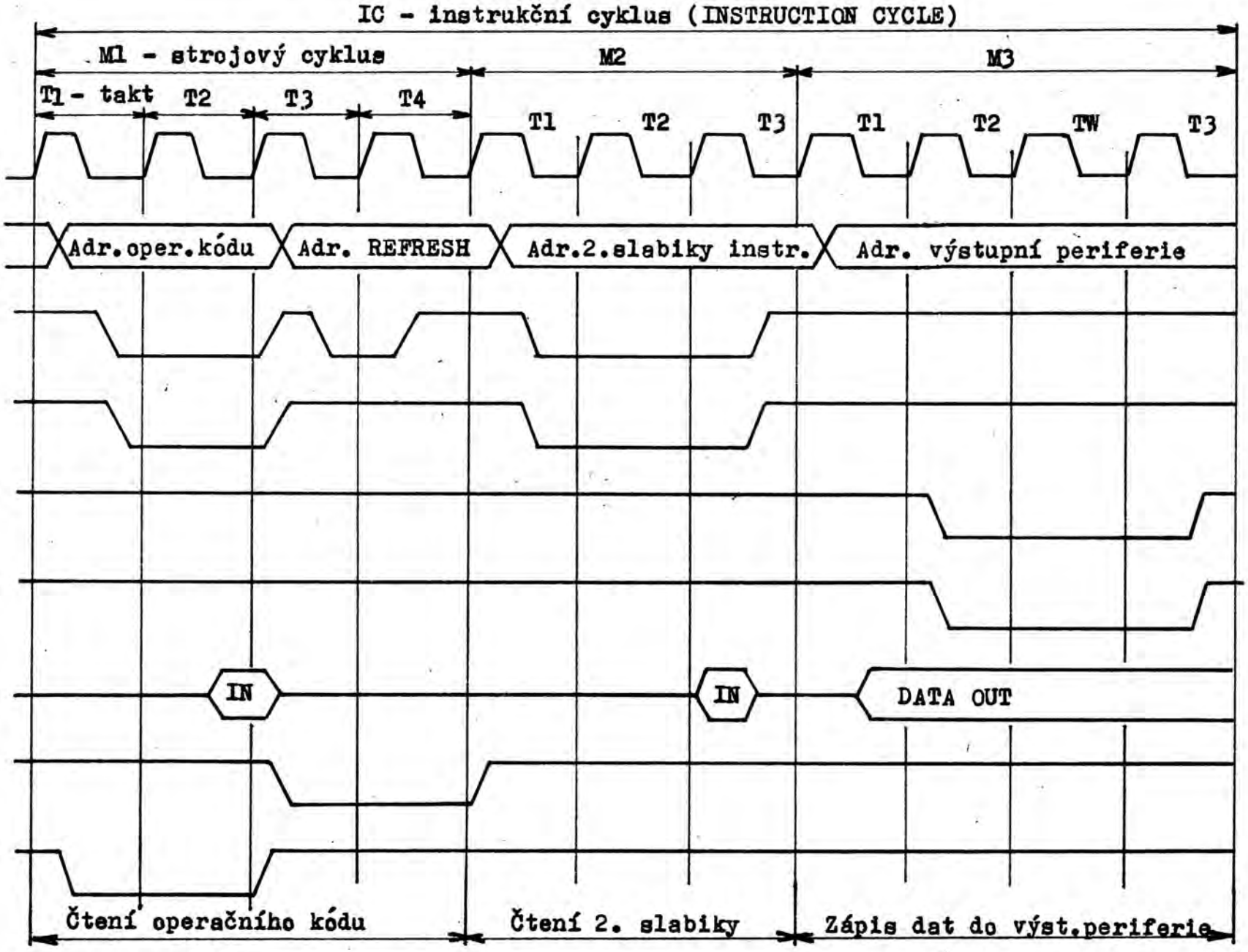
Mikroprocesory pracují synchronně tzn., že jsou přísně časovány veškeré činnosti. Procesní element potřebuje jako zdroj hodinových impulsů oscilátor, který dodává časové signály pro všechny operace mikroprocesoru, ale časuje práci i podpůrných obvodů systému. Výběr a provedení jedné instrukce se označuje jako instrukční cyklus IC (INSTRUCTION CYCLE). Instrukční cyklus sestává ze strojových cyklů M (MACHINE CYCLE), které na úrovni datové sběrnice představují vždy přenos slabiky z nebo do mikroprocesoru. Časový interval mezi dvěma po sobě následujícími

hodinovými impulsy se nazývá periodou hodin (CLOCK PERIOD), neboli taktem T (TIME PERIOD). Platí, že jeden instrukční cyklus obsahuje obvykle více strojových cyklů, které se rozpadají na jednotlivé takty. Vazba mezi instrukčním cyklem IC, strojovým cyklem M a taktem T je na obr. 11. Časový rozbor zachycuje načtení a provedení dvouslabikové instrukce OUT (n),A. Po načtení operačního kódu do instrukčního registru a adresy periferie do dalšího programátorsky nepřístupného registru se provede přenos dat z mikroprocesoru na lokalizovanou výstupní periferii. Adresa periferie byla obsažena ve druhé slabice instrukce. Řídicí funkce synchronizují signály $\overline{\text{MREQ}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{IORQ}}$, $\overline{\text{RFSH}}$, $\overline{\text{M1}}$.

1.6 Synchronizace mikroprocesoru s vnějšími bloky

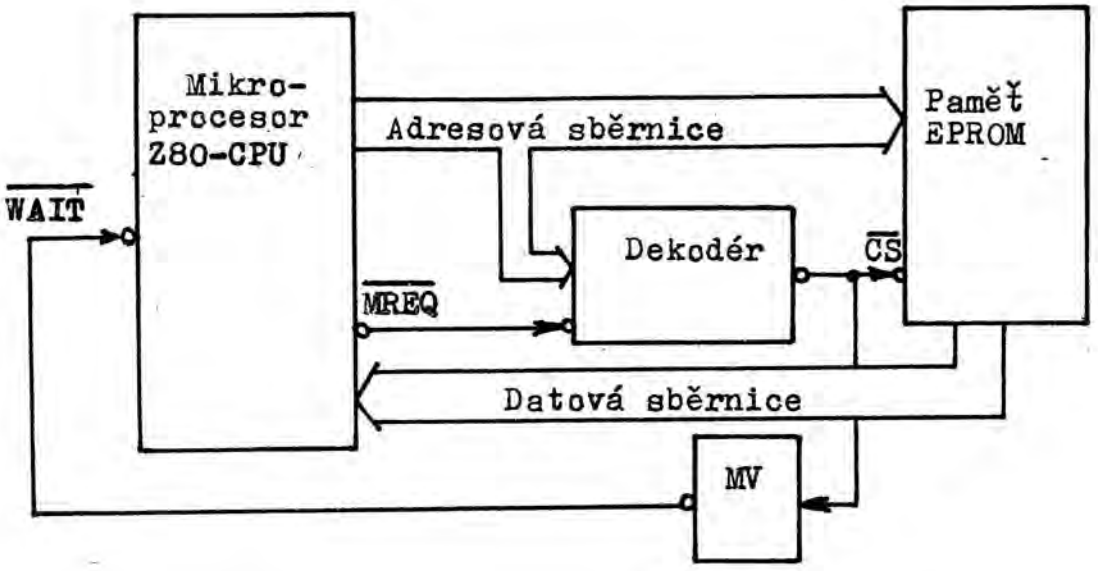
Rytmus mikroprocesoru řídí frekvence hodinových kmitů, z které se odvozují takty T. Ne všechny polovodičové paměti a periferní jednotky jsou dostatečně rychlé, aby tomuto rytmu stačily. Jejich omezení vyplývají z použité technologie (paměti) nebo mechanické konstrukce (periferie). Charakteristickým parametrem pro součinnost s pamětí je její doba přístupu k informaci (ACCESS TIME), která je podrobně vysvětlena v kapitole 1.11.2.

Když mikroprocesor odešle do paměti adresu a řídicí signály, nemůže v činnosti dál pokračovat, dokud paměť nedodá na své výstupy vyčtená data. Některé paměti mají příliš dlouhou přístupovou dobu k informaci, a proto jediným způsobem sesynchronizování je vložit do rytmu mikroprocesoru tzv. čekací takty TW (TIME WAIT), při kterých procesor nepokračuje v činnosti. Procesor disponuje pro tento účel např. vstupem $\overline{\text{WAIT}}$. Jestliže pomalejší paměť dostane výběrový signál $\overline{\text{CS}}$, vyšle se zároveň na vstup $\overline{\text{WAIT}}$ žádost o čekání. Mikroprocesor vyhodnotí stav vstupu $\overline{\text{WAIT}}$ a přejde do stavu čekání WAIT, ve kterém setrvá po celou dobu požadavku na vstupu $\overline{\text{WAIT}}$. Stav čekání trvá vždy celistvý počet period hodin (taktů).



Obr. 11 Časová vazba mezi IC, M a T

Příklad aplikace paměti EPROM s dlouhou přístupovou dobou je na obr. 12. Požadavek čekání je generován přes monostabilní klopný obvod, který je spouštěn při výběru paměti signálem $\overline{CS}$ (CHIP SELECT).



Obr. 12 Synchronizace paměti EPROM s mikroprocesorem Z80-CPU

1.7 Komunikace se vstupy/výstupy

Přenosy mezi vstupy/výstupy mohou probíhat sériovým nebo paralelním způsobem v závislosti na možnostech mikroprocesoru. Např. I 8085A má řešenu komunikaci sériově i paralelně, kdežto systém I 8080A a Z80-CPU jsou typickými osmibitovými paralelními mikroprocesory.

Paralelní přenosy mezi řídicím prvkem a vstupy/výstupy jsou blízké operacím čtení/zápis paměti. Po generování adresy periferie (PORTU) jsou vyslány řídicí signály určující operaci čtení vstupu nebo zápis výstupu. Poté se uskuteční přenos dat po sběrnici. Typickými podpůrnými komunikačními obvody jsou prvky Z80-PIO, Z80-SIO, Z80-DART firmy Zilog.
Paralelní přenos probíhá vždy mezi mikroprocesorem a těmito prvky. Obvody slouží pro připojení periferií s paralelním nebo sériovým tokem dat. Komunikační obvod zároveň zabezpečuje synchronizaci přenosu. Většinou je celá problematika opřená o přerušovací systém. Obvody jsou nejčastěji vybaveny třístavovou

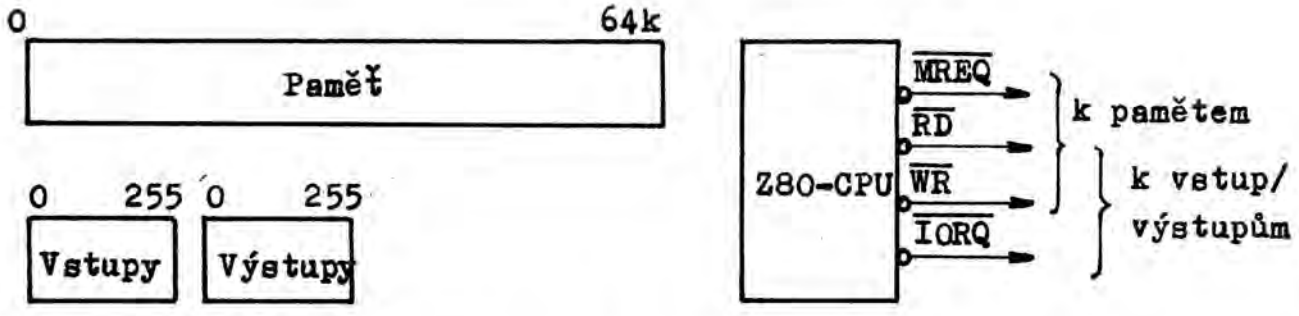
obousměrnou datovou sběrnicí, přes kterou jde datový tok,ale také programovací slova nutná pro volbu režimu práce a stavové údaje, informující mikroprocesor o současném stavu komunikačního prvku a za ním připojené periferie. Třístavovost sběrnice umožňuje multiplexní práci na systémové sběrnici.

Programovou spolupráci zabezpečují se vstupy/výstupy buď specializované instrukce vstupu/výstupu typu IN a OUT, anebo lze použít typické instrukce paměťové, které mají širší adresní možnosti.

1.7.1 Adresování vstupů/výstupů

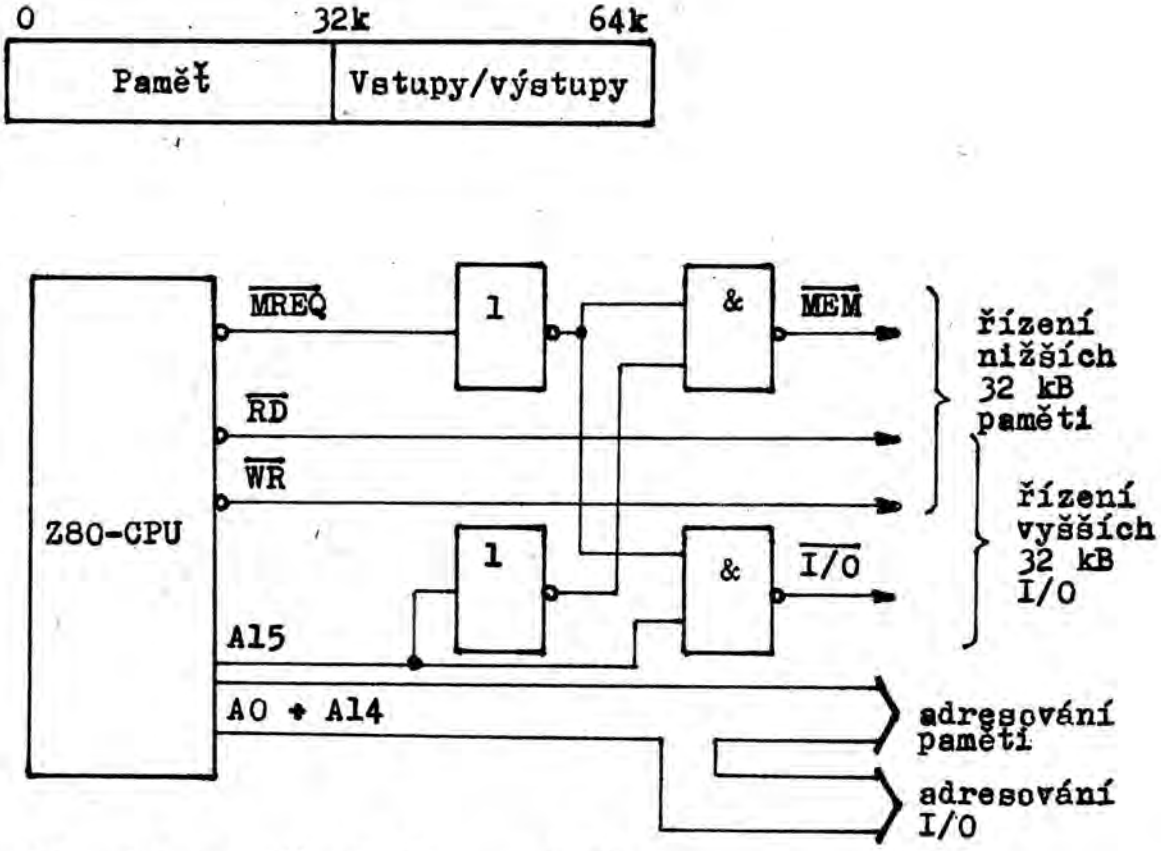
Adresování a komunikaci s periferními jednotkami např. se snímačem a děrovačem děrné pásky, tiskárnou, disketovou jednotkou, klávesnicí,displejem je možné vést dvěma způsoby:

a) izolované I/O (ISOLATED INPUT/OUTPUT) - při tomto způsobu jsou vstupy/výstupy izolovány od paměťové matice a nezabírají žádnou část z množiny adres paměti. Systém Z80 disponuje celkem 24 instrukcemi typu IN a OUT. Procesor při provádění instrukcí aktivuje řídicí signál $\overline{IORQ}$ spolu s $\overline{RD}$ (v případě instrukcí typu IN), anebo $\overline{WR}$ (v případě instrukcí typu OUT). Řídicí signál $\overline{MREQ}$ se při vlastní spolupráci s periferním obvodem nevyužívá. Vstupy/výstupy zaujímají rozsah 256 vstupních a 256 výstupních adresovatelných míst ležících mimo rozsah paměťové matice. Podrobný rozbor instrukcí typu IN a OUT mikroprocesoru Z80-CPU je v kapitole 3.4.8. Rozdělení adresní matice při izolovaných I/O je na obr. 13 spolu s částí řídicí sběrnice Z80-CPU.



Obr. 13 Rozdělení adresní matice při izolovaných I/O v systému s mikroprocesorem Z80-CPU

b) paměťově situované I/O (MEMORY MAPPED I/O) - v tomto případě jsou vstupy/výstupy začleněny do množiny paměťových míst. Vstupy a výstupy sice zabírají část paměťové matice, ale pro práci s nimi lze využít řadu výkonných paměťových instrukcí. Paměťové instrukce používají při svém provedení řídicího signálu $\overline{MREQ}$ ve spolupráci se signály $\overline{RD}$ a $\overline{WR}$, a tím umožňují adresovat libovolně široké pole I/O na úkor paměťového prostoru. Řídicí signál $\overline{IORQ}$ není vůbec aktivován. Obr. 14 je příkladem rozdělení 64 kadresní matice na dvě poloviny, kde nižších 32 kadres lokalizuje paměť a vyšších 32 kadres lokalizuje vstupy/výstupy. Rozlišení jedné a druhé poloviny řídí adresový vodič A15, jehož stav 0 a 1 je využit ke generování separovaných řídicích signálů pro paměť ($\overline{MEM}$) a vstupy/výstupy ($\overline{I/O}$).



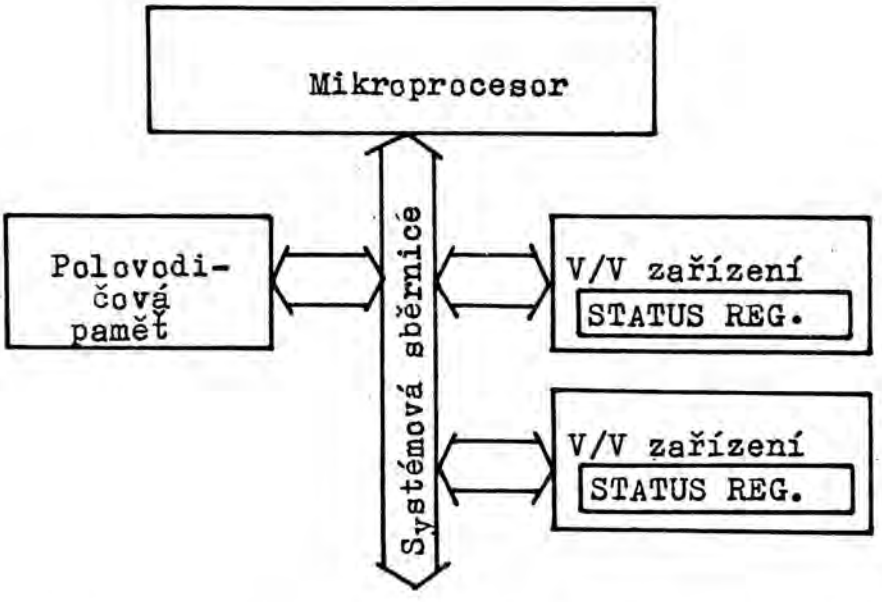
Obr. 14 Paměťově situované I/O

1.8 Přerušení programu

Během zpracování programu mohou v počítači vzniknout požadavky, které mají vysokou důležitost, a proto systém má technické a programové prostředky, pomocí kterých dojde k pozastavení práce na hlavním programu a bude následovat odskok do tzv. podprogramu přerušení. Po skončení obsluhy přerušovacího požadavku je proveden návrat do hlavního programu na místo, kde byl opuštěn resp. na adresu o jedničku vyšší. Principu přerušení se využívá pro obsluhu vstupních a výstupních zařízení, vnějších podmínek, havarijních příčin atd..

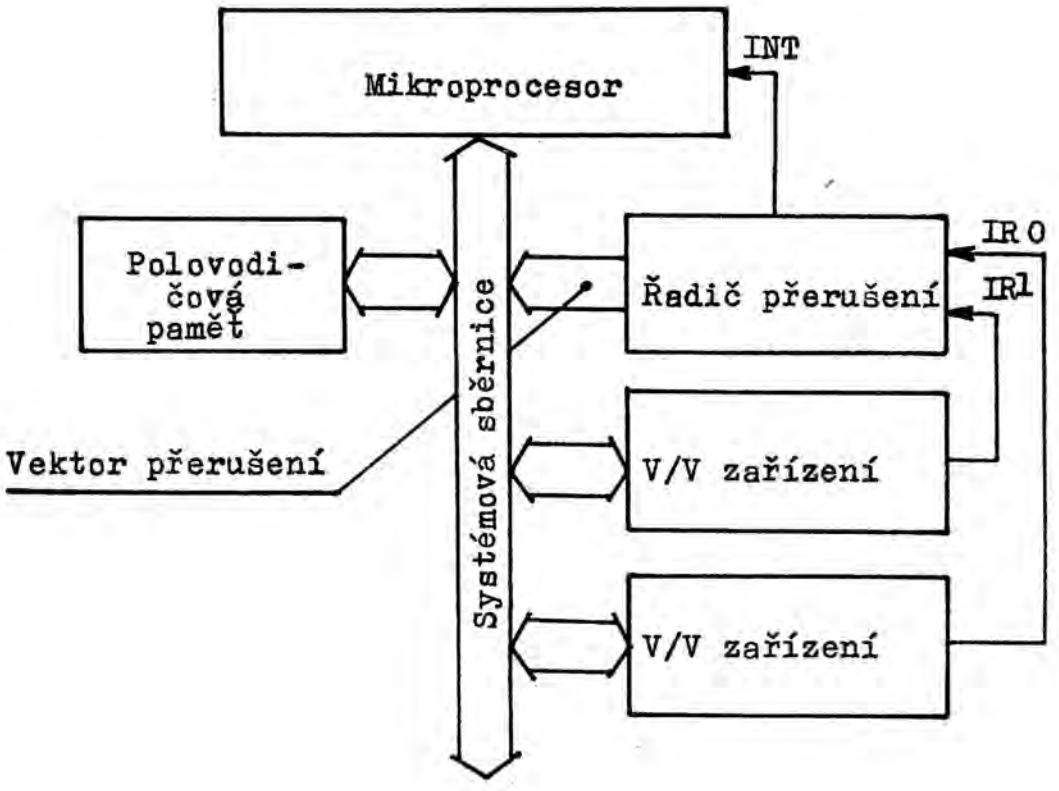
Především při ošetření vstup/výstupních zařízení je možné postupovat několika cestami:

a) dotazovací metoda (POLLING METHOD) - mikroprocesor se postupně dotazuje jednotlivých zařízení, zda žádají o obsluhu. Žádosti o obsluhu jsou pamatovány v registru uvnitř čipu (STATUS REGISTER) nebo vně obvodu na adresovatelném registru. Dotazovací metoda je časově náročná, protože většina dotazů je zbytečná, a tím se značně zpomaluje průchodnost mikroprocesoru. Obr. 15 ukazuje POLLING METHOD s vnitřními registry pro pamatování žádosti o přerušení.



Obr. 15 Dotazovací metoda (POLLING METHOD)

b) přerušovací metoda (INTERRUPT METHOD) - je nejpružnější způsob odbavení požadavku přerušení. Mikroprocesor přechází na obsluhu zařízení teprve tehdy, je-li periferním zařízením žádán. Žádosti o přerušení IR (INTERRUPT REQUEST) se prioritně seřazují v řadiči požadavků přerušení, který svým výstupem INT (INTERRUPT) žádá mikroprocesor o přerušení hlavního programu. Jako řadič přerušení může být aplikován obvod I 8214 nebo I 8259, a to jednotlivě nebo kaskádně. Tím vzniká osmiúrovňový nebo více jak osmiúrovňový přerušovací systém. Řadič přerušení mimo požadavku INT dodává na systémovou sběrnici také vektor přerušení, obsahující zakódovanou startovací adresu podprogramu přerušení. Každému perifernímu zařízení je přiřazen konkrétní podprogram přerušení (INTERRUPT SERVICE ROUTINE). Na obr. 16 je obecný příklad přerušovací metody.

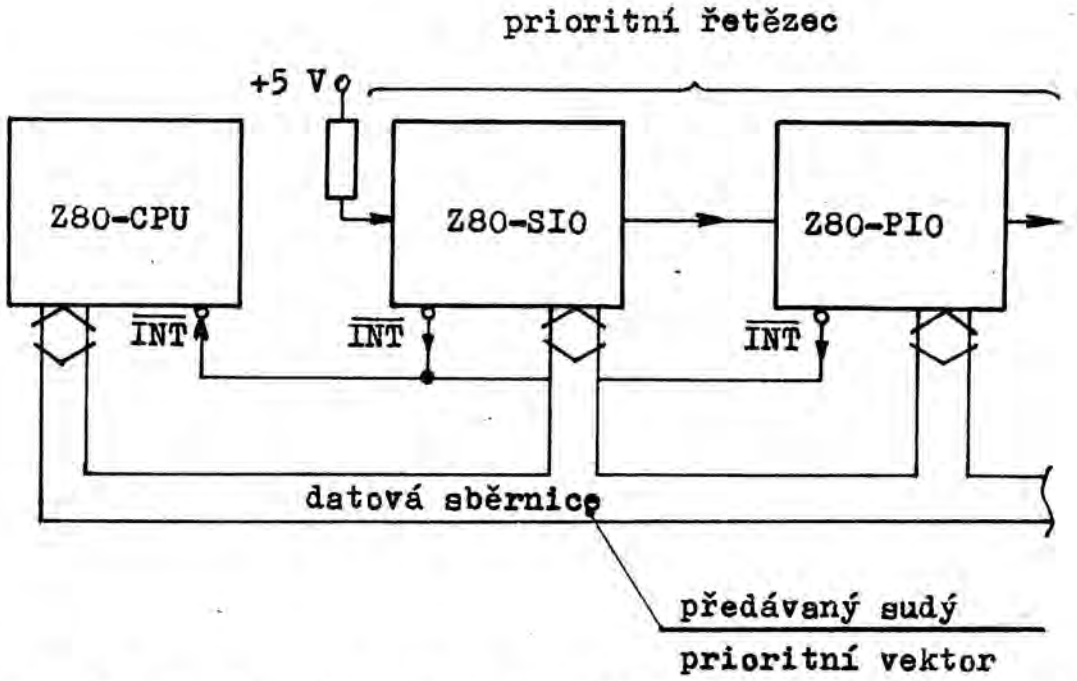


Obr. 16 Přerušovací metoda (INTERRUPT METHOD)

c) specifická metoda pro Z80-CPU pracující v MODE 2 - touto metodou lze odbavit až 128 požadavků přerušení generovaných programovatelnými podpůrnými obvody firmy Zilog typu Z80-PIO, Z80-SIO, Z80-CTC, Z80-DMA, Z80-DART. Mikroprocesor musí být programově nastaven do přerušovacího režimu 2 (MODE 2) instrukcí IM2. Podpůrný obvod, jehož požadavek přerušení byl potvrzen, dodává na datovou sběrnici osmibitový, vždy sudý, periferní vektor, který na úrovni mikroprocesoru spolu s registrem I vytváří šestnáctibitovou sudou adresu tabulky paměti. Teprve na sudé a následné liché adrese této tabulky se nachází šestnáctibitová počáteční adresa podprogramu přerušení.

Metoda nevyžaduje žádné další technické prostředky pro řešení přerušení. Umožňuje vytváření prioritních řetězců podpůrných obvodů a navíc koncipování začátků obslužných podprogramů může být na jakékoliv adrese paměti vzhledem k nepřímé formě volání podprogramu přerušení.

Vytváření prioritního řetězce s vazbou na mikroprocesor Z80-CPU zachycuje obr. 17.



Obr. 17 Specifická metoda pro Z80-CPU pracující v MODE 2

Jiný pohled na odbavení přerušení je z hlediska masky přerušení na úrovni mikroprocesoru:

a) maskovatelné požadavky přerušení - každý mikroprocesor má jeden nebo několik maskovatelných požadavků přerušení.

Např. mikroprocesor Z80-CPU disponuje klopnými obvody IFF - povolení přerušení (INTERRUPT ENABLE), jejichž stav je rozhodující pro možnost akceptování přerušení ze vstupu $\overline{INT}$. Žádost o přerušení je testována na konci provedení každé instrukce.

Mikroprocesor I 8085A má čtyři přerušovací maskovatelné vstupy INTR, RST7.5, RST6.5, RST5.5, které mají možnost centrální masky a u požadavků RST7.5, RST6.5, RST5.5 i separované masky.

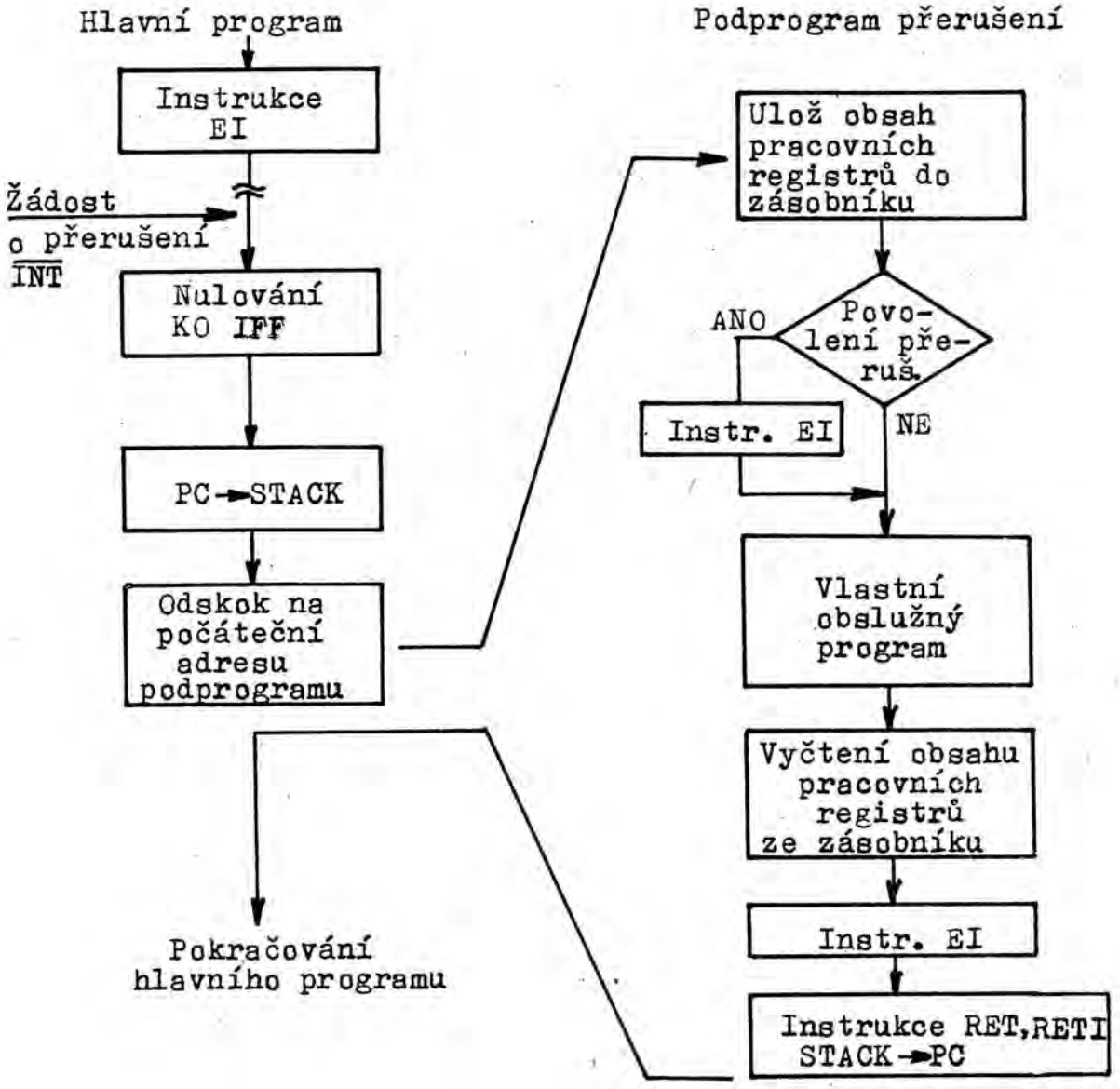
Programově se klopné obvody IFF ovládají instrukcemi EI (ENABLE INTERRUPT) a DI (DISABLE INTERRUPT). Z technických prostředků klopné obvody IFF nuluje vstup $\overline{RESET}$ a vlastní žádost o přerušení. Posloupnost operací po žádosti o maskovatelné přerušení je na obr. 18.

b) nemaskovatelné požadavky přerušení - jsou využity pro ošetření výpadku napájení nebo dalších havarijních stavů. Nemaskovatelný požadavek přerušení je ošetřen vždy, tj. nezávisle na stavu vnitřní logiky mikroprocesoru. Testování žádosti o přerušení spadá opět na konec provedení instrukce.

I 8080A nemá nemaskovatelný požadavek přerušení, I 8085A má k dispozici vstup TRAP a Z80-CPU vstup $\overline{NMI}$. Nemaskovatelné požadavky přerušení vyvolávají start podprogramu přerušení na konkrétní adrese paměti. Požadavek TRAP na adrese 24H a $\overline{NMI}$ na adrese 66H.

1.9 Konstrukce výstupů LSI obvodů

Bloky mikropočítačů jsou paralelně propojeny společnou obousměrnou datovou sběrnicí, přes kterou probíhá přenos informací oběma směry. Na datovou sběrnici se postupně připojují jednotlivé zdroje informací. Ostatní zdrojová místa musí být v daném

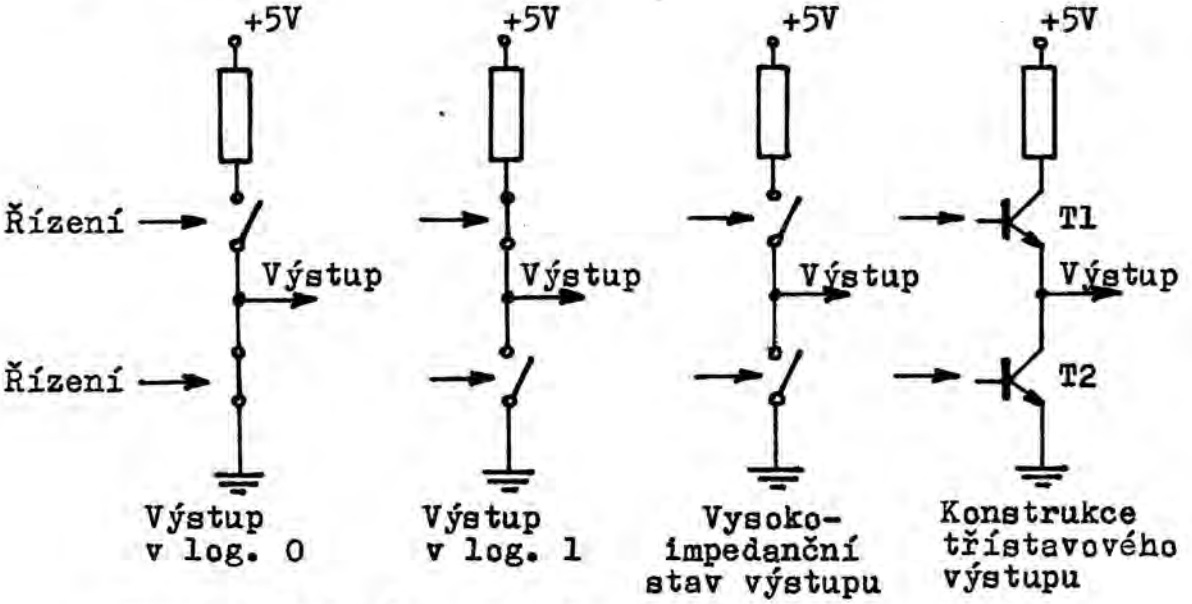


Obr. 18 Vývojový diagram odbavení žádosti o maskovatelné přerušení u I 8080A

momentě uskutečněného přenosu neaktivní. Výběr zdrojového místa provádí mikroprocesor (případně DMA řadič) za pomoci adresové a řídicí sběrnice. Zdrojová místa mají své obvodové struktury vytvořeny z budičů s třístavovým výstupem nebo z budičů s otevřeným kolektorem. Konstrukce třístavových výstupů a výstupů na otevřených kolektorech se netýká jen datové sběrnice (DATA BUS), ale stejně jsou realizovány výstupy polovodičových pamětí, umožňující vytváření širokých paměťových matic paralelním propojením

více čipů. Jiným příkladem je třístavová konstrukce adresové a řídicí sběrnice mikroprocesoru, která dává možnost sdílení těchto sběrnic i jiným řídicím prvkem např. řadičem DMA.

Principiální a realizační schéma třístavového budiče je na obr. 19.

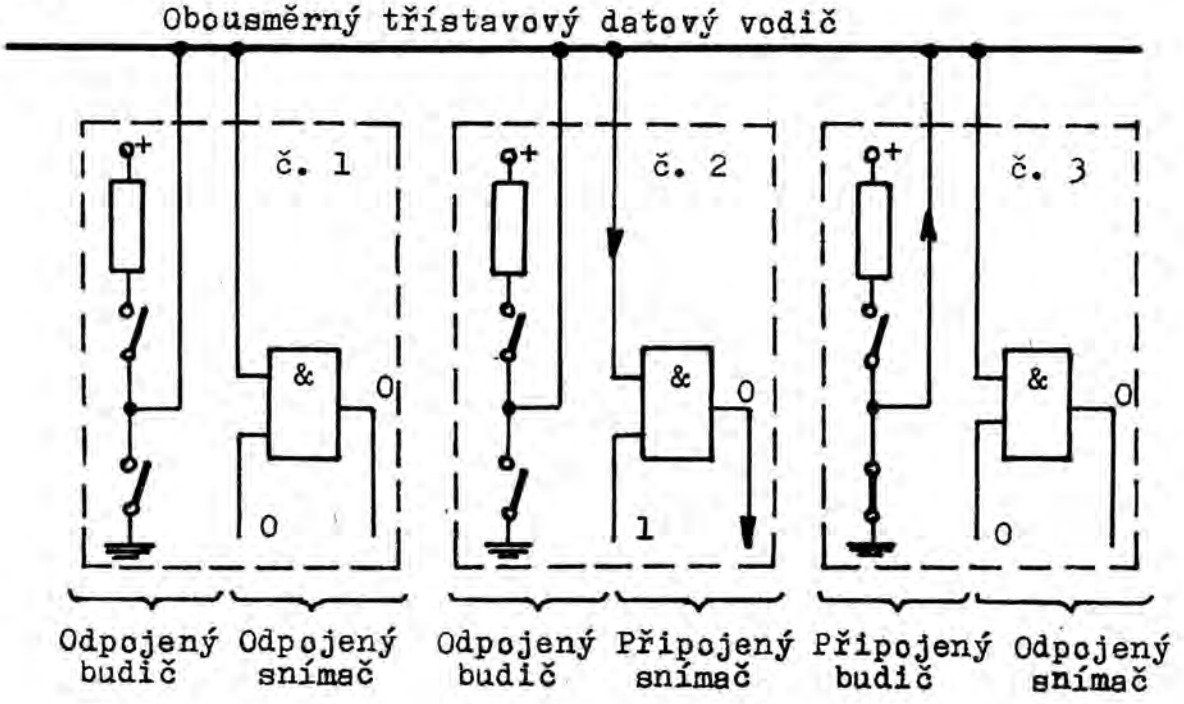


Obr. 19 Princip a realizace třístavového budiče

Koncový stupeň třístavového budiče je realizován ze dvou tranzistorů T1, T2, které lze ovládat do třech rozdílných stavů. Stavu T1 a T2 zavřen, odpovídá stav odpojenosti výstupu od zdroje a země. Tento stav se označuje jako vysokoimpedanční neboli třetí stav. Případ sepnutého tranzistoru T1 a zavřeného tranzistoru T2 odpovídá výstupu logické jedničky. Poslední případ sepnutého tranzistoru T2 a zavřeného tranzistoru T1 nastaví výstup do logické nuly. Schematické značky třístavových obvodů jsou na obr. 20.

Obr. 20 Schematická značka třístavových budičů

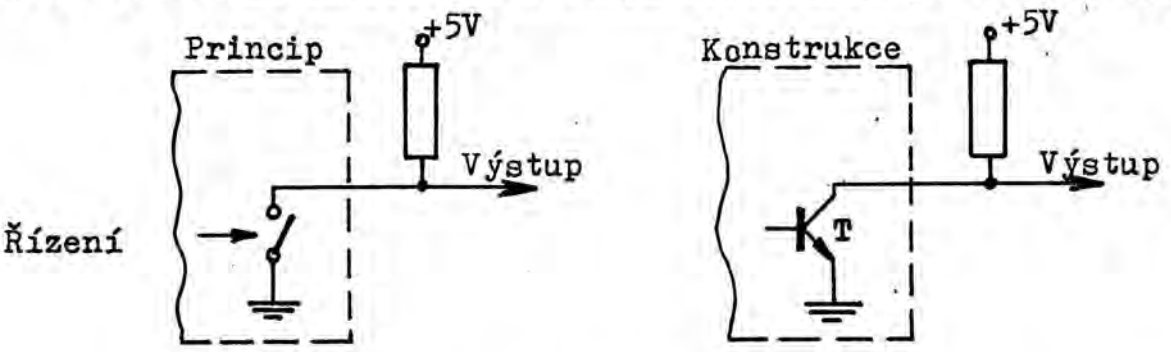
Obr. 21 je příkladem propojení tří uzlů datové obousměrné sběrnice. V zapojení je blok č. 3 zdrojem informace (log. 0) a tato informace je akceptována v bloku č. 2, jehož snímač je aktivován jedničkovým řídicím signálem. Ostatní snímače č. 1 a č. 3 jsou odpojeny. Podobně jsou neaktivní budiče v bloku č. 1 a č. 2.



Obr. 21 Propojení obousměrné třístavové sběrnice

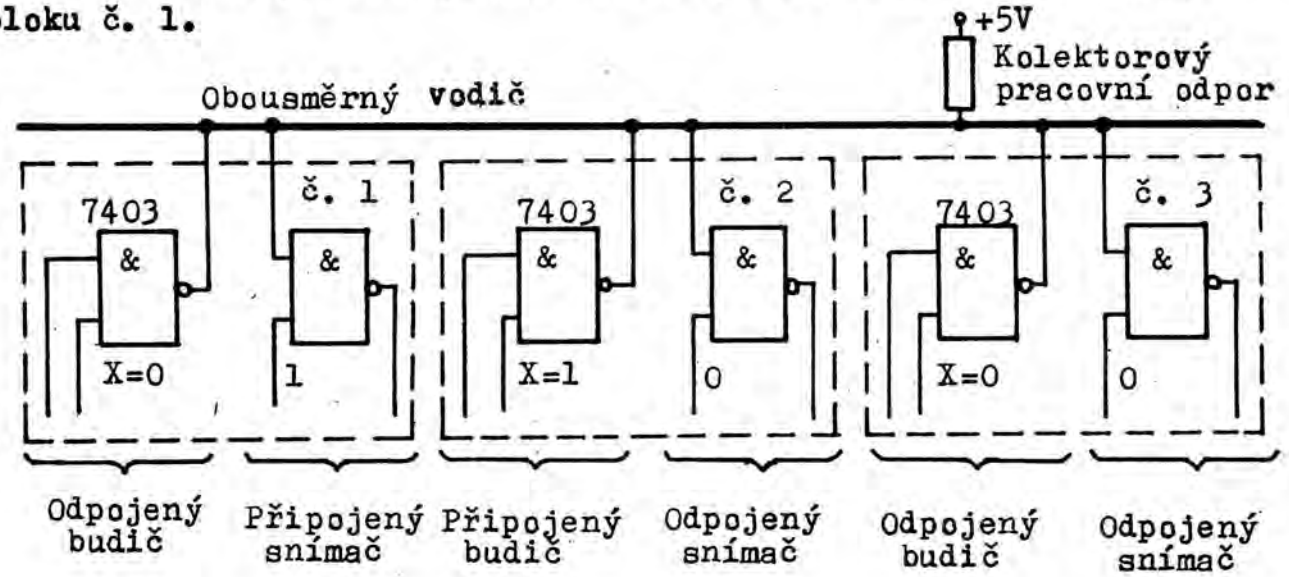
Nevýhodou třístavových výstupů je možnost vzniku značných vyrovnávacích proudů mezi několika výstupy v aktivním stavu vlivem nevhodného časování. Je-li více jak jeden výstup v aktivním režimu a propojené výstupy budou představovat rozdílnou napěťovou hladinu, může dojít ke zničení obvodu vyrovnávacími proudy. Tato problematika u otevřených kolektorů nepřichází v úvahu. Obvody s třístavovými výstupy jsou svými elektrickými parametry vhodnější, ale jsou mnohem choulostivější na časovou přesnost ovládacích signálů.

Otevřený kolektor má pouze jediný tranzistor T, který podle stavu řídicí elektrody definuje stav na kolektoru, a to stav logické nuly nebo logické jedničky. Výstupní napětí se snímá na vnějším přídavném kolektorovém odporu. Principiální a realizační schéma budiče s otevřeným kolektorem je na obr. 22.



Obr. 22 Princip a realizace budiče s otevřeným kolektorem

Nevýhoda otevřených kolektorů je především v použití vnějších kolektorových odporů, s kterými je nutné počítat při návrhu tištěného spoje a rozmístění součástek. Propojení obousměrné sběrnice s budiči na otevřených kolektorech ukazuje obr. 23. Zdroji informací jsou budiče MH 7403, řízené přes vstup X. V uvedeném příkladě probíhá přenos ze zdroje bloku č. 2 do snímače bloku č. 1.



Obr. 23 Propojení obousměrné sběrnice s budiči na otevřených kolektorech

1.10 Přímý přístup do paměti

Přenosové frekvence některých periferních zařízení jsou natolik vysoké, že je nestačí obsluhovat mikroprocesor s použitím systému přerušení, protože časová náročnost obslužného programu se značně prodlužuje vlivem načtení instrukcí z paměti do mikroprocesoru a vlivem nezbytného využití vnitřních registrů procesoru při přenosech dat. Např. se jedná o spolupráci s pevným nebo pružným diskem. Styk s těmito rychlými zařízeními se proto zajišťuje bez účasti mikroprocesoru, ale za pomoci speciálního obvodu DMA (DIRECT MEMORY ACCESS CONTROLLER), který zprostředkuje tzv. přímý přístup informací do paměti od periferie nebo ve směru opačném, tj. přímý přístup informací z paměti do periferie. Formy přímých přenosů u vyspělejších DMA řadičů mohou probíhat i mezi dvěma rychlými periferiemi anebo lze DMA řadičem řídit blokové přenosy mezi dvěma zónami paměti. Veškeré uvedené formy přenosů umí řadič Z80-DMA.

Řadiče DMA jsou programovatelné obvody vysoké složitosti. Programování zajišťuje mikroprocesor před uskutečněním DMA přenosu. Součástí programové sekvence je zadání počáteční adresy zdrojového a cílového místa přenosu, délky přenášené zóny, směru přenosu apod..

DMA přenos může probíhat dvěma metodami:

a) přerušením činnosti mikroprocesoru - požadavek na zahájení DMA přenosu obdrží řadič od periferie nebo programově. Řadič pak žádá mikroprocesor o uvolnění sběrnic, kterými mikroprocesor dosud disponoval. Na základě předání sběrnic zahajuje DMA řadič fyzický přenos mezi zdrojovým a cílovým místem systému. Po ukončení přenosu řadič DMA vrátí zpět sběrnice mikroprocesoru, a ten může pokračovat v pozastaveném provádění programu. Aby bylo možné sdílet adresovou, datovou a řídicí sběrnici více řídicími prvky (mikroprocesor, DMA řadič), je jejich konstrukce třístavová. V případě vedení DMA přenosu je stav na sběrnicích mikroprocesoru vysokoimpedanční tj. stav odpojenosti.

   Tato metoda snižuje propustnost systému při DMA přenosu a zvětšuje odezvu na vnější požadavky přerušení, protože při DMA přenosu nelze přenos přerušit.

b) cyklickým sdílením - tato forma vedení přenosu odstraňuje některé nedostatky metody přerušení činnosti mikroprocesoru. Cyklické sdílení může probíhat dvěma způsoby:

- prokládáním časové činnosti mikroprocesoru a činnosti DMA přenosu např. po každém instrukčním cyklu. Mikroprocesor se odpojuje od systémové sběrnice obvykle jen na přenos jednoho slova. Propustnost systému je sice srovnatelná s předcházející metodou, ale zkrátila se časová odezva na vnější události (požadavky přerušení).
- transparentní způsob. Rychlé DMA přenosy se uskutečňují v části instrukčního cyklu mikroprocesoru, ve kterém mikroprocesor nevyžaduje přístup na systémovou sběrnici, a tím nedochází k blokování práce programu. Vhodným prostorem pro uskutečňování této formy přenosu je část strojového cyklu M1 (FETCH), při kterém je dokódován operační kód instrukce na úrovni mikroprocesoru a systémová sběrnice je nevyužita. Přenosové rychlosti vedení DMA jsou při transparentním způsobu závislé na typu instrukcí.

1.11 Polovodičové paměti

K uchování informací v číslicové technice se využívá pamětí dvojkové informace. Klasické počítače a minipočítače byly osazeny nejčastěji pamětmi feritovými nebo pamětmi na tenkých magnetických vrstvách. Doba cyklu se pohybuje u feritových pamětí kolem 1 ÷ 2 /us. Výhodou feritových pamětí je uchování vnitřní informace i při výpadku napájení. Tuto základní vlastnost ztrácejí polovodičové operační paměti typu RWM, které jsou nyní výhradně aplikovány v mikropočítačových systémech. Udržení informací se řeší zálohováním napájení z akumulátoru. Paměti RWM se pak volí v technologii C-MOS, která se vyznačuje nízkou spotřebou a značnou necitlivostí na pokles napájecího napětí. Polovodičové paměti disponují ale mnohými výhodami např. nízký příkon, malá rozměrovost, vysoká kapacita, možnost dalšího rozšiřování kapacity vzájemným propojením pouzder, krátká přístupová doba, spolehlivost, nízká cena.

1.11.1 Členění polovodičových pamětí

Soustředění rozsáhlých paměťových matic do jediného integrovaného obvodu umožnil rozvoj technologie TTL-S, N-MOS a C-MOS. Nabízí se rozdělit paměti podle použité technologie výroby, vnitřní struktury, způsobu přístupu k informacím, možnosti záznamu a čtení atd..

Základní členění pamětí lze provést do čtyř skupin:

a) dle základní technologie na bipolární a unipolární

b) dle způsobu realizace paměťové buňky na paměti statické a dynamické

c) dle záznamu a čtení na paměti s měnitelným obsahem buněk a neměnitelným obsahem buněk

d) dle přístupu k vnitřní informaci na paměti s libovolným přístupem a sériovým přístupem.

Podrobný rozbor skupin:

a) dle základní technologie na bipolární a unipolární - realizace paměti bipolární technikou TTL-S ovlivňuje mimo jiné parametry hlavně přístupovou dobu k informaci a kapacitu paměti.

Paměti provedené na technikách unipolárních N-MOS, C-MOS pracují s delší přístupovou dobou, ale kapacita je vyšší než u bipolárních struktur.

Ideální technologie by měla skloubit nároky na rychlost s potřebou vysoké hustoty integrace. Bohužel dosud jdou tyto vlastnosti proti sobě. Proto při aplikování např. rychlého bipolárního mikroprocesorového systému MH 3000 z koncernového podniku Tesla Rožnov je možné výhradně použít bipolárních pamětí. Paměti MOS by značně zpomalily rychlostní možnosti řezového mikroprogramovaného systému, který se vyznačuje výraznou univerzálností.

b) dle způsobu realizace paměťové buňky na paměti statické a dynamické - základem statické paměti je paměťová matice se statickými buňkami. Statickou buňku tvoří bistabilní klopný obvod, složený ze čtyř tranzistorů. Nejčastější provedení je v technologii N-MOS a C-MOS. Bipolárně řešené statické paměti mají kratší přístupovou dobu, menší hustotu a větší příkon. Obvod odebírá stálý příkon, ať je v kterémkoliv stavu, takže celá matice odebírá příkon úměrný stavbě vlastního klopného obvodu a především kapacitě paměťové matice.

Základem dynamické paměťové buňky je využití kapacity řídicí elektrody MOS tranzistoru. Hodnota náboje ovšem klesá s časem, a proto musí docházet k pravidelnému zotavování náboje tzv. REFRESH. Perioda zotavovacích cyklů se pohybuje u dynamických pamětí maximálně na hodnotě 2 ms. Zotavovací impulsy adresují vždy více buněk paměťové matice např. řádek či sloupec. Existují dva základní principy zotavení informace u dynamických pamětí, a to celkové zotavení paměti vždy na začátku každých 2 ms, nebo rovnoměrné rozložení zotavovacích cyklů do doby 2 ms. V současnosti je paměťová buňka

jednotranzistorová, což přináší zvětšení paměťové kapacity oproti statickým pamětem.

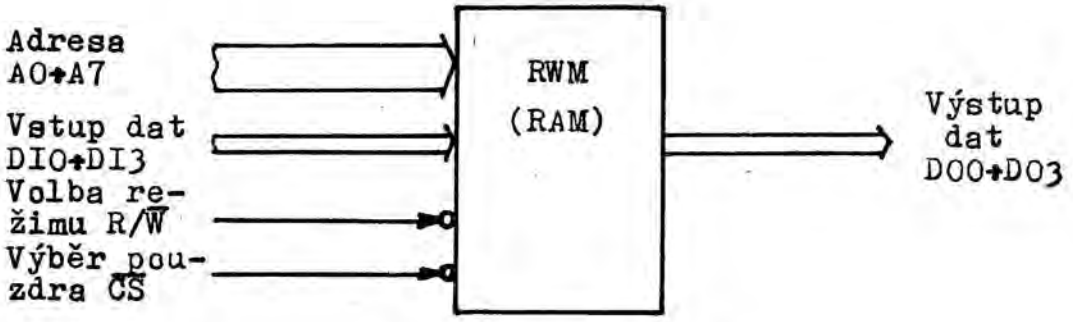
Prakticky se obnova informací provádí v době, kdy řídicí prvek (mikroprocesor) nevyužívá sběrnice. Na tomto principu provádí automaticky obnovu informací mikroprocesor druhé generace Z80-CPU firmy Zilog, který generuje příslušnou adresu a řídicí signál $\overline{\text{RFSH}}$ pro paměť v taktu T3 a T4 ve strojovém cyklu M1 - FETCH, kdy probíhá dekódování instrukce uvnitř mikroprocesoru a sběrnice nejsou využity. I 8080A nemá vnitřní aparát pro REFRESH pamětí, a proto je nutné zajistit obnovu pamětí pomocnými technickými prostředky.

K. p. Tesla Piešťany vyrábí 16 kbitovou dynamickou paměť MHB 4116, která vyžaduje obnovu všech 16 384 buněk každé 2 ms. Obnova se provádí ve 128 cyklech, kde každý cyklus aktivuje sumu 128 bitů.

c) dle záznamu a čtení na paměti s měnitelným obsahem buněk a neměnitelným obsahem buněk - z hlediska způsobu záznamu či čtení paměti lze dělit paměti na RAM (RWM) a ROM. Paměti ROM lze ještě dále dělit na PROM, EPROM a EAROM.

Polovodičové paměti RAM (RANDOM ACCESS MEMORY) tj. paměti s libovolným přístupem k vnitřní informaci jsou paměti, které se nyní používají hlavně jako paměti operační. Slouží k zápisu a čtení dat potřebných pro program. Symboly RAM jsou převzaty z počátku historie výpočetní techniky, kdy byly používány výhradně operační feritové paměti a jiné typy nebyly známy, kromě periferií pro uchování informací. Bubnové, páskové a diskové periferní jednotky měly ale přístup sériový SAM (SERIAL ACCESS MEMORY), a tím se odlišovaly od pamětí typu RAM. V současné době mají všechny polovodičové paměti libovolný přístup k vnitřním údajům, a proto je daleko výstižnější označení pro operační paměti RWM (READ WRITE MEMORY) tj. paměť určená ke čtení a k zápisu informace. V technické literatuře se uvádí symboly RAM i RWM. Paměti RWM jsou v bipolárních technikách realizovány nejčastěji v technice TTL-S, v oblasti unipolárních technik jde o N-MOS a C-MOS paměti.

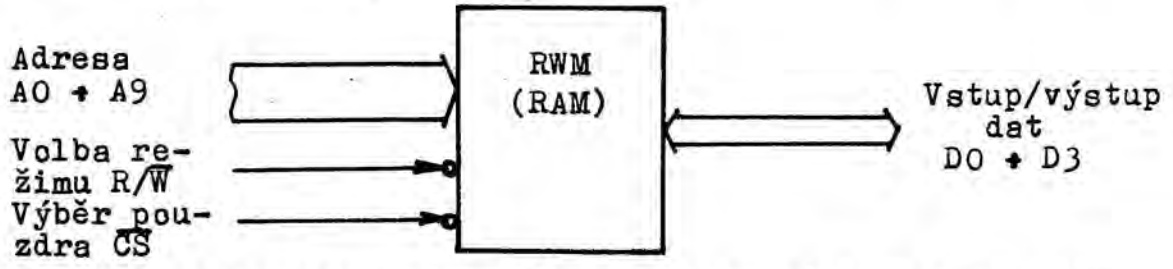
Zapojení vývodů paměti RWM o kapacitě 1 kbitů s organizací 256 x 4 bity je na obr. 24.



Obr. 24 Zapojení paměti RWM s organizací 256 x 4 bity s oddělenými vstupy a výstupy

Paměť má oddělené vstupy a výstupy. Adresování 256 slov zajišťuje osmibitová adresa A0 ÷ A7, výběr pouzdra provádí signál $\overline{CS}$ = 0. Ve stavu $\overline{CS}$ = 1 je funkce paměti blokována a její výstupy jsou v neaktivním stavu (třístavové nebo na otevřených kolektorech). Při vybrané paměti se volí dva základní režimy práce, a to zápisový a čtecí na základě stavu signálu R/$\overline{W}$. Vodič R/$\overline{W}$ = 1 definuje vyčtení vnitřního obsahu na výstupy DO_0 ÷ DO_3 z naadresovaného slova. Vstupní vodič R/$\overline{W}$ = 0 lokalizuje zápis vstupů DI_0 ÷ DI_3 do naadresovaného slova.

Na obr. 25 je příklad paměti RWM o kapacitě 4 kbitů se společnou obousměrnou datovou sběrnicí, která umožňuje přímé připojení na obousměrně řešené datové sběrnice mikroprocesorů. U nás je v tomto provedení vyráběna unipolární paměť MHB 2114 s třístavovými výstupy.

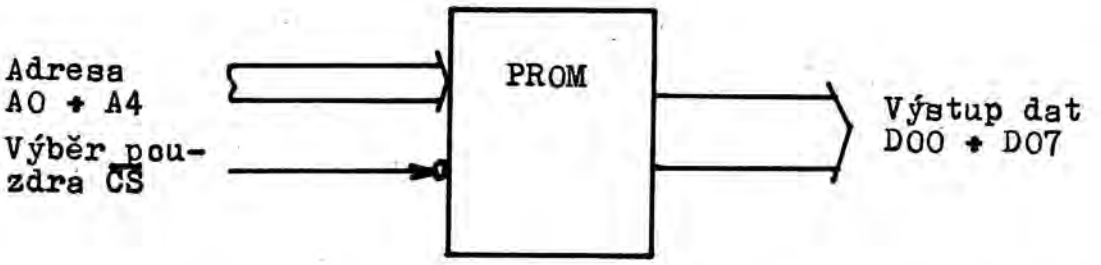


Obr. 25 Zapojení paměti RWM s organizací 1 024 x 4 bity se společnou obousměrnou datovou sběrnicí

Polovodičová paměť ROM (READ ONLY MEMORY), tj. paměť jen pro čtení informací neboli pevná paměť, slouží k uložení mikroprogramů, programů, konstant atd.. Vnitřní obsah je dán ve výrobním závodě programovací maskou, která se použije při konečné fázi výroby. Maska umožňuje u MOS technologie nanesení tenké a silné vrstvy SiO_2. Tranzistor MOS vznikne jen v oblasti tenké vrstvy SiO_2. U bipolárních pamětí ROM se volba 1 a 0 v paměťové matici provádí pouze propojením tranzistorů s řídicími vodiči. Zákaznické paměti ROM jsou drahé, a proto se uživatelům vyplácí jen při velkých sériích. Bipolární paměti ROM jsou velmi rychlé, doba přístupu je desítky ns, MOS paměti pracují s přístupovými dobami stovek ns. Představitelem unipolárních pamětí ROM je paměťová řada MHB 2500 vyráběná v k. p. Tesla Piešťany, v rámci které jsou vyráběny generátory znaků v latince a azbuce, převodníky kódu IS07 na dálnopisný kód atd..

Variantou paměti ROM je paměť PROM (PROGRAMMABLE READ ONLY MEMORY), tj. programovatelná pevná paměť, do které si uživatel sám může zapsat jednou provždy informace, takže odpadají náklady na zhotovení masek ve výrobním závodě a dlouhé dodací lhůty. Bipolární paměti PROM jsou vyrobeny na principu přerušitelných tavných spojek. Spojky jsou chromniklové a potřebují k přepálení proud asi 10 x větší, než je proud pracovní. Jiným materiálem pro spojky může být polykrystalický křemík. V programovací matici v nenaprogramovaném stavu jsou všechny buňky v log. 0, naprogramováním příslušný bit přechází do stavu log. 1. Jiný způsob programování spočívá v prorážení diodového přechodu. Nenaprogramovaná matice představuje prakticky soustavu rozepnutých kontaktů, kdežto matice naprogramovaná prezentuje soustavu sepnutých kontaktů. Dodatečné změny lze provádět jen na dosud nenaprogramovaných bitech, protože přepálenou spojku nebo proražený diodový přechod není možné vrátit do původního stavu. Paměti typu PROM se aplikují do funkčně zcela ověřených zařízení, jejichž programy nebudou po nasazení podléhat změnám. Zapojení vývodů paměti PROM (viz obr. 26) se proti paměti RWM koriguje o datové

vstupní vodiče a režimový signál R/$\overline{W}$, protože paměti typu ROM, PROM, EPROM mohou pracovat jen ve čtecím režimu. Programování u paměti PROM a EPROM je vedeno přes výstupní datové linky.



Obr. 26 Zapojení paměti PROM o kapacitě 256 bitů s organizací 32 x 8 bitů

Paměti EPROM (ERASABLE PROGRAMMABLE READ ONLY MEMORY) jsou paměti programovatelné a mazatelné. Tedy paměti, do kterých lze zapsat a zapsanou informaci též zrušit. Postup programování a mazání lze mnohokrát opakovat, proto se tyto paměti označují také symbolicky REPROM (REPROGRAMMABLE READ ONLY MEMORY). K záznamu informací se využívá izolované řídicí elektrody plovoucí tranzistorové buňky FAMOS (FLOATING AVALANCHE INJECTION MOS), do které se při programování akumuluje náboj lavinovou injekcí. Tím vznikne vodivý kanál mezi emitorem a kolektorem. Paměti typu EPROM, ale i PROM, se programují pomocí programátorů pamětí. Tato zařízení jsou součástí vývojových mikropočítačových systémů. Vymazání informace se děje ultrafialovým zářením o vlnové délce okolo 225 nm. Ozářením dochází k odvedení náboje z izolované elektrody do substrátu. Doba mazání je závislá na zdroji ultrafialového záření a je dána energií, potřebnou k výmazu všech bitů v paměti.

Např. paměti EPROM MHB 8708C a MHB 2716 mají minimální dávku energie potřebnou k vymazání informace (intenzita záření x čas) 15 $Wscm^{-2}$ při použití Hg výbojky s vlnovou délkou 253,7 nm.

Pro nedostatek vhodných tuzemských zdrojů se často používají výbojky horského slunce. Tento zdroj produkuje také velké množství tepelného záření, které paměti škodí, a tím se značně sníží počet reprogramovacích cyklů.

Energie vyzářená zářičem ultrafialového světla se provozem snižuje, což způsobuje prodlužování doby mazání. Dochází též k posuvu vlnové délky záření. Ideální by bylo v pravidelných intervalech kontrolovat intenzitu vyzařované energie, ale to je pro běžného uživatele obtížné. Proto se prakticky volí výměna zářiče starého za nový.

Mezi mazatelné paměti patří také paměti EAROM (ELECTRICALLY ALTERNABLE ROM) s plovoucím hradlem. Struktura je obdobná paměti EPROM s centrálním ultrafialovým mazáním informací. Odpadá křemenné okénko a zvláštní hradlo řídí elektrický zápis a mazání náboje v plovoucím hradlu. Programování je relativně složité a vyžaduje zvláštní programátor. Doba mazání se proti paměti EPROM zkracuje pod 1 min., přístupuvá doba je řádově stovky ns, paměť je dražší, ale vyznačuje se nižším příkonem.

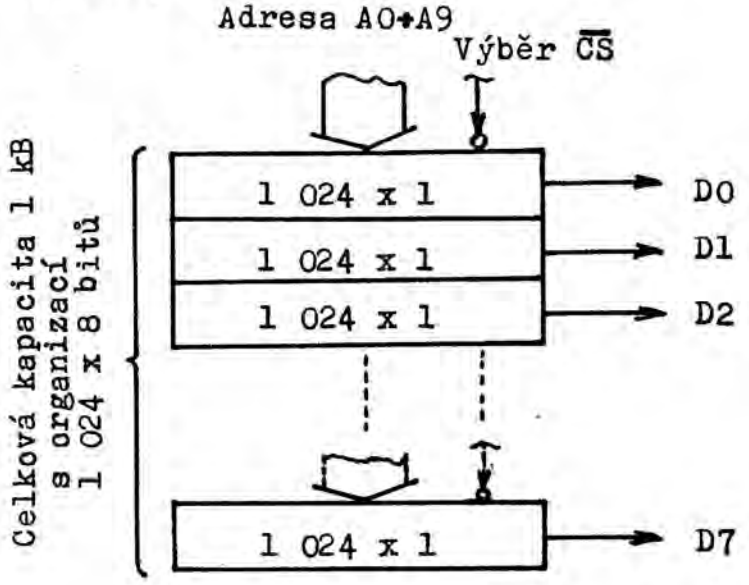
d) dle přístupu k vnitřní informaci na paměti s libovolným přístupem a sériovým přístupem - veškeré polovodičové paměti v technologiích bipolárních, unipolárních, s různou realizací paměťové buňky, určené ke čtení nebo ke čtení i zápisu, jsou paměti s libovolným přístupem k informaci. Podrobnější rozbor byl vysvětlen v této kapitole při objasnění zkratky RAM. Příkladem sériové paměti je posuvný registr.

Prakticky nepřicházejí v úvahu dle bodů a) až d) bipolární dynamické paměti a dynamické paměti jen pro čtení dat.

1.11.2 Typické parametry polovodičových pamětí

Polovodičové paměti charakterizuje jejich kapacita, vnitřní organizace, přístupová neboli vybavovací doba k informaci, spolehlivost, dynamické parametry, cena aj.

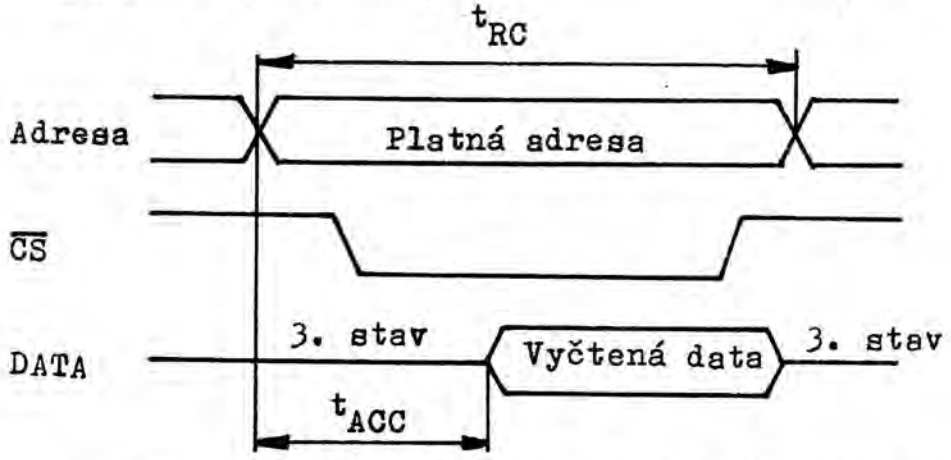
Paměťová kapacita se udává počtem bitů, popř. bytů obsažených v jednom pouzdru. S kapacitou přímo souvisí organizace paměti. Ta se udává výrazem N x B, kde N je počet adresovatelných slov a B definuje hloubku slova. Běžné paměti mají organizaci N x 1 bit, N x 4 bity, N x 8 bitů. Širších paměťových matic se dosáhne vzájemným propojením jednotlivých pouzder. Obr. 27 je příkladem vytvoření paměťové matice PROM o kapacitě 1 kbyte s organizací 1 024 x 8 bitů z paměťových pouzder organizovaných 1 024 x 1 bit. Adresové vodiče A0 ÷ A9 i výběrový vodič $\overline{CS}$ jsou rozvedeny paralelně do všech 8 pouzder, výstupní vodiče D0 ÷ D7 jsou vyvedeny separovaně.



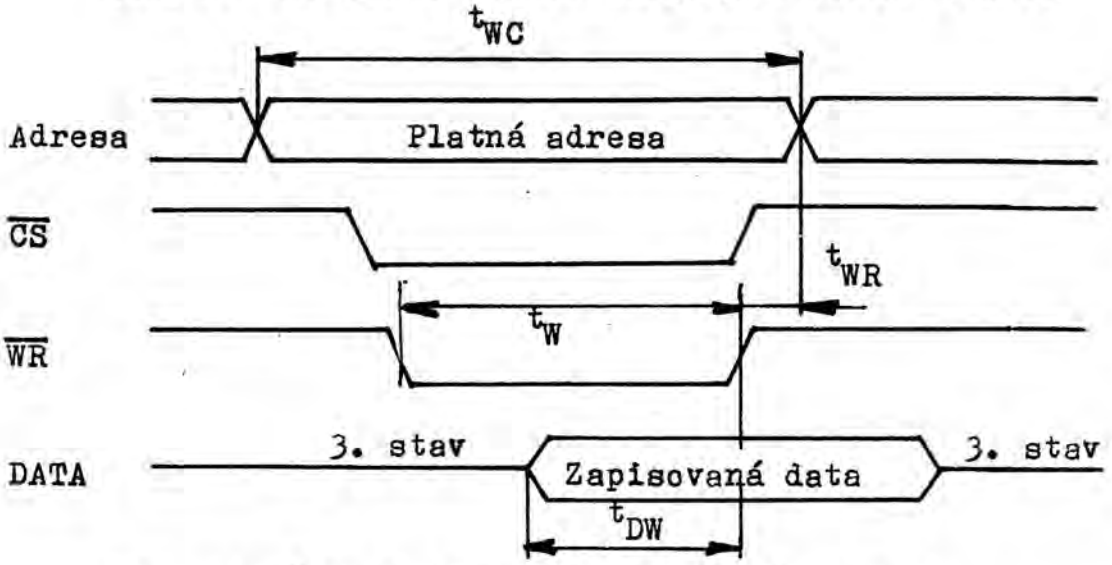
Obr. 27 Paměťová matice PROM 1 kB vytvořená z pouzder organizovaných 1 024 x 1 bit

Přístupová doba k informaci t_{ACC} (ACCESS TIME) čili vybavovací doba je čas, za který po přiložení platné adresy jsou na výstupu paměti k dispozici přečtená data. To vše za podmínky, že paměť je aktivována signálem $\overline{CS}$ (CHIP SELECT). Ve čtecím cyklu je dále důležitá doba t_{RC} (READ CYCLE), která udává minimální potřebný čas pro přečtení jednoho slova. Při zápisu slova do paměti RWM je nejdůležitějším údajem šířka zapisovacího signálu t_W (WRITE TIME), doba předstihu platných dat před ukončením zápisu t_{DW}, doba udávající, jak dlouho musí zůstat platná adresa po ukončení zápisu t_{WR}. Doba t_{WC} (WRITE CYCLE) udává minimální

potřebný čas pro zápis jednoho slova do paměti. Tyto údaje značně ovlivňuje použitá výrobní technologie. Důležité časové údaje při čtecí operaci na paměti typu RWM nebo ROM jsou na obr. 28, údaje při zápisovém cyklu paměti RWM ukazuje obr. 29.



Obr. 28 Důležité časové údaje při čtecí operaci



Obr. 29 Časové údaje při zápisové operaci

Vzhledem k dosud používaným technologiím působí přístupová doba a kapacita paměti proti sobě. Bipolární paměti mají vybavovací doby kratší, ale jejich kapacita je nižší. U unipolárních pamětí jsou vybavovací doby zhruba o řád delší, ale dosahují vyšších kapacit, především u dynamických struktur.

2. MIKROPROCESOR Z80-CPU

2.1 Mikropočítačový systém

Jakýkoliv mikropočítačový systém lze sestavit ze tří základních bloků, umožňující zpracování informací v číslicové formě. Jedná se o:

a) mikroprocesor - centrální procesorová jednotka (CPU - CENTRAL PROCESSING UNIT) je obvod, který provádí vlastní zpracování informací dle určitého algoritmu uloženého v paměti. Mikroprocesor provádí generování adresy, řídicích signálů pro zajištění přenosu dat, zabezpečuje logické a aritmetické operace s daty, zajišťuje provádění instrukcí uložených v paměti a provádí obsluhu přerušení. V mikropočítačovém systému Z-80 zastává funkci centrální jednotky mikroprocesor Z80-CPU, který má ekvivalent v dovozním prvku z NDR pod označením U880D.

b) paměť - slouží k uložení programu, podle kterého mikropočítač pracuje a dále slouží k uložení zpracovávaných informací. Ve většině případů je paměť členěna na dvě části. V jedné části je uložen program sestávající z jednotlivých instrukcí, zatímco druhá část slouží k uchování dat. V systému může být použito rozličných typů polovodičových pamětí. Připojení dynamických pamětí typu RAM, které dosahují největších kapacit, je velmi jednoduché vlivem konstrukce procesního obvodu a řídicí sběrnice.

c) V/V obvody - tento blok zprostředkovává styk mikroprocesoru s okolím, tj. např. s obrazovkovým displejem, který umožňuje zasahovat operátorovi k průběhu řízení, s tiskárnou, děrnopáskovými periferiemi, převodníky atd.

V systému Z-80 je k dispozici celkem pět základních LSI obvodů:

a) Z80-PIO - paralelní vstup/výstup, ekvivalent U855D

b) Z80-SIO - sériový vstup/výstup, ekvivalent U856D

c) Z80-DART - asynchronní sériový vstup/výstup

d) Z80-CTC - čítač/časovač, ekvivalent U857D

e) Z80-DMA - přímý přístup do paměti, ekvivalent U858D

K mikroprocesoru Z80-CPU lze aplikovat i periferní obvody jiných firem,např. firmy Intel.

Na zjednodušeném blokovém schematu(obr.31) jsou jednotlivé bloky mezi sebou propojeny pomocí tří typů sběrnic:

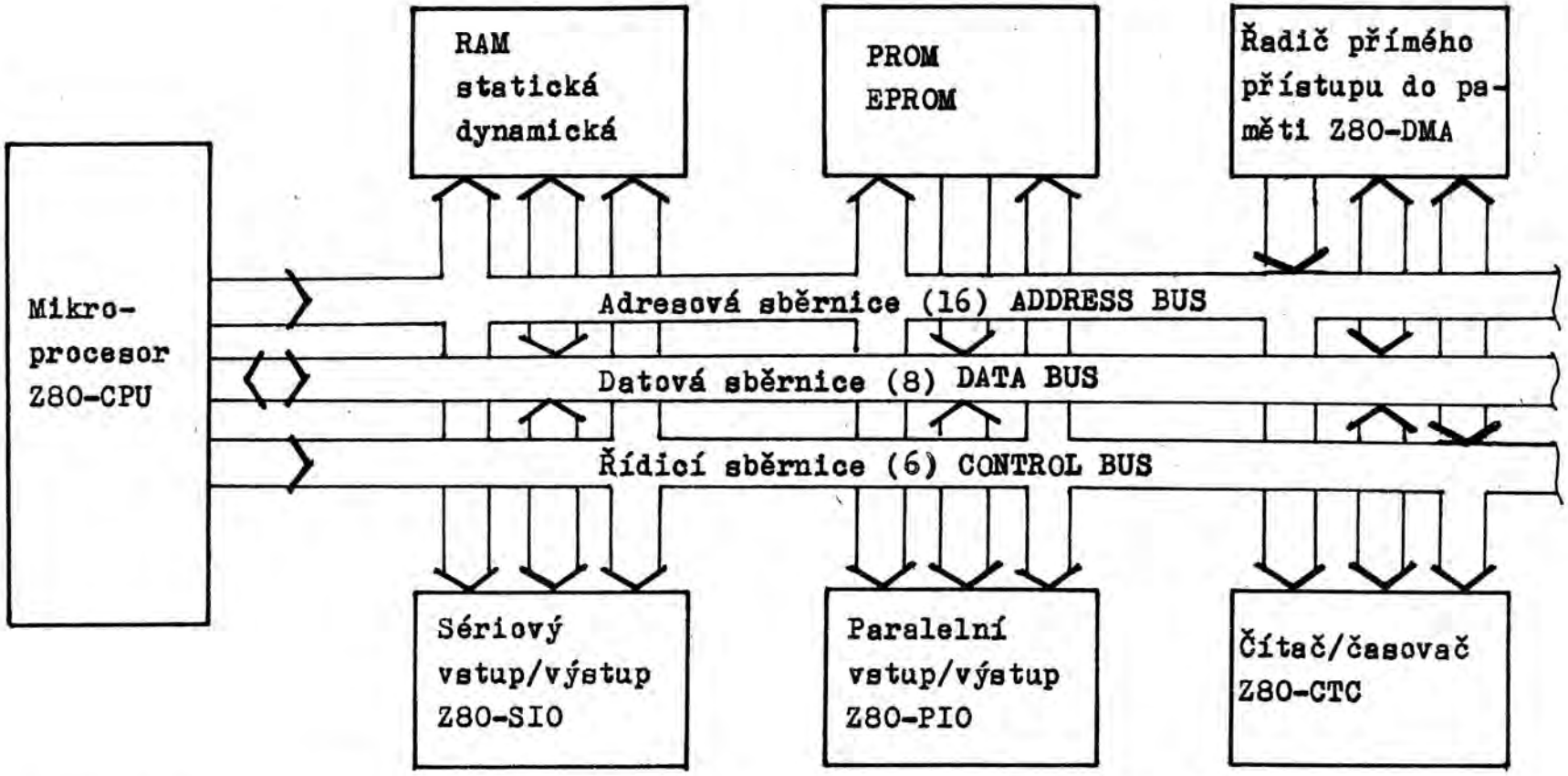
a) adresová sběrnice - tato skupina vodičů přenáší adresu směrem od CPU k paměti anebo k V/V zařízení. Adresa identifikuje paměťovou buňku nebo V/V zařízení. V systému Z-80 je adresová sběrnice jednosměrná, třístavová a má šířku 16 bitů. S touto šířkou má paměťová oblast celkovou kapacitu 64 kByte. Pro adresování V/V zařízení se využívá nižších 8 bitů adresové sběrnice.

b) řídicí sběrnice - je skupina vodičů, po nichž jsou přenášeny informace, koordinující činnost všech tří základních bloků. Řídicí signály mohou být vytvářeny buď mikroprocesorem,nebo jinými zařízeními, schopnými převzít řízení sběrnic(např. Z80-DMA). V systému Z-80 je řídicí sběrnice 6bitová.

c) datová sběrnice - po datových vodičích je přenášena informace ze zdrojového do cílového místa. Za určitou jednotku času lze přenést jen určité množství informací. Po dobu přenosu musí být ostatní zdroje informací, které sdílejí sběrnici, od sběrnice odpojeny tj. nachází se ve vysokoimpedančním stavu. Datová sběrnice v systému Z-80 má šířku 8 bitů, je obousměrná a třístavová.

Mimo výše rozebraných tří skupin signálů jsou jednotlivé bloky propojeny napájecími linkami, hodinovacími signály atd.

Kromě takto jednoduše popsané sběrnice existují sběrnice systémové,umožňující vytvářet multiprocesorové systémy. V těchto systémech je sběrnice sdílena více mikroprocesory. Typicky se jedná o sběrnici MULTIBUS firmy Intel nebo Z-BUS firmy Zilog.



Obr.31 Zjednodušené blokové schéma mikropočítače

2.2 Součástkový přehled řady Z-80 (U880)

Firma Zilog uvedla na trh v roce 1976 moderní paralelní osmibitový mikroprocesor 2. generace Z80-CPU, který byl v příštích letech doplněn několika dalšími rychlostními variantami. K procesnímu typu je dodáváno pět programovatelných podpůrných obvodů Z80-PIO, Z80-SIO, Z80-CTC, Z80-DMA a Z80-DART. Kompletní systém Z-80 je vyráběn od roku 1980 v NDR pod souhrnným názvem U880. Procesní element má označení UB880D, podpůrné obvody nesou typový kód UB855D, UB856D, UB857D, UB858D a UB8563D. Obvody jsou plně ekvivalentní řadě Zilog. V NDR se nejvíce těchto obvodů používá v počítačích firmy Robotron, u nás se uplatňují např. jako řídicí systémy textilních strojů.

Srovnávací tabulka 1 ukazuje přehled prvků řady Z-80 a ekvivalentní řady U880. A to jednak v základní rychlostní variantě, kde Z80-CPU (2,5 MHz) odpovídá obvodu UB880D, a jednak v rychlostně vyšší variantě, kde Z80A-CPU (4 MHz) odpovídá obvodu UA880D. V systémech, vytvářených z této součástkové základny, lze uplatnit podpůrné obvody řady MH 3000, které jsou společné řadě 8080 anebo některé speciální programovatelné prvky systému 8080.

Srovnávací tabulka řady Z-80 a U880 — Tabulka 1

ZILOG - USA		VEB FUNKWERK ERFURT - NDR	
Typ	Název obvodu	Analog	Název obvodu
Z80-CPU (2,5 MHz)	CENTRAL PROCESSING UNIT	UB880D (2,5 MHz)	Centrální procesní jednotka
Z80A-CPU (4 MHz)		UA880D (4 MHz)	
Z80-PIO Z80A-PIO	PARALLEL INPUT/OUTPUT CONTROLLER	UB855D UA855D	Obvod paralelního styku vstup/výstup
Z80-SIO Z80A-SIO	SERIAL INPUT/OUTPUT CONTROLLER	UB856D UA856D	Obvod sériového styku vstup/výstup
Z80-CTC Z80A-CTC	COUNTER/TIMER CONTROLLER	UB857D UA857D	Generátor hod. impulsů - čítač/časovač
Z80-DMA Z80A-DMA	DIRECT MEMORY ACCESS CONTROLLER	UB858D UA858D	Obvod přímého přístupu do paměti
Z80-DART Z80A-DART	DUAL ASYNCHRONOUS RECEIVE/TRANSMITTER	UB8563D UA8563D	Obvod asynchronního styku vstup/výstup

Tyto aplikační možnosti vyplývají nejen z ekonomických důvodů, ale i personálních, poněvadž hlavní konstruktéři stavebnice Z-80 byli tvůrci stavebnice MCS-8 a MCS-80 firmy Intel.

2.3 Srovnání řady Z-80 s I 8080

Obvody řady Z-80 poskytují uživateli několik výhod oproti I 8080:

a) jednotné napájení + 5 V u veškerého použitého sortimentu prvků

b) zdvojené vnitřní pracovní registry, umožňující operativní úschovu pracovních registrů jedné sady při jednoúrovňovém volání podprogramu přerušení

c) vyšší frekvence hodin. Základní typ Z80-CPU má periodu hodin 400 ns. Odvozené verse Z80A-CPU 250 ns a Z80B-CPU 166 ns.

d) množina instrukcí I 8080 je podmnožinou procesních instrukcí Z80-CPU. Mikroprocesor má rozšířenou instrukční síť z původních 78 instrukcí, použitých u typu I 8080, na celkem 158 instrukcí. Zejména instrukce blokového přenosu, použité v několika versích při transportu dat v paměti, ze vstupů a na výstupy, výrazně zkracují programy.

e) pružný systém přerušení, pracující v režimu 0, 1, 2, je ovládaný vstupními signály $\overline{INT}$ a $\overline{NMI}$. Přerušovací systém dovoluje volání obslužného podprogramu v rozsahu celé operační paměti. Volací adresy lze podle potřeby měnit, což umožňuje skutečně universální použití mikroprocesoru. Pokud použijeme pro mikroprocesor, nastavený do režimu 2, speciální podpůrné periferní obvody, není pro přerušení zapotřebí dalších prvků, protože tyto periferní obvody mají systém přerušení již zabudován.

f) Z80-CPU dovoluje několik způsobů adresování, které nejsou možné u typu I 8080, např. adresování pomocí dvou indexregistrů, tzv. indexové adresování nebo relativní adresování za spolupráce programového čítače.

g) mikroprocesor obsahuje registr pro obnovování dynamických pamětí (tzv. Refresh registr). Obsah tohoto registru se zvyšuje o jednotku při každém čtení nové instrukce z operační paměti a přivádí se na adresovou sběrnici v době, kdy se načtená instrukce dekóduje v mikroprocesoru. Tím se dosáhne toho, že inkrementace Refresh registru nastává současně s instrukčním cyklem M1 a nedochází k časovému zpoždění. Za těchto podmínek je možné obnovování dynamické paměti bez zpomalení činnosti mikroprocesoru.

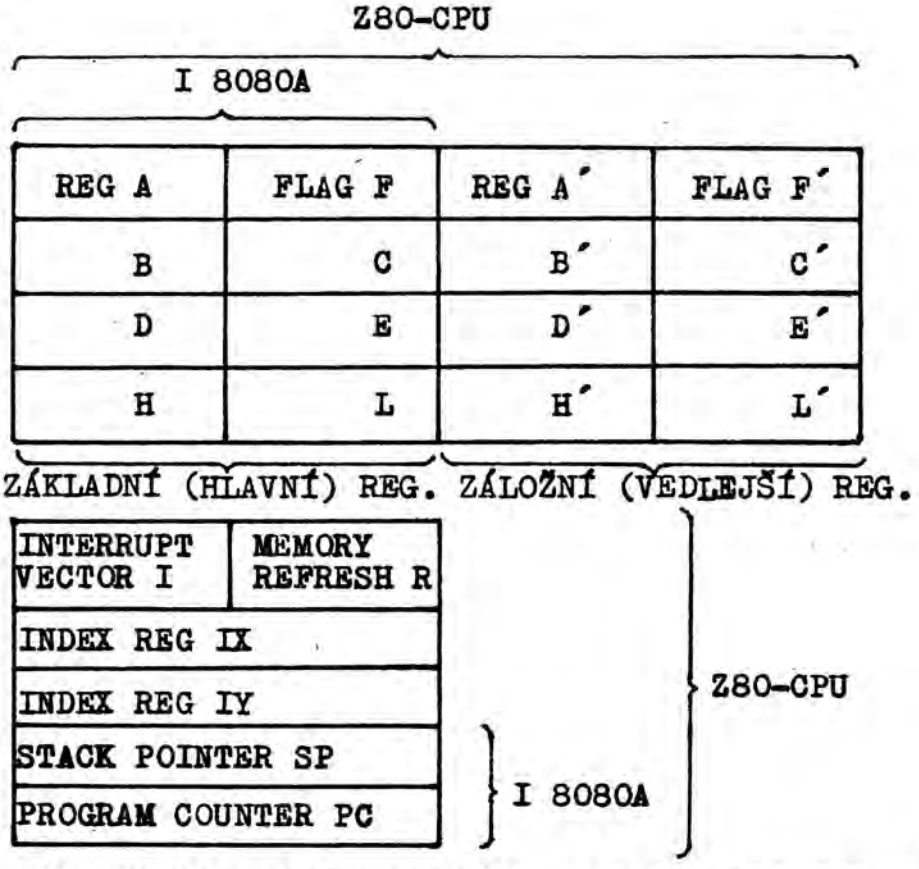
Uvedené vlastnosti značně zjednodušují návrh systému tj. šetří počet nezbytně nutných podpůrných obvodů, snižují nároky na paměť, optimalizují návrh programů. Celkově se zvyšuje spolehlivost při vyšší operační rychlosti a nižších výrobních nákladech.

Následující přehled parametrů Z80-CPU a I 8080A shrnuje vlastnosti obou systémů.

Základní parametry	Mikroprocesor Z80-CPU	Mikroprocesor I 8080A
technologie	N-MOS	N-MOS
počet aktivních prvků	8 000	5 000
napájecí napětí	+ 5 V/150 mA	+ 12 V/10 mA, + 5 V/80 mA, - 5 V/1 mA
pouzdření	40 DIL	40 DIL
systémové hodiny	jednofázové	dvoufázové s amplitudou + 9 V (I 8224)
adresová sběrnice	16 bitů	16 bitů
datová sběrnice	8 bitů	8 bitů (I 8228)
charakter I/O	paralelní	paralelní
řídicí sběrnice	6 bitů	5 bitů (I 8228)
nemaskovatelné přerušení	ano	ne
maskovatelné přerušení	ano	ano

Základní parametry	Mikroprocesor Z80-CPU	Mikroprocesor I 8080A
obnovování dynamických pamětí	ano	ne
rychlost základního typu	400 ns	488 ns
soubor základních instrukcí	158	78
formát instrukcí	1 ÷ 4 byte	1 ÷ 3 byte
zásobníková paměť	vnější	vnější
možnost DMA	ano	ano

Rozšířená konfigurace vnitřních registrů Z80-CPU oproti I 8080A je na obr. 32. Význam těchto registrů popisuje kapitola 2.4.1.



Obr.32 Srovnání vnitřních registrů Z80-CPU a I 8080A

2.4 Architektura Z80-CPU

Mikroprocesor Z80-CPU (CENTRAL PROCESSING UNIT) je osmibitová procesorová jednotka vyráběná technologií N-MOS. Vhodná koncepce CPU a pomocných obvodů dovoluje velmi efektivně konstruovat mikroprocesorové systémy s minimálním množstvím doplňkové logiky. Sestavu mikropočítače tvoří Z80-CPU ve spojení s polovodičovými pamětmi a obvody I/O. Mikroprocesor obsahuje 2 krát 6 vnitřních pracovních registrů dat, dva osmibitové střadače, dva osmibitové registry příznaků, osmibitovou paralelní aritmetickou jednotku ALU, šestnáctibitový programový čítač PC, šestnáctibitový registr adresy zásobníkové paměti SP, registr vektoru přerušení I, obvody obnovy dynamických pamětí a řídící/časovací obvody s dekodérem instrukcí. Z80-CPU obsahuje rovněž dva šestnáctibitové indexregistry, využívané při indexovém adresování paměti.

Pracovní registry zahrnují dvě skupiny 6 všeobecně použitelných registrů B, C, D, E, H, L a B′, C′, D′, E′, H′ a L′, které mohou být použity samostatně jako osmibitové registry nebo jako šestnáctibitové registrové páry. Mikroprocesor má také dva osmibitové střadače A, A′ a dva registry testovatelných příznakových bitů F, F′. Všechny registry nebo páry jsou programově přístupné. Tyto dvě skupiny registrů mohou sloužit také pro velmi rychlou odezvu na příchod přerušovacího signálu. Šestnáctibitový STACK POINTER umožňuje adresovat zásobníkovou paměť ve velmi širokém rozsahu, což dovoluje pamatovat značné množství odskokových adres při vykonávání procedur jak hlavního programu, tak procedur zajišťujících ošetření přerušení. Dva šestnáctibitové registry IX, IY rozšiřují možnosti adresování operandů. REFRESH registr zajišťuje automatickou obnovu externích dynamických pamětí při každém FETCH cyklu. Registr INTERRUPT je používán pro vytváření osmi horních bitů šestnáctibitové adresy při ošetření přerušení v nastaveném režimu 2. Osm spodních bitů adresy generuje periferie žádající o obsloužení přerušení. Osmibitová ALU umožňuje provádění operací jak s binárními, tak s dekadickými čísly v rozsahu osmi popřípadě šestnácti bitů. CPU dovoluje softwarově

zvolit tři možné způsoby zpracování přerušení MODE 0, 1, 2 s možností maskování přerušení a jedno nemaskovatelné přerušení. K externímu řízení procesoru slouží signály $\overline{WAIT}$, tj. požadavek na stav WAIT, ve kterém CPU zůstane, pokud externí paměť nebo periferie není hotova k předání požadovaných dat. Signál $\overline{NMI}$ - nemaskovatelný interrupt, který je testován na konci každé vykonané instrukce, $\overline{INT}$ - interrupt maskovaný klopným obvodem přerušení je testován rovněž na konci každé vykonané instrukce. Signál $\overline{RESET}$ nuluje registry CPU a uvádí celou jednotku do počátečního stavu. Signál $\overline{BUSRQ}$ je požadavek vnější jednotky na sběrnici a uvádí mikroprocesor do vysokoimpedančního stavu na sběrnicích. Blokové schéma mikroprocesoru Z80-CPU je na obr. 33.

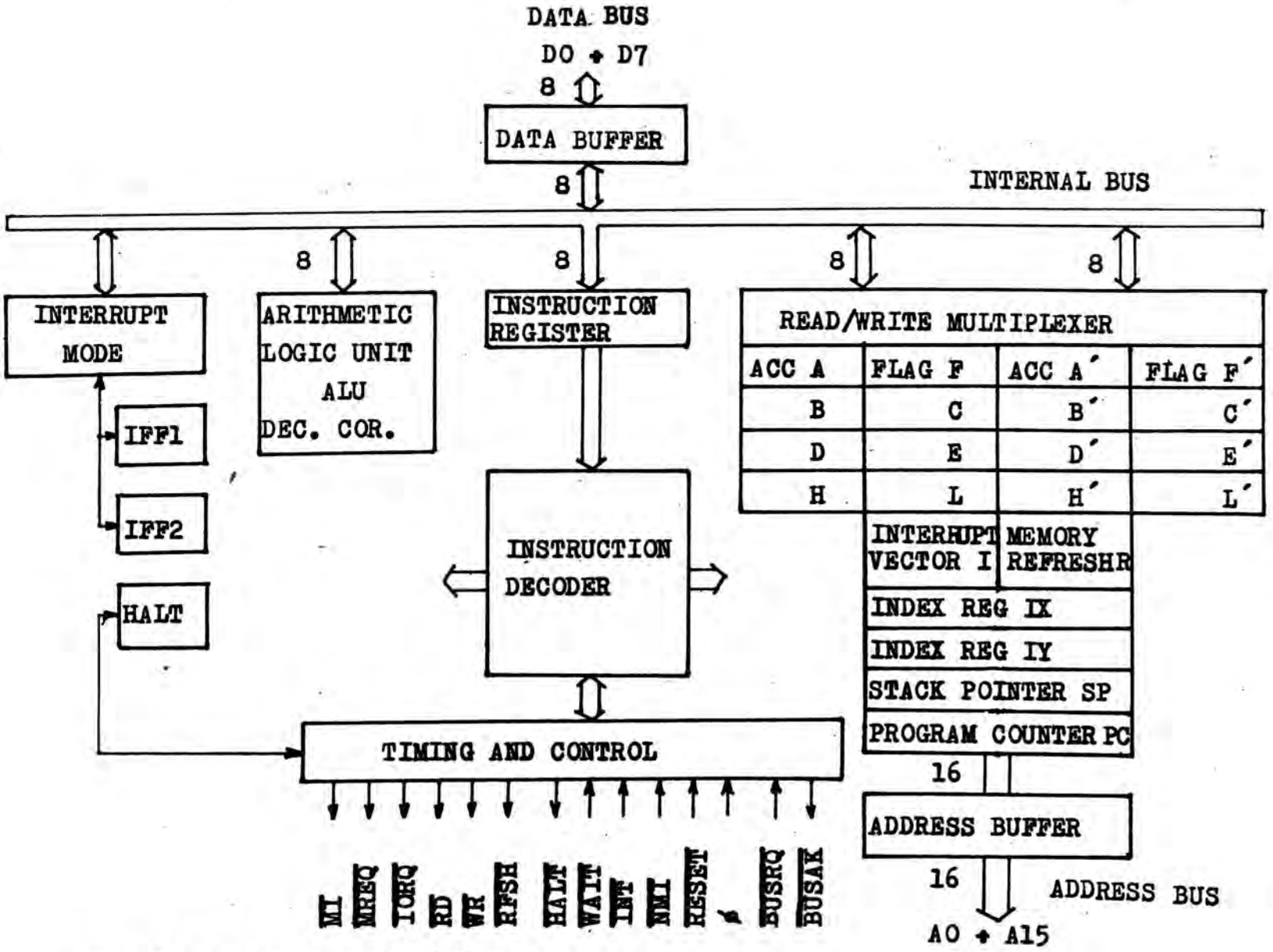
2.4.1 Vnitřní bloky CPU

Vnitřní strukturu mikroprocesoru lze rozdělit na následující bloky:

a) střadače (A, A´ - ACCUMULATOR) a registry příznaků (F, F´ - FLAGS) - jsou speciální typy osmibitových registrů. Ve střadači se uchovávají výsledky aritmetických a logických operací prováděných s daty, zatímco reg. F obsahuje stavové informace tj. přenos, nulovost výsledku, znaménkový příznak, paritu, pomocný přenos, indikátor sčítání/odčítání.

b) obecně použitelné registry B, C, D, E, H, L a záložní registry B´, C´, D´, E´, H´, L´-jsou osmibitové registry s možností slučování do registrových párů BC, DE, HL (BC´, DE´, HL´) o dvojnásobné šířce dat tj. šestnáct bitů. Tyto dvě sady registrů umožňují velmi rychlý úklid obsahu základních registrů do záložních a zpětné obnovení základních registrů při obsluze přerušení pomocí nových instrukcí EX a EXX.

c) programový čítač (PC - PROGRAM COUNTER) - definuje šestnáctibitovou adresu paměti, ze které je čtena instrukce. Obsah programového čítače se zvýší o jedničku po každém provedení čtení instrukce. Pakliže se v programu provádí skoky, je PC automaticky nastaven na novou požadovanou hodnotu.



Obr.33 Blokové schéma Z80-CPU

d) ukazatel zásobníku (SP - STACK POINTER) - je registr, ve kterém je uložena šestnáctibitová adresa vrcholu zásobníku, umístěného kdekoliv v paměťové matici 64 kByte. Zásobník je organizován jako paměť typu LIFO (LAST IN, FIRST OUT - poslední dovnitř, první ven), do kterého se vkládají data instrukcemi PUSH a vybírají instrukcemi POP. Z toho vyplývá, že data čtená ze zásobníkové paměti jsou vždy posledními zapsanými daty. Zásobníková paměť umožňuje vhnizdování podprogramů, přerušovacích podprogramů a usnadňuje manipulaci s daty.

e) index registry (IX, IY - INDEX REGISTER) - jsou dva nezávislé registry, v nichž je uložena šestnáctibitová bázová adresa, udávající paměťovou oblast, do níž mají být zapsána nebo čtena data. Vytvoření konečné (tzv. indexované) adresy získáme součtem obsahu index registru tj. báze a operandové části instrukce tzv. posunutí (DISPLACEMENT). Např. indexové adresování používají instrukce LD r, (IX+d), LD r,(IY+d) nebo LD (IX+d),r ,LD (IY+d),r atd.

f) registr obnovování dynamických pamětí (R - MEMORY REFRESH) - pracuje jako sedmibitový čítač, inkrementovaný po každém čtení operačního kódu instrukce z paměti. V době, kdy CPU dekóduje právě načtený operační kód instrukce, je obsah tohoto čítače vložen na nižších sedm bitů adresové sběrnice současně s aktivací signálu $\overline{\text{RFSH}}$, což umožňuje obnovení dat v připojené dynamické paměti, aniž by došlo ke zpomalení práce mikroprocesoru.

g) registr stránkové adresy přerušení (I - INTERRUPT VECTOR) - umožňuje CPU pracovat v takovém režimu přerušení, kde odezvou na přerušení je volání jakékoliv paměťové adresy. V I registru je uložena vyšší část šestnáctibitové nepřímé adresy (8 bitů), zatímco zbývající část (nižších 8 bitů) dodává zařízení, tj. obvod Z80-PIO, Z80-SIO, Z80-CTC, Z80-DMA, které požaduje přerušení. Tímto způsobem může být obslužný program přerušení uložen kdekoliv v paměti.

h) aritmeticko-logická jednotka (ALU - ARITHMETIC LOGIC UNIT) - blok ALU provádí aritmetické a logické operace s daty. Pro

zajištění jejich provedení ALU komunikuje s 22 vnitřními programátorsky přístupnými registry, s instrukčním registrem a s blokem řízení. Aritmeticko-logická jednotka umožňuje provádět tyto operace s daty:

- sčítání, odčítání
- inkrementace a dekrementace
- logický součet OR, logický součin AND a exclusive OR
- posuny a rotace
- komparace
- nastavení bitu
- nulování bitu
- testování bitu.

Blok dekadické korekce, který je součástí ALU, zajišťuje převod šestnáctkového výsledku na dekadický po provedeném aritmetickém sčítání nebo odčítání za pomoci indikátorových bitů.

i) instrukční registr, instrukční dekodér, časování a řízení (INSTRUCTION REGISTER, INSTRUCTION DECODER, TIMING AND CONTROL) - po načtení instrukce z paměti do instrukčního registru je její obsah okamžitě dekódován. Funkce, které mají být na základě tohoto dekódování provedeny, se zajišťují blokem časování a řízení. Blok časování a řízení koordinuje součinnost veškerých vnitřních částí Z80-CPU a generuje řídící signály vystupující z mikroprocesoru.

j) blok režimu přerušení (INTERRUPT MODE, IFF1, IFF2) - blok režimu přerušení sestává mimo jiné z klopných obvodů IFF1 a IFF2. Klopný obvod IFF1 je ovládán jako v systému I 8080, tzn., že je setován instrukcí EI a resetován instrukcí DI nebo potvrzením přerušení, čímž se blokuje případná další žádost o přerušení na vstupu $\overline{\text{INT}}$. Poslední případ nulování klopného obvodu IFF1 nastane signálem $\overline{\text{RESET}}$.
Druhý klopný obvod IFF2 pracuje identicky s klopným obvodem IFF1 při přerušení vstupem $\overline{\text{INT}}$, při instrukcích EI, DI a RESETu. Klopný obvod IFF2 se uplatní v případě žádosti o nemaskovatelné přerušení (vstup $\overline{\text{NMI}}$), která má maximální prioritu na odbavení a tudíž může přerušit přerušovací podprogram, vyžádaný vstupem $\overline{\text{INT}}$. Pro zajištění informace o stavu

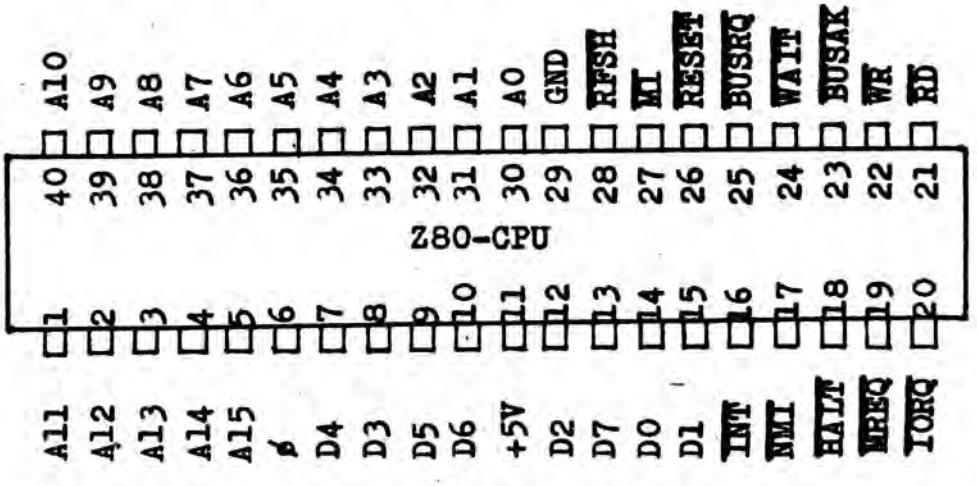
přerušovacího systému před odbavením nemaskovatelného přerušení je stav klopného obvodu IFF1 přepsán do klopného obvodu IFF2. Přitom IFF1 se nuluje a tím blokuje odbavení případných dalších požadavků přerušení $\overline{INT}$.
Obnova stavu klopného obvodu IFF1 od klopného obvodu IFF2 nastává programově až po skončení obslužného podprogramu nemaskovatelného přerušení. Tuto činnost zajistí instrukce RETN (RETURN FROM NONMASKABLE INTERRUPT).

k) blok stavu HALT - vložením instrukce HALT do programu dojde k nahození vnitřního klopného obvodu HALT. Na základě dekódování operačního kódu instrukce vzniká aktivní výstupní signál $\overline{HALT}$ a nadále probíhá obnova dynamických pamětí, poněvadž instrukce HALT vyvolá systematické generování prázdné instrukce NOP.

l) datový registr a adresový registr (DATA BUFFER AND ADDRESS BUFFER) - výstup dat nebo adres na vnější datovou nebo adresovou sběrnici z interní sběrnice a vstup dat do mikroprocesoru zajišťují bloky datové a adresové sběrnice.

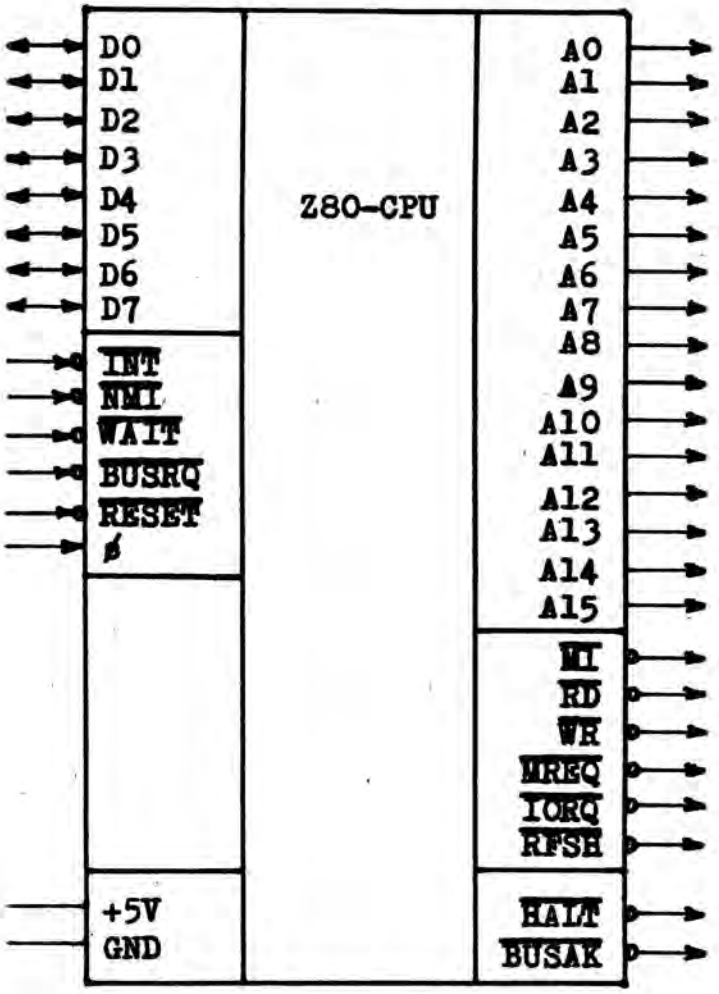
2.4.2 Přehled uspořádání vývodů pouzdra

Mikroprocesor je zapouzdřen v 40vývodovém pouzdru DIL s napájením + 5 V. Fyzické umístění vývodů pouzdra je na obr.34.



Obr. 34 Umístění vývodů pouzdra Z80-CPU

Následující obr.35 ukazuje přehled vývodů mikroprocesoru Z80-CPU. Šipky označují jednosměrnost nebo obousměrnost signálů a zároveň jejich orientaci.



Obr.35 Přehled vývodů mikroprocesoru Z80-CPU

D0 ÷ D7 datová sběrnice (DATA BUS)	třístavový vstup/výstup, aktivní úroveň H. Vodiče D0 ÷ D7 nesou osmibitovou datovou obousměrnou sběrnici, po které jsou přenášena data z a do paměti nebo V/V zařízení.

A0 ÷ A15 adresová sběrnice (ADDRESS BUS)	třístavová výstupní sběrnice, aktivní úroveň H. Vodiče A0 ÷ A15 tvoří šestnáctibitovou adresovou sběrnici. K adresování paměti se využívá 16 bitů, zatímco k adresování V/V zařízení se využívá nižších 8 bitů sběrnice, což umožňuje lokalizovat 256 vstupních a 256 výstupních zařízení. V době provádění obnovy dat v dynamických pamětech je na nižších 7 bitech adresové sběrnice adresa obnovení. Při přímém přístupu do paměti (DMA přenos) je adresová sběrnice ve vysokoimpedančním stavu.
$\overline{RD}$ čtení dat (READ)	třístavový výstup s aktivní úrovní L. Mikroprocesor generováním řídicího signálu $\overline{RD}$ vydává požadavek na čtení dat z paměti nebo V/V zařízení.
$\overline{WR}$ zápis dat (WRITE)	třístavový výstup s aktivní úrovní L. Z80-CPU signálem $\overline{WR}$ informuje paměť nebo V/V zařízení, že na datové sběrnici jsou k převzetí data.
$\overline{M1}$ prvý strojový cyklus (MACHINE CYCLE ONE)	výstupní signál, aktivní úroveň L. Výstup $\overline{M1}$ patří do skupiny řídicích signálů a udává, že právě probíhající cyklus je cyklem čtení operačního kódu instrukce. Při provádění instrukcí, které mají 2 byte dlouhý operační kód, se generuje signál $\overline{M1}$ pro každý byte operačního kódu. Řídicí signál $\overline{M1}$ spolu se signálem $\overline{IORQ}$ vytváří cyklus potvrzení přerušení $\overline{INTA}$, který je pod tímto označením znám ze systému I 8080.
$\overline{IORQ}$ požadavek na V/V zařízení (INPUT/OUTPUT REQUEST)	třístavový výstup s aktivní úrovní L. Jedná se o řídicí signál, kterým se V/V zařízení upozorňuje, že na nižších 8 bitech adresové sběrnice je platná adresa. Dohromady se signálem $\overline{M1}$ vytváří signál $\overline{INTA}$, po jehož

dobu trvání dodává zařízení, které požaduje přerušení, vektor přerušení na datovou sběrnici.

$\overline{\text{MREQ}}$
požadavek na paměť
(MEMORY REQUEST)

třístavový výstup s aktivní úrovní L. Tento řídicí signál udává, že na adresové sběrnici je platná adresa paměti A0 ÷ A15, z které nebo do které se mají přesunout data. Signály A0 ÷ A15, D0 ÷ D7, $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$ jsou konstruovány třístavově proto, aby byl umožněn přímý přístup do paměti řízený obvodem Z80-DMA.

$\overline{\text{RFSH}}$
obnova dat v dynamických pamětech
(REFRESH)

třístavový výstup, aktivní do L. V době trvání řídicího signálu $\overline{\text{RFSH}}$ je na nižších 7 bitech adresové sběrnice platná adresa obnovení dynamické paměti.

$\overline{\text{RESET}}$
nulování
(RESET)

vstupní signál s aktivní úrovní L. Vstup $\overline{\text{RESET}}$ zabezpečuje počáteční inicializaci mikroprocesoru:
- programový čítač PC se nastaví na adresu 0000
- registr R se nastaví na nulu
- registr I se nastaví na nulu
- klopné obvody uvolnění přerušení IFF1 a IFF2 se nulují (přerušení je blokováno)
- režim přerušení je nastaven na MODE 0

Během nulování Z80-CPU je datová a adresová sběrnice ve třetím stavu a řídicí sběrnice v neaktivním stavu. Signál $\overline{\text{RESET}}$ musí trvat nejméně tři takty.

$\overline{\text{HALT}}$
dynamický stop
(HALT STATE)

výstupní signál s aktivní úrovní L. Mikroprocesor signálem $\overline{\text{HALT}}$ signalizuje, že provádí instrukci HALT. Touto instrukcí se přivede Z80-CPU do stavu opakovaného provádění instrukce NOP (NO OPERATION). V cyklickém provádění instrukce NOP mikroprocesor setrvává až do okamžiku vzniku

přerušení maskovatelného nebo nemaskovatelného, případně do vzniku signálu RESET. Během provádění instrukce HALT je zajištěno obnovování dynamických pamětí vzhledem ke generování instrukcí NOP.

BUSRQ
požadavek na převzetí řízení sběrnic
(BUS REQUEST)

vstupní signál s aktivní úrovní L. V systémech, kde pracuje více aktivních zařízení, se požadavek na převzetí řízení sběrnic předává mikroprocesoru přivedením úrovně L na vstup BUSRQ. Z80-CPU se od sběrnic odpojí, tj. adresová, datová a řídicí sběrnice se nastaví do třetího stavu. Ve vysokoimpedančním stavu sběrnice setrvávají po celou dobu aktivního signálu BUSRQ. Požadavek BUSRQ je testován na konci každého strojového cyklu CPU a má vyšší prioritu než NMI.

BUSAK
potvrzení uvolnění sběrnic
(BUS ACKNOWLEDGE)

výstupní signál s aktivní úrovní L. Výstupním signálem BUSAK mikroprocesor upozorňuje zařízení, které požadovalo převzetí sběrnic, že se odpojil od sběrnic.

Ø
taktovací impulsy
(CLOCK PULSE)

vstupní hodinovací signál. Na tento vstup se přivádí jednofázové systémové hodiny se střídou 1 : 1, s TTL úrovní. Pro zajištění zvýšené úrovně H je nutné zapojit mezi vývod a napájení + 5 V odpor 330Ω.

WAIT
čekací stav CPU
(WAIT STATE)

vstupní signál s aktivní úrovní L. Aktivní úroveň na vstupu WAIT upozorňuje mikroprocesor, že zařízení, s kterým provádí komunikaci, není ještě připraveno s Z80-CPU spolupracovat. Tímto způsobem se synchronizuje činnost procesní jednotky s pomalejšími zařízeními. Dobu, po kterou CPU čeká na pomalejší zařízení, nesmíme v případě použití dynamických pamětí prodloužit nad hodnotu, po kterou výrobce dynamických pamětí zaručuje platnost dat v paměti. Během čekacího

stavu mikroprocesor neprovádí obnovu dynamických pamětí.

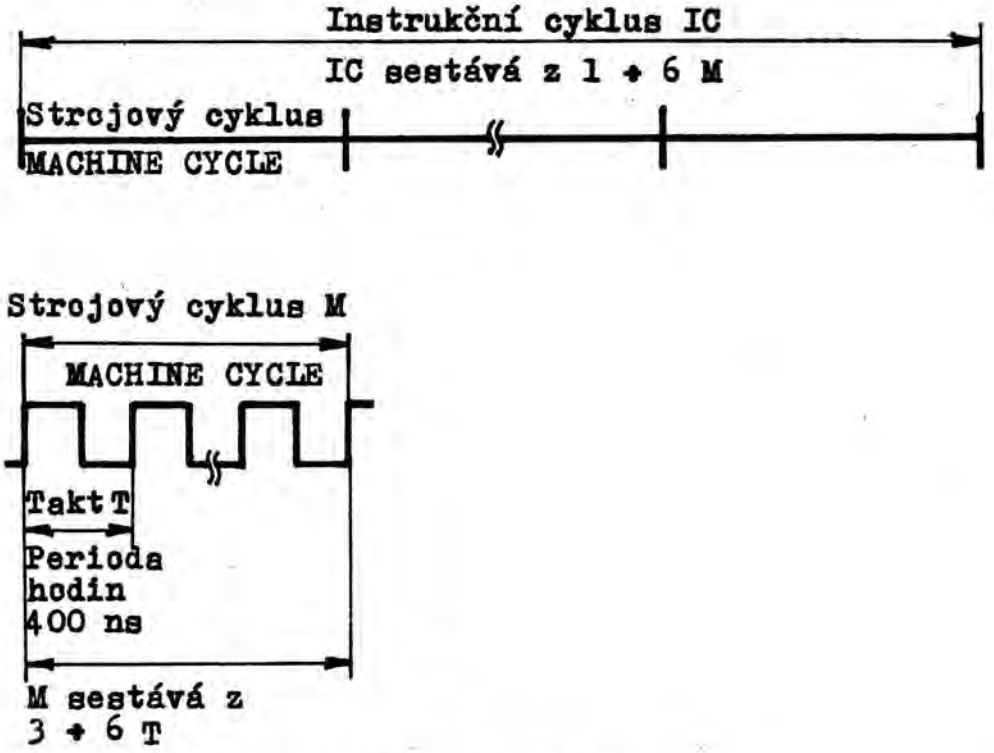
$\overline{\text{INT}}$ maskovatelné přerušení (INTERRUPT REQUEST)	vstupní signál s aktivní úrovní L. Aktivní úroveň na vstupu $\overline{\text{INT}}$ upozorňuje mikroprocesor, že některé V/V zařízení požaduje maskovatelné přerušení. Z80-CPU požadavek přerušení vezme na vědomí až na konci prováděné instrukce za předpokladu, že vnitřní klopné obvody uvolnění přerušení (IFF) jsou nastaveny do stavu, kdy neblokují přerušení a signál $\overline{\text{BUSRQ}}$ není aktivní. Povolení přerušení se provádí instrukcí EI (ENABLE INTERRUPT). Blokování přerušení zabezpečí instrukce DI (DISABLE INTERRUPT). Akceptování požadavku na přerušení potvrzuje Z80-CPU současnou aktivací signálů $\overline{\text{MI}}$ a $\overline{\text{IORQ}}$ (cyklus $\overline{\text{INTA}}$). Mikroprocesor pracuje ve třech režimech přerušení, tj. MODE0,1,2.
$\overline{\text{NMI}}$ nemaskovatelné přerušení (NON MASKABLE INTERRUPT)	vstupní signál s aktivní hranou L. Nemaskovatelné přerušení má nejvyšší prioritu. Toto přerušení je vždy Z80-CPU akceptováno pokud není aktivní signál $\overline{\text{BUSRQ}}$. Signál $\overline{\text{NMI}}$ způsobuje restart na adrese 66H. Vznik přerušení je zajištěn přechodem signálu z H do L. Stav programového čítače je po přijetí nemaskovatelného přerušení automaticky uložen do vnější zásobníkové paměti.

2.5 Časový průběh vykonání instrukce

Program uložený v paměti sestává z jednotlivých instrukcí. Instrukce potřebuje na načtení z paměti do CPU a provedení určitý čas, který se označuje jako instrukční cyklus IC. Instrukční cyklus se rozpadá na 1 ÷ 6 strojových cyklů (MACHINE CYCLES) M1 ÷ M6 a tyto jsou tvořeny 3 ÷ 6 periodami hodin ϕ neboli takty (TIME PERIOD) T3÷T6. Výjimečné doby potřebují instrukce přenosu bloků dat a vyhledávání v bloku dat (LDIR, LDDR, INIR, INDR,

OTIR, OTDR, CPIR, CPDR), u kterých je počet taktů závislých na zpracovávaných zónách dat. Dobu na načtení a provedení konkrétní instrukce lze přesně stanovit na základě známého počtu taktů.

Mikroprocesor Z80-CPU pracuje s nejkratší periodou hodin 400 ns, což odpovídá hodinovému kmitočtu 2,5 MHz. Firma Zilog nabízí procesní čipy i s vyššími frekvencemi; např. typ Z80A má nejvyšší kmitočet 4 MHz a typ Z80B 6 MHz, to odpovídá taktu 250 ns a 166 ns. Vztah mezi instrukčním cyklem, strojovým cyklem a taktem je na obr.36.



Obr.36 Rozdělení instrukčního cyklu

Strojové cykly M mohou být prodlouženy o libovolný počet čekacích taktů TW aktivací signálu WAIT. Čekací takty se vkládají pouze mezi takt T2 a T3. Při strojovém cyklu čtení/zápis dat do V/V zařízení mikroprocesor automaticky vkládá jeden čekací takt a při přerušovacím strojovém cyklu se automaticky vkládají dva čekací takty.

Zpracování celého instrukčního souboru v systému Z-80 je řešeno během 7 typů strojových cyklů:

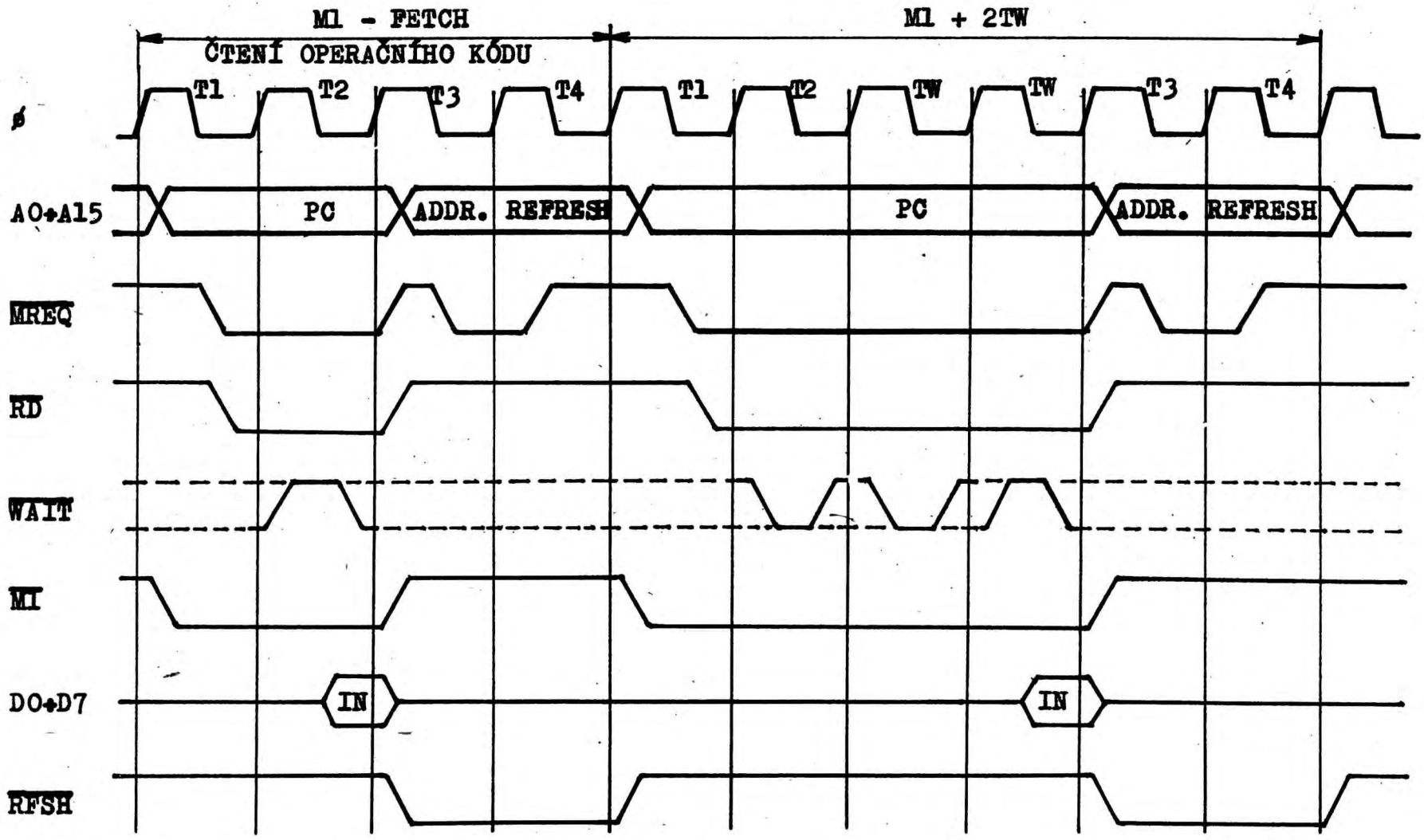
- čtení operačního kódu instrukce, tj. cyklus M1 (FETCH)
- čtení/zápis dat do V/V zařízení
- čtení/zápis dat do paměti
- cyklus požadavku/potvrzení požadavku na přerušení
- cyklus požadavku/potvrzení požadavku na nemaskovatelné přerušení
- cyklus požadavku/potvrzení požadavku na převzetí řízení sběrnic
- ukončení provádění instrukce HALT

2.5.1 Čtení operačního kódu instrukce

Základním cyklem, na který navazuje další činnost Z80-CPU, je čtení operačního kódu instrukce. V prvé polovině tohoto cyklu se z paměti přečte operační kód instrukce. Hodnota programového čítače PC je na adresové sběrnici od začátku operačního cyklu. Spádová hrana $\overline{\text{MREQ}}$ se používá pro povolení pamětí, signál $\overline{\text{RD}}$ povoluje připojit data paměti na datovou sběrnici. Data procesor vzorkuje náběžnou hranou hodin v taktu T3. V druhé polovině operačního cyklu se instrukce dekóduje. Při dekódování se provádí obnovování dynamických pamětí. Cyklus M1 může být prodloužen ze čtyř taktů na libovolný počet taktů vložením TW taktů pomocí aktivace vstupu $\overline{\text{WAIT}}$, pakliže přístupová doba k informaci je příliš dlouhá. Základní a rozšířený cyklus M1 je zobrazen na obr.37.

2.5.2 Čtení/zápis dat do paměti

Časové průběhy signálů při provádění čtení/zápisu dat do paměti jsou na obr.38. Tyto cykly jsou podobné cyklu M1, protože při čtení dat z paměti jsou aktivovány signály $\overline{\text{MREQ}}$ a $\overline{\text{RD}}$ stejně jako v cyklu M1. Při zápisu do paměti jsou aktivovány signály $\overline{\text{MREQ}}$ a $\overline{\text{WR}}$. Signál $\overline{\text{MREQ}}$ je použit pro povolení zápisu do paměti a $\overline{\text{WR}}$ definuje stabilní stav dat na DATA BUSu. $\overline{\text{WR}}$ lze připojit na vstupní vodič paměti R/$\overline{\text{W}}$. Cykly mohou být rozšířeny ze základních tří taktů na požadovaný počet vložením TW taktů (viz obr.39). Nulová hodnota signálu $\overline{\text{WAIT}}$ v taktu T2 a v prvním čekacím taktu

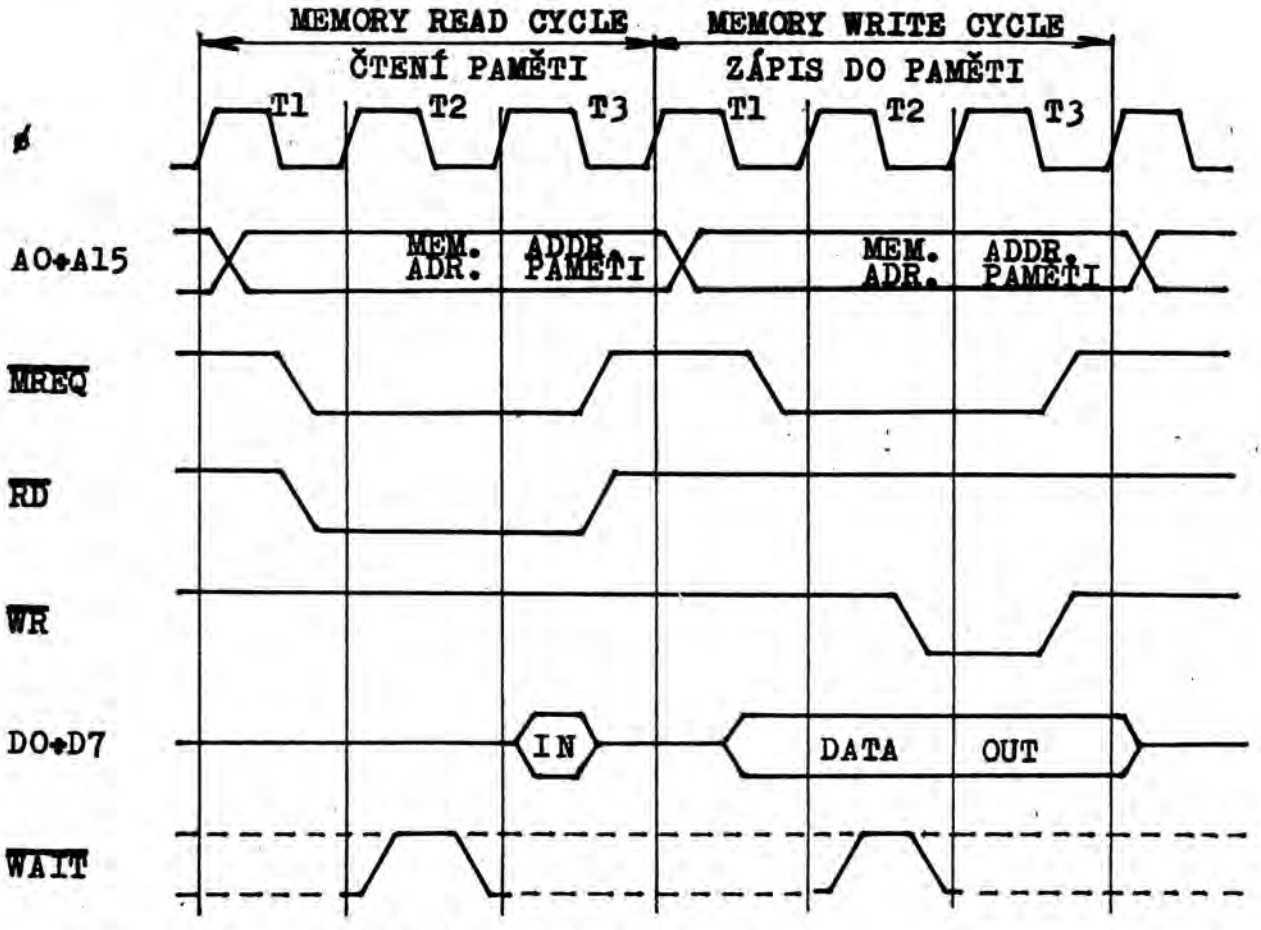


Obr.37 Čtení operačního kódu instrukce

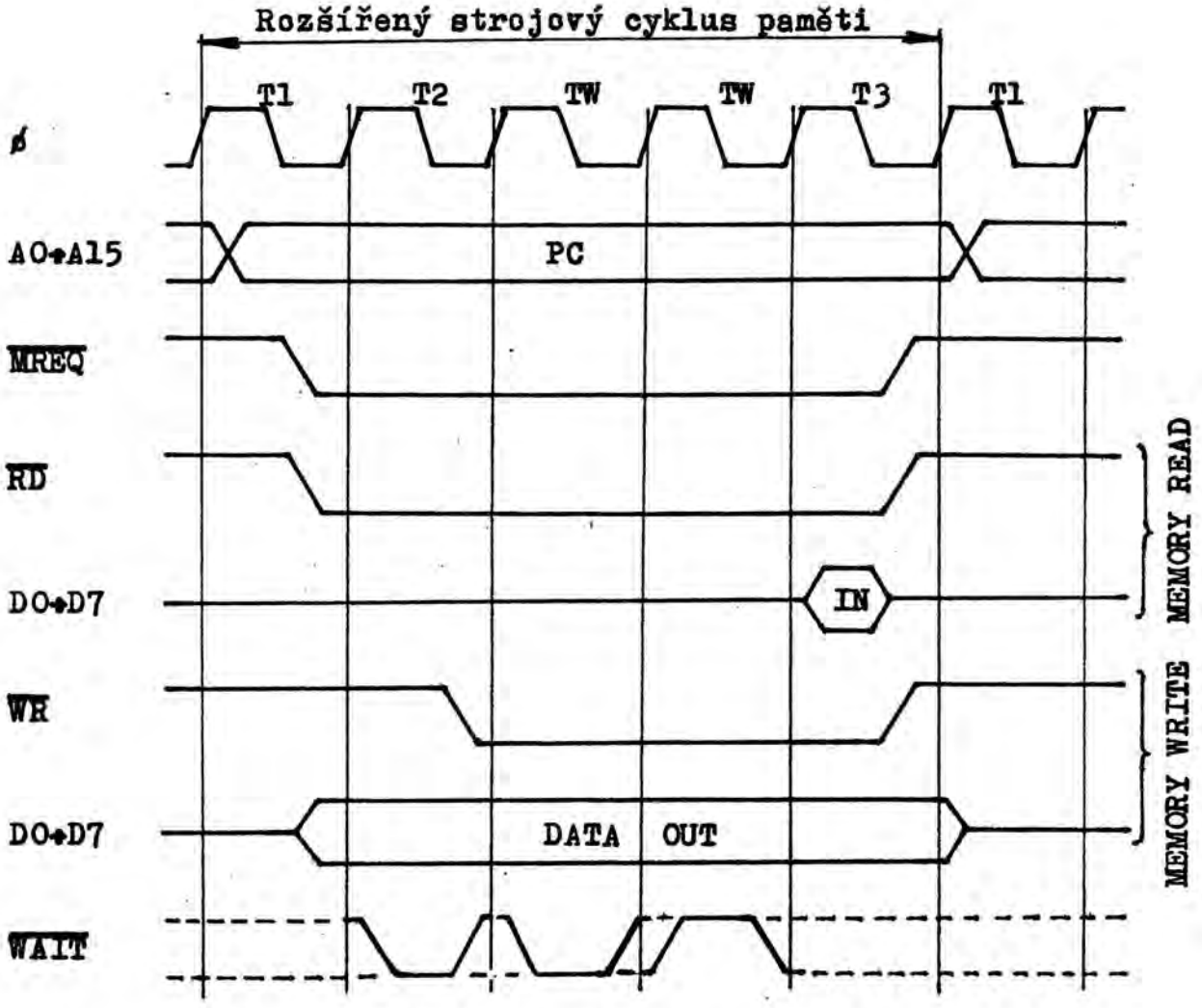
TW zajistí vložení dvou čekacích taktů. V druhém TW taktu je signál WAIT již jedničkový, proto mikroprocesor pokračuje dále v paměťovém cyklu taktem T3.

2.5.3 Čtení/zápis dat do V/V zařízení

Na těchto cyklech je zajímavé, že Z80-CPU automaticky vkládá jeden TW takt. Prodloužení cyklu je nutné, protože jinak by V/V zařízení nemělo dostatek času na aktivování vstupu WAIT procesoru. Z80-CPU testuje úroveň na vstupu WAIT se sestupnou hranou T2 taktu, přičemž signál IORQ, jímž se aktivuje V/V zařízení, je aktivní od nástupní hrany taktu T2. Vložením jednoho TW taktu se vzorkování úrovně na vstupu WAIT posune, čímž se V/V zařízení poskytne dostatek času na vydání požadavku WAIT. Obr. 40 zobrazuje cyklus čtení/zápis dat do V/V zařízení.



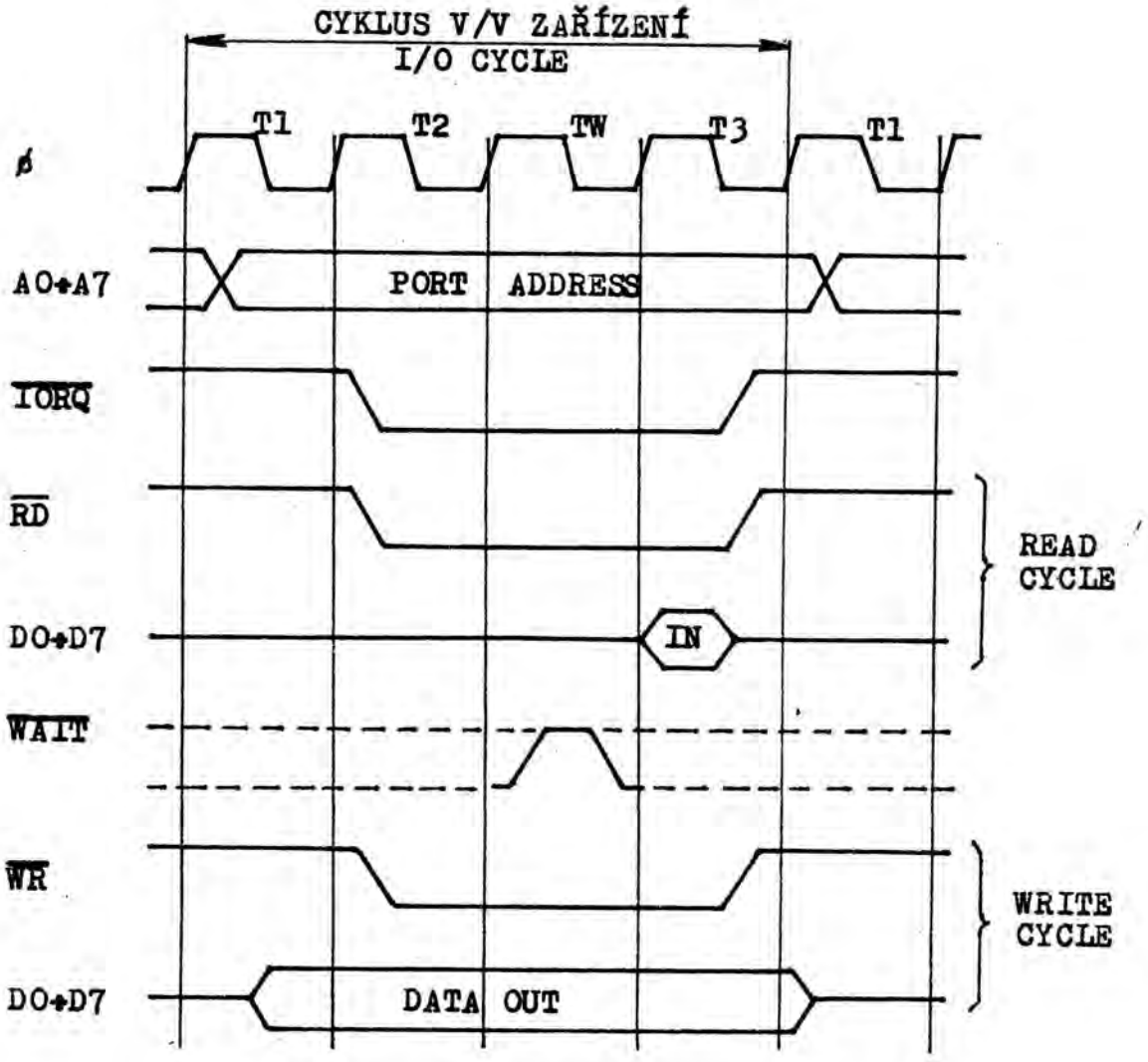
Obr. 38 Čtení/zápis dat do paměti



Obr.39 Rozšířený cyklus paměti

2.5.4 Cyklus požadavku/potvrzení požadavku na převzetí řízení sběrnic

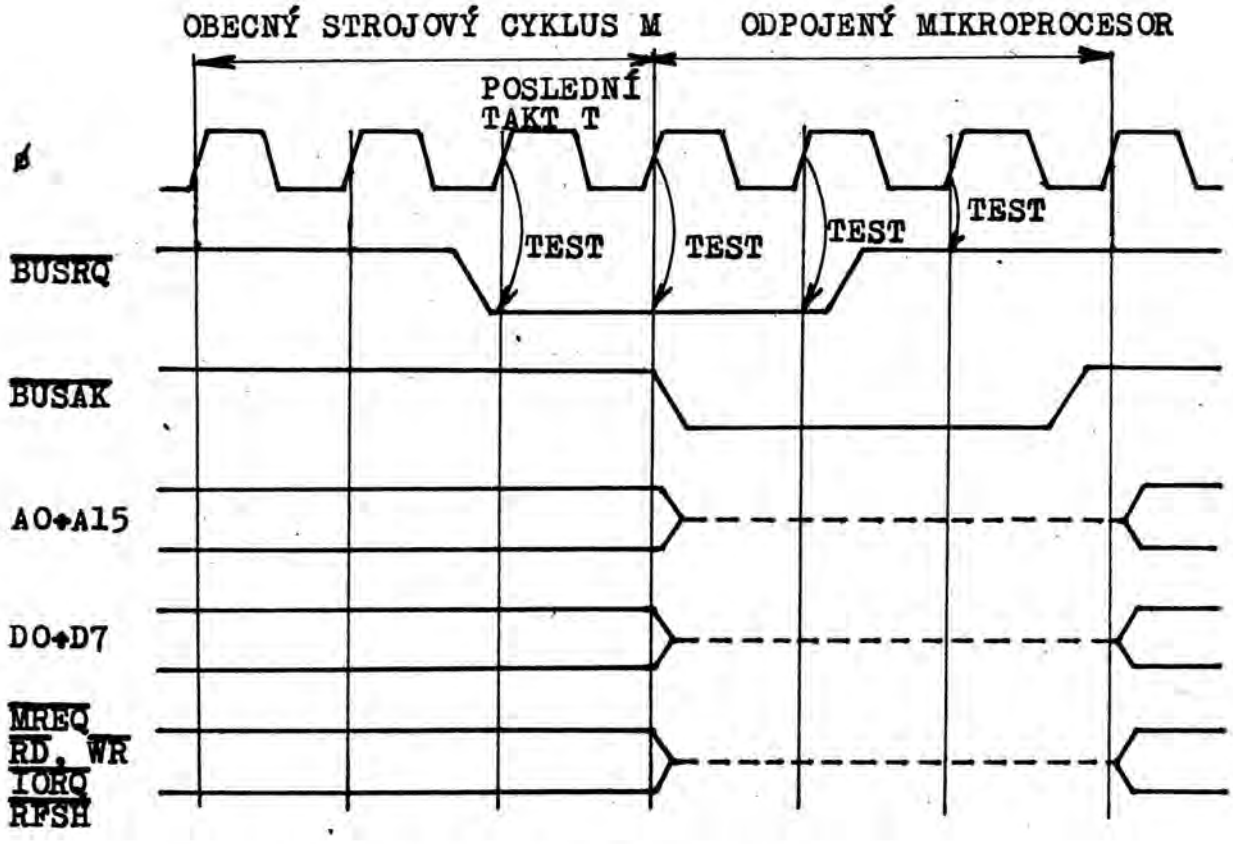
Žádá-li některý z řídících obvodů převzetí řízení sběrnic, aktivuje vstupní signál $\overline{\text{BUSRQ}}$ mikroprocesoru. Úroveň na tomto vstupu je vzorkována vždy v posledním taktu každého strojového cyklu. Pakliže je v tomto taktu vstup $\overline{\text{BUSRQ}}$ na úrovni L, dojde k odpojení Z80-CPU od sběrnic v následujícím taktu a zároveň výstup potvrzení uvolnění sběrnic $\overline{\text{BUSAK}}$ přejde do aktivní úrovně. Adresová, datová a řídicí sběrnice mikroprocesoru se nachází ve vysokoimpedančním stavu. Převzetí řízení sběrnic může trvat libovolně dlouho, ale je třeba si uvědomit, že po dobu, kdy je mikroprocesor odpojen od sběrnic, neprovádí obnovování dat v dynamických pamětech. Časový rozbor je na obr. 41.



Obr. 40 Čtení/zápis dat do V/V zařízení

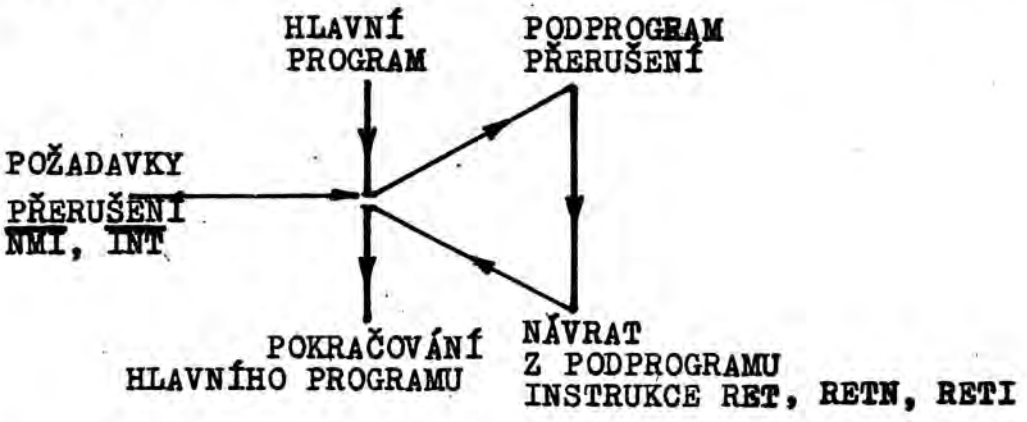
2.5.5 Cyklus požadavku/potvrzení požadavku na přerušení

Na požadavky přerušení jednotka CPU reaguje přerušením provádění právě probíhajícího programu a začíná provádět obslužný program zařízení, které vydalo požadavek na přerušení (INTERRUPT SERVICE ROUTINE). Od strojového cyklu přerušení se vyžaduje co nejrychlejší odezva na požadavek obsluhy, poněvadž v celé řadě případů je tento požadavek podmíněn vznikem havarijních stavů v řízeném procesu. Obecně může být počet zařízení žádajících přerušení značně velký a ne všechny mohou být z hlediska



Obr. 41 Cyklus odpojení sběrnic

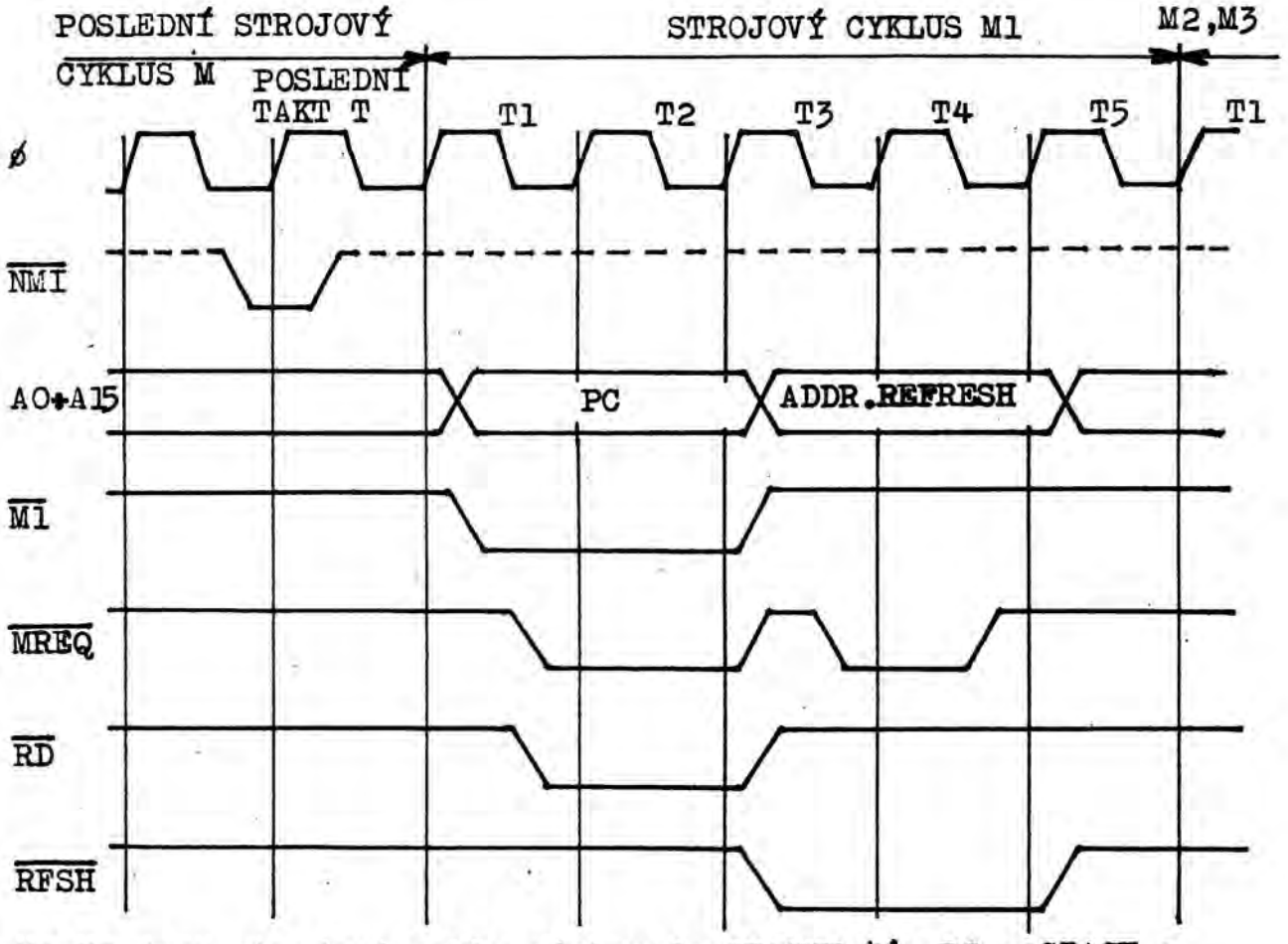
důležitosti obsluhy na stejné úrovni. Pak je nutné, aby struktura mikropočítače umožnila přiřadit prioritu obsluhy jednotlivých přerušujících zařízení. Po skončení podprogramu přerušení procesor pokračuje opět ve zpracování hlavního programu od místa, kde byl přerušen signálem NMI nebo INT. Tento obecný případ zachycuje obr. 42.



Obr. 42 Obecný případ ošetření přerušení

Mikroprocesor zpracovává požadavek nemaskovatelného přerušení ze vstupu $\overline{NMI}$ a požadavek maskovatelného přerušení ze vstupu $\overline{INT}$:

nemaskovatelné přerušení (aktivní signál $\overline{NMI}$) — tímto vstupem se provede přerušení prováděný programu vždy, tzn., že tento požadavek má nejvyšší prioritu. Používá se pro úklid registrů Z80-CPU při výpadku napájení atd. Při tomto přerušení mikroprocesor ignoruje další prováděnou instrukci a provede odskok na adresu 66H, kde je začátek obslužného programu. Časový diagram je na obr. 43. Před fyzickým odskokem na obslužný program



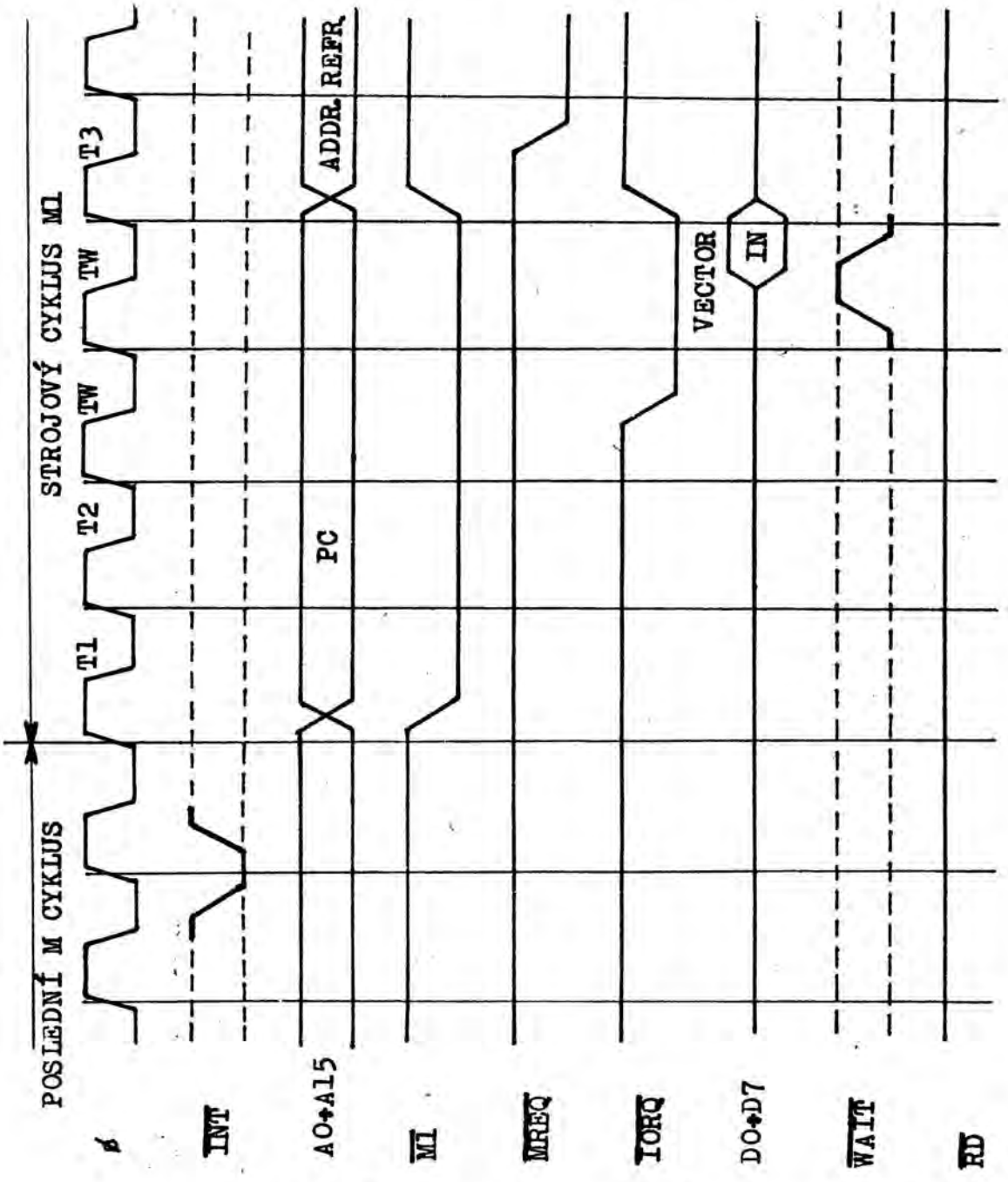
M2,M3 jsou strojové cykly zápisu do STACKU, tj. $PC_H \rightarrow$STACK a $PC_L \rightarrow$STACK.

Obr. 43 Cyklus požadavku a potvrzení nemaskovatelného přerušení

dojde k automatickému uložení obsahu programového čítače do zásobníku.

maskovatelné přerušení (aktivní signál $\overline{INT}$)

časové odbavení maskovatelného přerušení je na obr. 44. Přerušení je programově blokovatelné instrukcí DI (DISABLE ENABLE).



Obr. 44 Cyklus požadavku a potvrzení maskovatelného přerušení

Přerušovací signál je testován Z80-CPU s náběžnou hranou posledního hodinového pulsu na konci každého instrukčního cyklu. Pakliže existuje požadavek na přerušení, je zařazen speciální cyklus M1. Během tohoto cyklu se stává aktivní signál $\overline{\text{IORQ}}$ a indikuje, že periferie,která vyvolala přerušení,má dodat osmibitový vektor na DATA BUS. Automaticky se zařazují dva takty TW, aby vzhledem k prioritnímu řetězci přerušení bylo dostatek času generovat vektor přerušení. Maskovatelné přerušení lze zpracovávat třemi odlišnými způsoby,tj. MODE0,1,2.

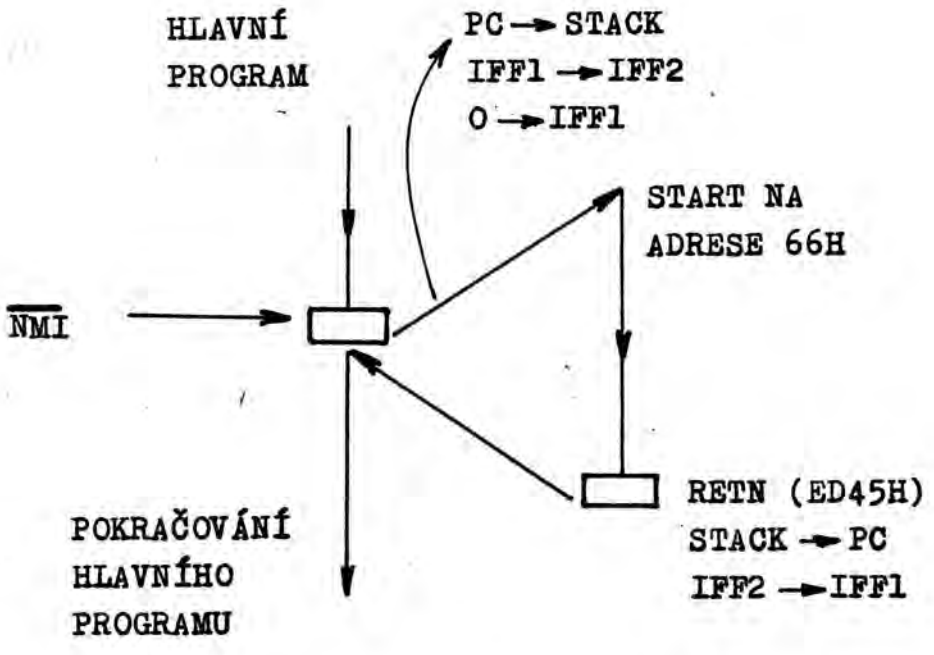
2.6 Přerušovací systém

V rozboru přerušovacího systému je využito poznatků získaných v kap. 2.5.5.

Požadavky přerušení Z80-CPU

a) Nemaskovatelné přerušení (aktivní signál $\overline{\text{NMI}}$) - přerušení nelze maskovat instrukcí DI a má nejvyšší prioritu. Vyšší prioritu než požadavek $\overline{\text{NMI}}$ má jen žádost o DMA přenos,tj. signál $\overline{\text{BUSRQ}}$. Mikroprocesor po příchodu nulového signálu $\overline{\text{NMI}}$ uloží poslední adresu hlavního programu z PC do zásobníku a čítač instrukcí nastaví tvrdě na adresu 66H. Na této adrese začíná první instrukce podprogramu přerušení. Na konci podprogramu je instrukce RETN (RETURN FROM NONMASKABLE INTERRUPT), která vyjme ze zásobníku návratovou adresu a hlavní program pokračuje ve své činnosti právě od této adresy. Průběh nemaskovatelného přerušení zachycuje obr. 45.

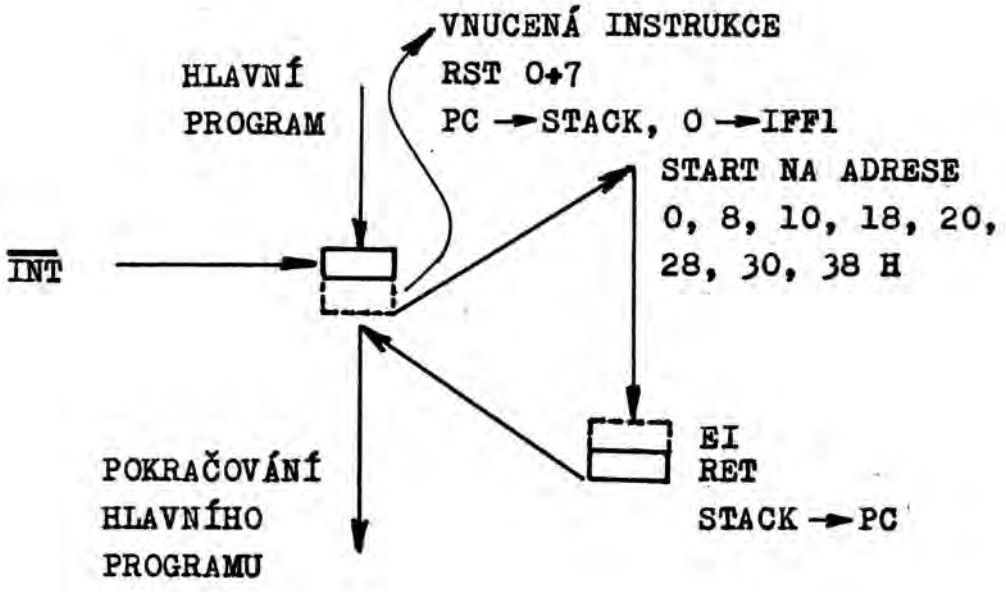
b) Maskovatelné přerušení (aktivní signál $\overline{\text{INT}}$) - maskovatelná přerušení jsou zpracovávána na základě tří programovatelných režimů (MODE) přerušení. Mode 0, 1, 2 se aktivuje instrukcemi IM0, IM1, IM2 (SET INTERRUPT MODE). V případě mode 0 a 1 lze nalézt analogii s odbavením přerušení u I 8080A, kdežto mode 2 zabezpečuje odbavení přerušení zcela novým způsobem.



Obr. 45 Nemaskovatelné přerušení NMI

- Přerušení typu 0 (MODE 0) - pro práci mikroprocesoru v tomto režimu je třeba buď přivést aktivní signál na vstup $\overline{\text{RESET}}$, kdy se mikroprocesor automaticky dostane na přerušovací režim 0, nebo musí být v programu zařazena instrukce IM0 (ED46H). V programu musí být také instrukce EI (ENABLE INTERRUPT), která přerušení povoluje. Nyní po přivedení aktivního signálu $\overline{\text{INT}}$ dojde k přerušení. Obsah čítače instrukcí se vloží do zásobníku a mikroprocesor si v cyklu $\overline{\text{INTA}}$, tj. $\overline{\text{IORQ}}.\overline{\text{MI}}$, vyžádá přes datovou sběrnici instrukci, kterou má provádět jako první v podprogramu přerušení. Tato instrukce je dodána technickým vybavením počítače a jde o instrukci RST (RESTART), která je jednobytová a zabezpečí odskok na možných 8 adres, tj. 0H, 8H, 10H, 18H, 20H, 28H, 30H, 38H. Na těchto adresách jsou starty podprogramů přerušení. Na konci podprogramů je instrukce RET (RETURN), která zabezpečí zpětný návrat poslední adresy ze zásobníku. Tuto adresu vloží do programového čítače PC, a tím dojde

k návratu do hlavního programu. Princip odskoku a návratu do a z podprogramu přerušení zpracovávaného v MODE 0 je na obr. 46.

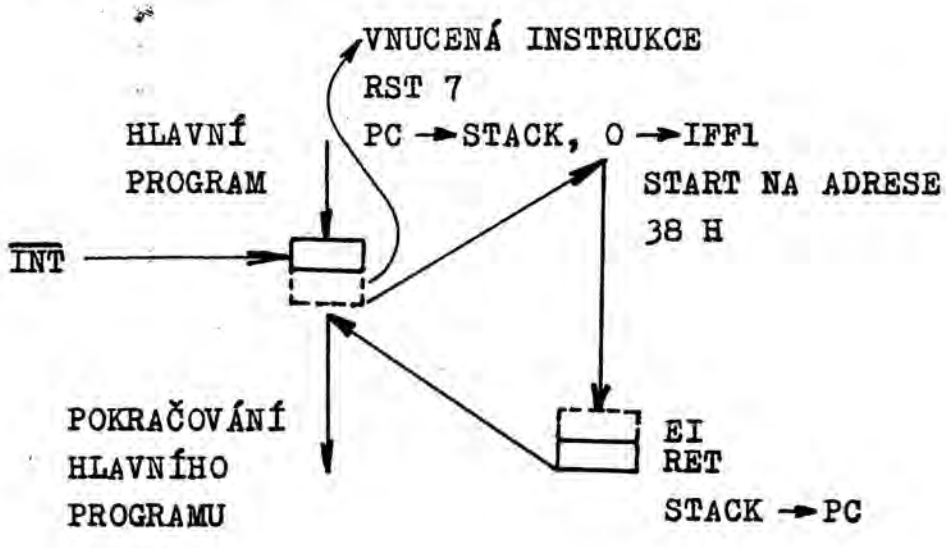


Obr. 46 Přerušení v MODE 0

Přerušovací režim je stejný jako u mikroprocesoru I 8080A. Proto je zapotřebí přídavného hardware, který dodá přerušovací instrukci typu RST. V zapojeních se nejčastěji uplatňuje obvod MH 3214 ve spolupráci s osmibitovým registrem MH 3212, který obsahuje i klopný obvod přerušení.

Závěr přerušovacího podprogramu ošetřuje instrukce EI, která odblokovává zpracování dalších přerušení. Při dřívějším začlenění instrukce EI do přerušovacího podprogramu je třeba předpokládat možnost přerušení zpracovávaného podprogramu novým podprogramem přerušení. Tímto způsobem dochází k vhnízdění podprogramu přerušení.

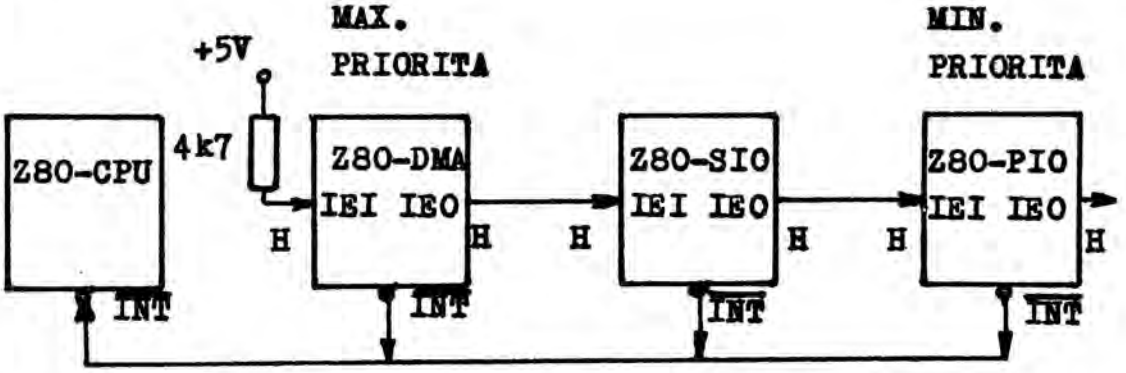
- Přerušení typu 1 (MODE 1) - v režimu 1 Z80-CPU reaguje na požadavek přerušení provedením startu na adrese 38H. Reakce je obdobná reakci na nemaskovatelné přerušení s tím rozdílem, že přerušení můžeme programově ovládat instrukcemi EI a DI. MODE 1 se nastavuje instrukcí IM1 (ED56H). Po příchodu signálu $\overline{INT}$ se nuluje vnitřní klopný obvod přerušení IFF1 a tím je blokováno opakované akceptování přerušení v době zpracování přerušení. Odblokování zpracování přerušení ze vstupu $\overline{INT}$ se provede programově instrukcí EI. Režim 1 nevyžaduje žádný přídavný hardware. Princip odbavení tohoto přerušení je na obr. 47.



Obr. 47 Přerušení v MODE 1

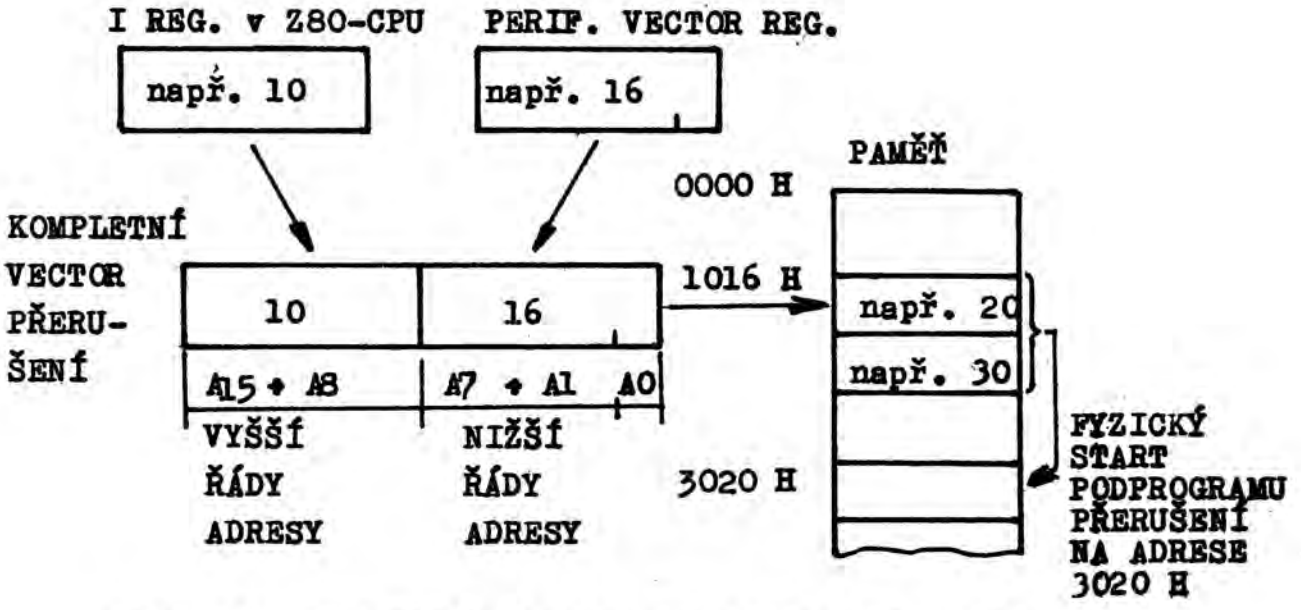
- Přerušení typu 2 (MODE 2) - režim 2 předpokládá použití programovatelných podpůrných obvodů řady Z80 (Z80-PIO, Z80-SIO, Z80-CTC, Z80-DMA). Veškeré tyto obvody umožňují pracovat v režimu 2, kde mohou vytvářet řetězce, které přiřazují každému kanálu konkrétní prioritu vzhledem k poloze obvodu v řetězci. Znázornění řetězce periferních obvodů je na obr. 48. MODE 2 se nastavuje instrukcí IM2 (ED5EH). Jestliže nastalo přerušení, oznamuje to periferní obvod signálem $\overline{INT}$ a předává do Z80-CPU obsah svého vektoru přerušení, tj.

hodnota, z které se vytvoří spolu s registrem I v mikroprocesoru adresa, na které se nachází adresa další (tzv. nepřímé adresování začátku podprogramu přerušení. Ne každý periferní obvod generuje jen jeden vektor přerušení. Např. Z80-PIO má dva porty, a proto generuje dva vektory přerušení. Každý port má svůj podprogram přerušení umístěn na konkrétním místě v paměti.



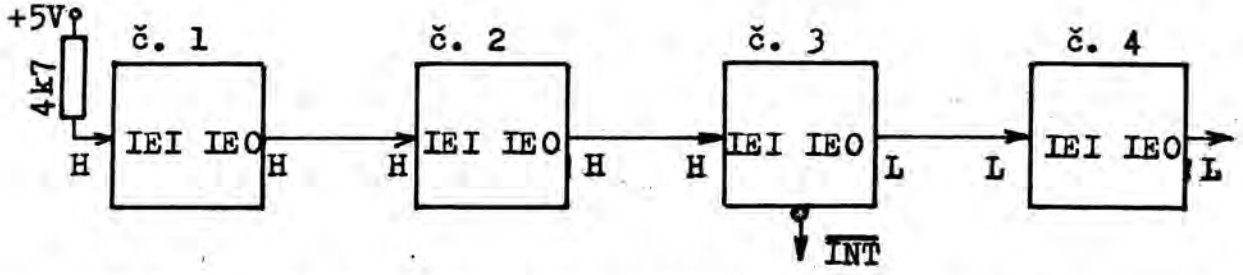
Obr. 48 Prioritní řetězec periferních obvodů pracujících v MODE 2

Připojený obr. 49 ukazuje tvorbu adresy začátku podprogramu přerušení.

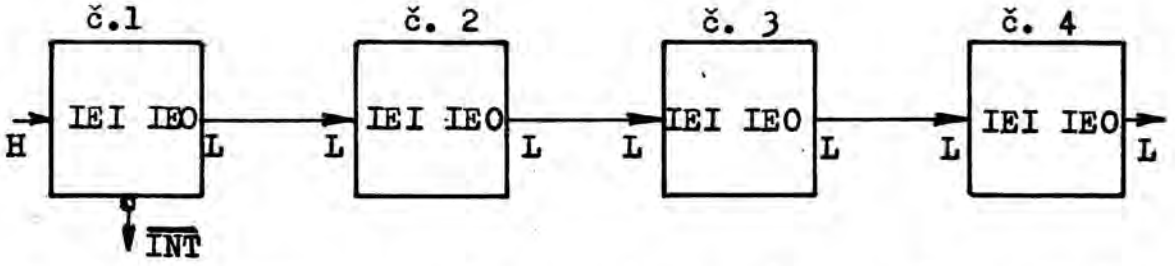


Obr. 49 Vytvoření startovací adresy v MODE 2

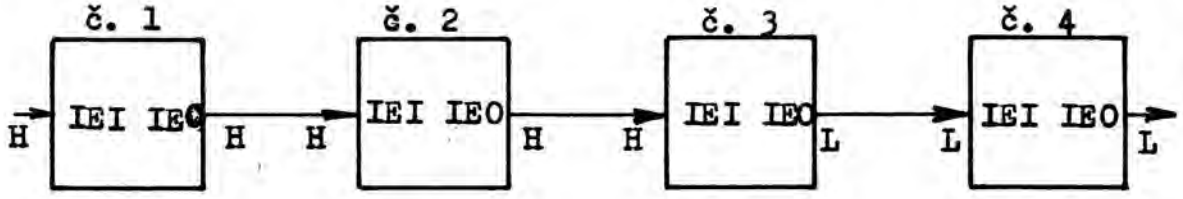
Ve vnitřním registru mikroprocesoru I je umístěna levá polovina adresy. Vektor přerušení, jehož obsah je při přerušení přečten z periferního zařízení,tvoří pravou část adresy. Sloučením obsahu registru I a vektoru přerušení vzniká celková šestnáctibitová adresa. Obsazení registru I se programuje instrukcí LD I,A. Periferní vektory přerušení jsou definovány řídicími slovy.
Periferní zařízení nebo dílčí port periferního zařízení, které žádá o přerušení,poskytne přes DATA BUS svůj přerušovací vektor, z kterého si mikroprocesor vytvoří adresu a nalezne podprogram přerušení.
Obr. 48 ukazoval prioritní řetězec vytvořený z periferních obvodů. Priorita přerušení je zabezpečena zapojením portů za sebou,tj. port 1 → 3. Nejvyšší prioritu má obvod Z80-DMA, nejnižší prioritu má Z80-PIO. Vzájemná priorita je zajišťována vstupem IEI (INTERRUPT ENABLE IN) a výstupem IEO (INTERRUPT ENABLE OUT). V případě stavu vstupního signálu IEI = H může periferní obvod generovat $\overline{INT}$, za stavu IEI = L je přerušení blokováno. Výstup IEO opakuje úroveň, která je na vstupu IEI periferního obvodu. V případě,že obvod signalizuje přerušení $\overline{INT}$, přechází výstup IEO na L. Poněvadž jsou signály IEI a IEO zařetězeny, přechází úroveň L z IEO na IEI dalšího obvodu, kde již blokuje generování $\overline{INT}$. Tato zásada platí i pro ostatní obvody v řetězci. Tím je zajištěno, že obvod s nižší prioritou nebude signalizovat přerušení, pokud neskončí podprogram přerušení obvodu s vyšší prioritou.
Konec podprogramu přerušení je ošetřen instrukcí RETI. Instrukce RETI zabezpečí v mikroprocesoru naplnění čítače instrukcí PC adresou, na které skončil hlavní program před odskokem do podprogramu přerušení. Instrukce RETI má také dopad na periferní obvod, který žádá o přerušení. Před instrukcí RETI byly signály některého obvodu IEI = H a IEO = L, ale po přijmutí RETI nastane IEO = H. Závěrem lze říci, že obvod,signalizující přerušení,má odlišný stav signálů IEI, IEO,a právě takovýto obvod je instrukcí RETI ošetřen.



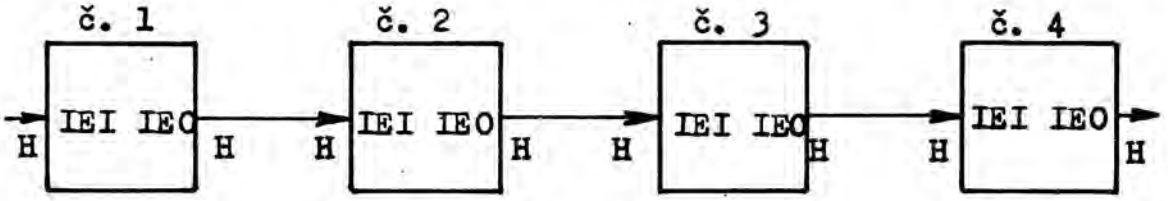
a) obvod č. 3 žádá o přerušení aktivním výstupem $\overline{INT}$



b) obvod č. 1 žádá o přerušení aktivním výstupem $\overline{INT}$ v době zpracování podprogramu přerušení obvodu č. 3



c) konec podprogramu přerušení obvodu č. 1. Instrukce RETI nuluje obvod č. 1, ale obvod č. 3 zůstává aktivní.

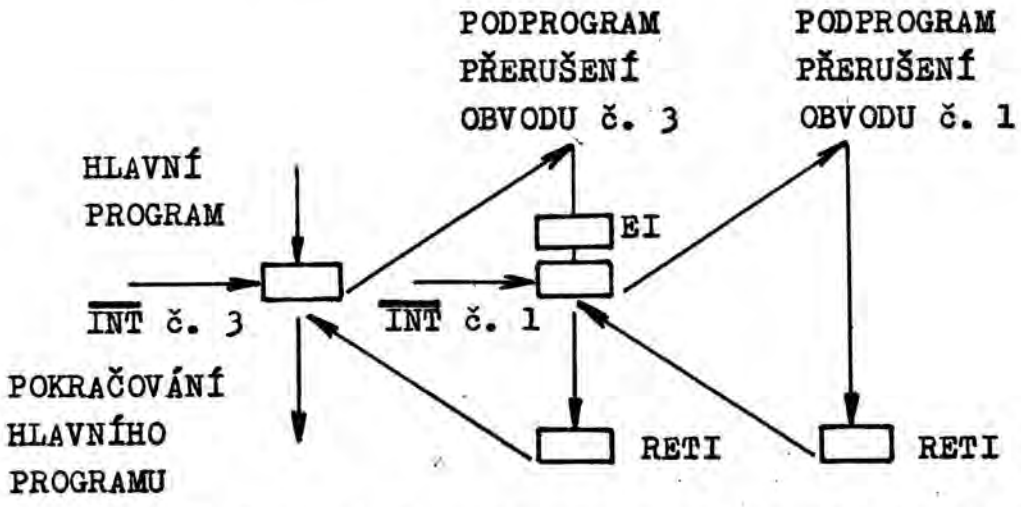


d) instrukce RETI nuluje obvod č. 3 a tím se dostáváme do původního výchozího stavu

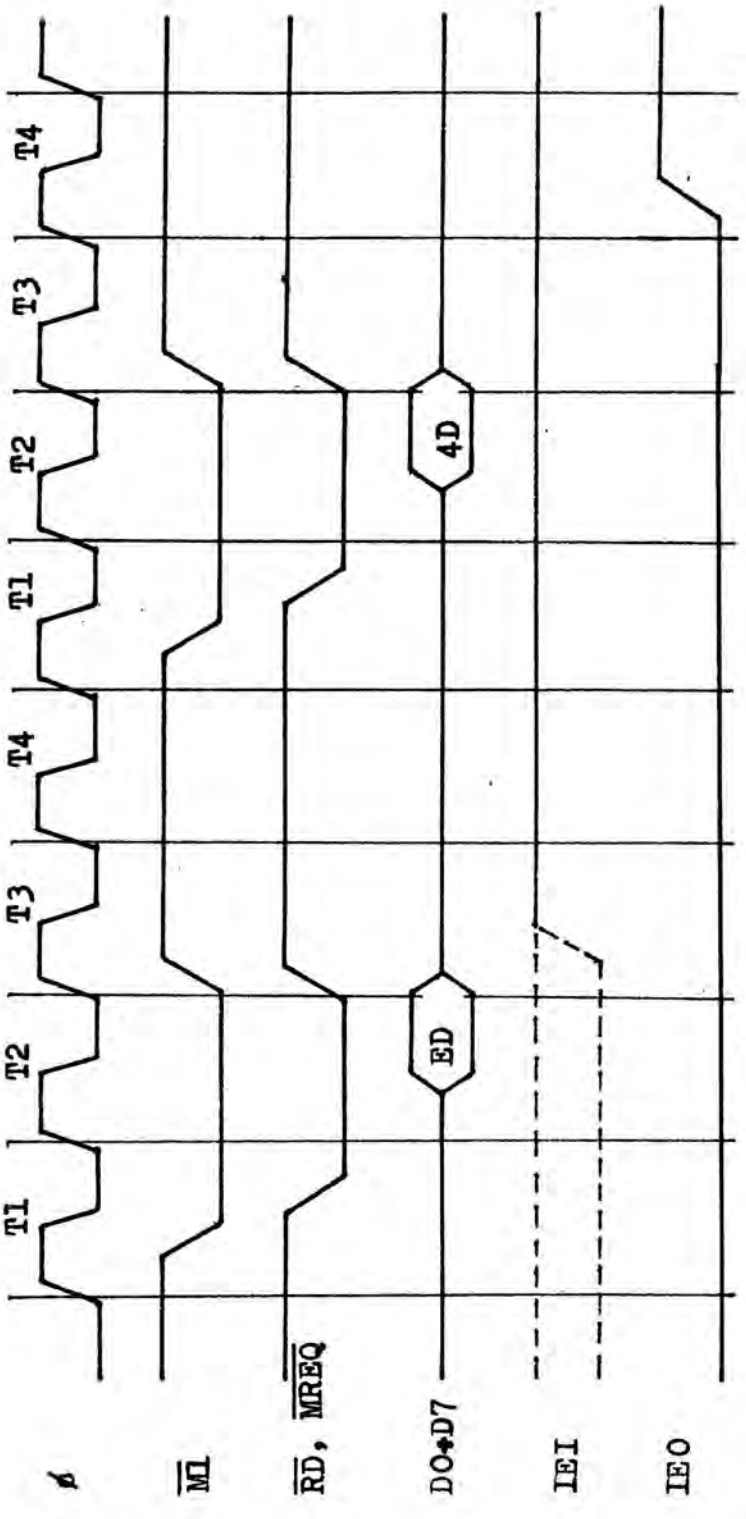
Obr. 50 Postupné odbavení požadavků přerušení od zařízení č. 3 a č. 1

Následuje příklad pro ošetření prioritního řetězce pracujícího v MODE 2. Např. zařízení č. 3 žádá přerušení a přerušení je přijato. Po dobu obsluhy tohoto zařízení dojde k požadavku na přerušení od zařízení č. 1. Poněvadž zařízení č. 1 má vyšší prioritu, je přerušení akceptováno a současně je přerušeno provádění obslužného podprogramu zařízení č. 3. Tento postup nastane za předpokladu, že v podprogramu přerušení zařízení č. 3 byla instrukce EI, která povolila příjem dalšího přerušení,tj. v našem případě od zařízení č. 1.

Na konci obslužného podprogramu zařízení č. 1 je dvoubytová instrukce RETI, která signalizuje zařízení, že jeho obslužný program byl dokončen,tj. dojde k obnovení provádění přerušeného podprogramu pro zařízení č. 3. Po dokončení podprogramu pro zařízení č. 3 se veškeré vstupy a výstupy IEI, IEO uvedou na úroveň H. Postupné odbavování požadavků přerušení od zařízení č. 3 a č. 1 je na obr. 50. Situaci, kdy přerušovací podprogram zařízení č. 3 přerušuje podprogram zařízení č. 1, který má vyšší prioritu, zachycuje obr. 51. Časové průběhy signálů návratu z podprogramu přerušení instrukcí RETI (ED4DH), kterou dochází ke shození signálu IEO, jsou na obr. 52.



Obr. 51 Přerušení hlavního programu obvodem č. 3 a č. 1 (obvod č. 1 má vyšší prioritu než obvod č. 3)



Obr. 52 Návrat z podprogramu přerušení v MODE 2 instrukcí RETI

2.7 Obnovování stavu dynamických pamětí

Pro obnovování informací v paměťových buňkách dynamických pamětí má Z80-CPU zabudován osmibitový REFRESH REGISTER a řídicí signál $\overline{RFSH}$. Běžně používané dynamické paměti pracují s obnovovacím cyklem 2 ms, za který je nutno aktivovat veškeré buňky paměti. Organizace dosud jediné tuzemské dynamické paměti MHB4116 je 128 řádek krát 128 sloupců. Každý obnovovací cyklus má dopad na všech 128 sloupců řádku, a proto na obnovu celé paměti stačí 128 cyklů obnovení. Tomu vyhovuje vnitřní aparát Z80-CPU. REFRESH REGISTER má celkem 8 bitů, z nichž prvních 7, tj. R0 ÷ R6, představuje čítač, jehož obsah se s každým cyklem $\overline{M1}$ inkrementuje a přivádí na adresovou sběrnici A0 ÷ A6. Veškerých 128 kombinací v REFRESH REGISTRu představuje adresy řádků dynamických pamětí, které se obnovují. REFRESH REGISTER je nulován signálem $\overline{RESET}$ a pracují s ním instrukce LD R,A a LD A,R (zápisová a čtecí instrukce). Nejvyšší bit R7 je inkrementací registru neovlivněn, zůstává v původním naprogramovaném stavu instrukcí LD R,A. V kapitole 2.5.1 je časový diagram instrukčního cyklu M1, z kterého je patrné, že na adresové sběrnici se nejprve objeví obsah programového čítače PC. Současně je signál $\overline{MREQ}$, $\overline{M1}$, $\overline{RD}$ aktivován a přečtená instrukce se dostává přes datovou sběrnici D0 ÷ D7 do instrukčního registru. Teprve potom se na adresovou sběrnici připojí REFRESH REGISTER a za součinnosti řídicích signálů $\overline{RFSH}$ a $\overline{MREQ}$ je dynamická paměť na svém jednom řádku obnovena.

2.8 Spolupráce s paměťovými a periferními obvody

Procesní jednotka Z80-CPU umožňuje pomocí zabudované řídicí sběrnice rozlišit adresaci periferie a paměti. Tuto diferenciaci řeší signály $\overline{MREQ}$ a $\overline{IORQ}$. Paměťové instrukce pracují při svém provádění se signálem $\overline{MREQ}$ a adresní sběrnicí, kde figuruje šestnáctibitová adresa, umožňující lokalizování jakéhokoliv místa v paměťové matici 64 kadres.

Instrukce vstupu/výstupu využívají při svém provedení řídicí vodič $\overline{IORQ}$ spolu s osmibitovou adresou vystupující na nižších

osmi bitech adresní sběrnice. Instrukce typu IN a OUT umožňují spolupracovat s 256 vstupními a 256 výstupními zařízeními.

V případě většího počtu periferních zařízení můžeme využít mapovacího způsobu adresace (MEMORY MAPPED I/O) a tím i bohatého sortimentu paměťových instrukcí.

Z časových průběhů v kap. 2.5 plyne, že ve strojovém cyklu M1 musí paměť programu dodat operační kód instrukce na sběrnici v době kratší než 450 ns od okamžiku platnosti signálu $\overline{\text{MREQ}}$. Kdyby tento požadavek polovodičová paměť nesplňovala, je možné strojový cyklus M1 prodloužit o čekací takty TW.

Při čtení dat jsou časové relace již příznivější, protože informace na datové sběrnici je snímána o půl periody hodin později, z čehož vyplývá požadavek na dobu přístupu kratší než 640 ns. Při komunikaci s periferními zařízeními jsou požadavky na rychlost odezvy ještě volnější, poněvadž data musí být na sběrnici dodána/sejmuta v nejhorším případě za 840 ns. Tyto časy jsou podmíněny automatickým vložením jednoho čekacího taktu TW vsunutého za takt T2.

2.9 DMA přenos

Pro zajištění přenosů mezi dvěma oblastmi mikropočítačového systému v režimu DMA je mikroprocesor Z80-CPU vybaven řídicími obvody, jejichž činnost je ovládána vstupním signálem $\overline{\text{BUSRQ}}$. Přijetím signálu $\overline{\text{BUSRQ}}$ dokončí mikroprocesor právě probíhající strojový cyklus a pak uvede veškeré třístavové sběrnice do vysokoimpedančního stavu. Zařízení, žádající přístup na sběrnici, informuje mikroprocesor o této skutečnosti aktivací výstupu $\overline{\text{BUSAK}}$. Obvod přímého přístupu do paměti Z80-DMA může disponovat systémovou sběrnicí po neomezenou dobu. Jestliže jsou v systému aplikovány také dynamické paměti RAM, musí převzít Z80-DMA i řízení zotavovacího čtení. Obvod Z80-DMA je určen pro rychlý přenos dat mezi pamětí a pamětí, pamětí a periferií, periferií a periferií. Protože obvod dovede generovat veškeré signály generované mikroprocesorem, je schopen převzít řízení sběrnice a tím vyloučit z přenosu dat mikroprocesor, což podstatně zkrátí dobu přemístění dat ze zdrojové do cílové adresy. Řadič Z80-DMA

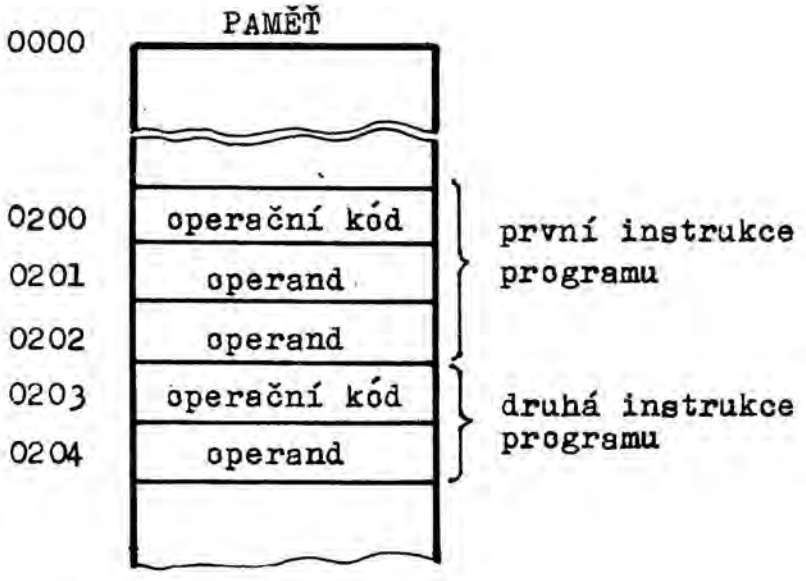
je programově řízená jednokanálová dvoubránová jednotka, která generuje adresy, časové sekvence a řídicí signály pro DMA přenosy dat. Dvě šestnáctibitové adresní brány umožňují generování adresy zdrojového a cílového místa přenosu. DMA přenos je veden přes osmibitový datový vyrovnávací registr umístěný přímo v řadiči Z80-DMA.

3. INSTRUKČNÍ SOUBOR

Firma Zilog uvádí pro svůj mikropočítačový systém Z-80 158 strojových instrukcí. Z toho je 78 instrukcí Intel 8080, které se liší od Zilogu pouze symbolickým kódem. Zbylých 80 instrukcí je nových. Tato programová slučitelnost pomohla k rychlému rozšíření řady Z-80, protože umožnila uživateli použít původní programové vybavení Intel při aplikování stavebnice Z-80. Instrukce Z-80 jsou 1 ÷ 4 bytové s 1 ÷ 2 bytovým operačním znakem. Zbylé byty tvoří operandovou část instrukce. Operandy mohou vyjadřovat:

- adresu paměti, jedná-li se o instrukci, která se odvolává na data uložená v paměti nebo jedná-li se o instrukci skoku
- adresu paměti, jedná-li se o instrukci skoku
- relativní vzdálenost při relativním způsobu adresování
- data, jde-li o instrukci přenosu dat

Program sestavený z jednotlivých instrukcí je uložen v paměti, ze které je postupně byte po byte načítán do registrů mikroprocesoru. Po načtení celé instrukce následuje její provedení. Pak je načtena a provedena další instrukce programu. Program v paměti je umístěn vzestupně, jak znázorňuje obr. 53.



Obr. 53 Umístění programu v paměti

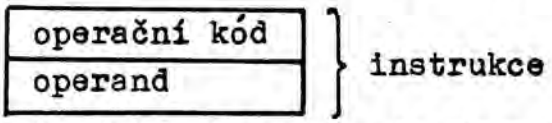
Doba načtení a provedení instrukce je definována počtem taktů T neboli period hodin. Počet těchto taktů je znám pro každou instrukci, a tudíž lze stanovit délku provádění programu. Minimální takt pro mikroprocesor Z80-CPU je 400 ns. Rychlejší verse Z80A-CPU a Z80B-CPU mají tuto dobu kratší.

3.1 Způsoby adresování

Paměť, ale i vstup/výstupní zařízení je zapotřebí pro získání dat adresovat. Z80-CPU oproti I 8080A umožňuje tzv. relativní a indexové adresování. Pro přehlednost jsou zopakovány i ostatní způsoby adresování.

3.1.1 Adresování s přímým operandem

Při adresování s přímým operandem je operandová část instrukce přímo hodnotou, kterou bude instrukce zpracovávat. Hodnota operandu má délku osm nebo šestnáct bitů.

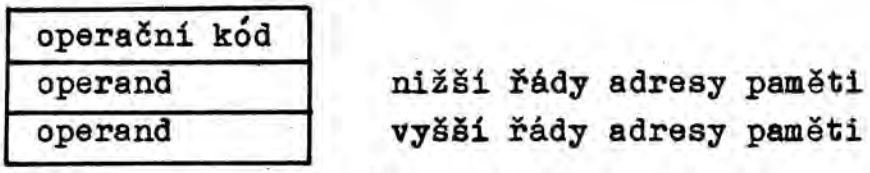


Příklad: Uložení konstanty 05 do akumulátoru:

```
LD A, 05
```

3.1.2 Přímé adresování

Instrukce pracující s přímým adresováním využívají šestnáctibitovou hodnotu obsaženou v instrukci jako adresu paměti. U některých instrukcí vstupu/výstupu je adresa periferie vyjádřena osmibitovou hodnotou v instrukci.



Příklad: Uložení hodnoty paměťového místa (nn) do akumulátoru

```
LD A,(nn)
```

3.1.3 Indexové adresování

Při tomto typu adresování je za operačním kódem uveden datový byte, tzv. posunutí (DISPLACEMENT), které se sčítá s obsahem jednoho ze dvou index registrů, a tím vzniká konečná adresa.

operační kód
operační kód
operand
operand

doplněk d - posunutí
konstanta n

Příklad: Přesun konstanty n na adresu IX + d, která je dána součtem index registru a doplňku d:

LD (IX + d),n

3.1.4 Relativní adresování

Relativní adresování je obdobou indexového adresování s tím rozdílem, že nepracuje s indexovými registry IX, IY, ale s čítačem instrukcí PC. Konečná adresa vzniká součtem obsahu programového čítače PC a jednobytového operandu obsaženého v instrukci.

operační kód
operand

doplněk d - odskok

Příklad: Relativní odskok na základě obsahu operandové části instrukce:

JR d PC ← PC + d

3.1.5 Adresování nulté stránky

Jednobytová instrukce typu RST umožňuje volat jednu z osmi adres na začátku paměťové matice.

operační kód

$11b_2b_1b_0111$

Příklad: Start podprogramu přerušení na adrese 38H:

RST7

Operační kód instrukce RST7 je FF tj. 11111111, po nahrání do mikroprocesoru je tento profil modifikován na profil 00111000 a zanesen na nižší řády adresní sběrnice, tzn. start podprogramu přerušení na adrese 38H.

3.1.6 Adresování registru

V operačním kódu instrukce jsou volitelné bity, které lokalizují registr používaný v instrukci.

operační kód	01 r r´

Příklad: Přesun obsahu registru r´ do registru r. Hodnoty registrů r´, r jsou definovány tříbitovými kódy ve vlastním operačním kódu instrukce.

LD r,r´ $r \leftarrow r'$

3.1.7 Nepřímé adresování šestnáctibitovým registrem

Šestnáctibitová adresa se nachází v některém z párových registrů mikroprocesoru. Instrukce neobsahuje adresu, ale lokalizuje párový registr, v kterém se teprve nachází adresa paměti. Tohoto způsobu adresování se používá v instrukcích blokových přenosů.

3.1.8 Adresování bitu v registru

Operační kód instrukce definuje nejen registr, ale i bit tohoto registru,s kterým bude instrukce pracovat.

operační kód	11001011
operační kód	11 b r

Příklad: Nastavení konkrétního bitu registru:

SET b,r $r_b \leftarrow 1$

3.2 Indikátory stavů

Osmibitový indikátorový registr F (FLAG) poskytuje informace o stavu Z80-CPU a posledně probíhající instrukci. Některými instrukcemi je indikátorový registr plněn nebo částečně plněn, jiné instrukce jeho obsahu využívají jako např. instrukce podmíněných skoků. Obsazení indikátorového registru je na obr. 54.

7	6	5	4	3	2	1	0
S	Z	X	H	X	P/V	N	C

Obr. 54 Obsazení indikátorového registru F

Přehled symbolů:

C	indikátor přenosu (CARRY)
N	indikátor sčítání/odčítání (NEGATIVE)
P/V	indikátor parity/přetečení (PARITY/OVERFLOW)
H	indikátor polovičního přenosu (HALF CARRY)
Z	indikátor nuly (ZERO)
S	indikátor znaménka (SIGN)
X	neobsazená pozice

3.2.1 Indikátor přenosu - bit C (CARRY)

Indikátor přenosu definuje vznik přenosu ze znaménkového bitu, tj. bitu A7 akumulátoru při aritmetických operacích. Dojde-li k přenosu, je nastaven bit C = 1. Rovněž je indikátor C setován instrukcí DAA za předpokladu, že jsou splněny podmínky pro převod do BCD kódu. Logickými operacemi OR, XOR a AND je tento bit nulován. Komplement bitu C provádí instrukce CCF, setování indikátoru C řeší instrukce SCF.

3.2.2 Indikátor sčítání/odčítání - bit N (NEGATIVE)

Indikátor je využíván pouze instrukcí DAA. Pakliže byla předcházející instrukce sčítání, je N = 0, pakliže byla předcházející instrukce odčítání, je N = 1.

3.2.3 Indikátor parity/přetečení - bit P/V (PARITY/OVERFLOW)

Bit P/V má dva významy. Při aritmetických operacích dochází k nastavení bitu, jestliže výsledek aritmetické operace je větší než kapacita registru, v kterém se provádí aritmetická operace. Při logických operacích a rotacích bit zaznamenává paritu výsledku. V případě, že výsledek má sudý počet jedničkových bitů, je bit nastaven do P/V = 1, v opačném případě, tj. má-li výsledek lichý počet jedničkových bitů, je bit P/V = 0.

3.2.4 Indikátor polovičního přenosu - bit H (HALF CARRY)

Tento indikátorový bit využívá při svém provádění instrukce DAA. Indikuje přenos, který nastal mezi 3 a 4 bitem při provádění aritmetických operací.

3.2.5 Indikátor nuly - bit Z (ZERO)

Při provádění instrukcí srovnávání se bit Z nastavuje při shodě srovnávaných veličin. Osmibitové aritmetické a logické operace nastavují indikátor Z = 1, jestliže je výsledek dané operace v akumulačním registru nulový. Při nenulovém výsledku je bit nulován, tj. Z = 0.

3.2.6 Indikátor znaménka - bit S (SIGN)

Znaménkový bit se nastaví, tj. S = 1, jestliže je výsledek provedené operace záporný. Význam znaménka má i nejvyšší bit akumulátoru A7.

3.3 Tabulky instrukcí

V tabulce 2 je plný výčet instrukcí souboru Z80-CPU. V úvodu tabulkových přehledů je vysvětlena použitá symbolika.

Tabulka 2 informuje o:
- mnemonickém tvaru instrukce
- druhu operace vykonaném při provedení instrukce
- obsazení indikátorového registru
- strojovém kódu 1 ÷ 4 bytových instrukcí
- délce instrukce, tj. počtu bytů
- počtu strojových cyklů nutných na načtení a provedení instrukce
- celkovém počtu taktů, za které je celá instrukce realizována
- ekvivalentní instrukci v souboru I 8080A

Rozbor jednotlivých typů instrukcí je uveden v následujících kapitolách.

3.3.1 Instrukce přesunu osmibitových dat (LOAD)

Oproti I 8080A jsou zde nové instrukce, zabezpečující přesuny mezi stradačem A a registrem I nebo R. V jazyku symbolických adres jsou jednotně označeny symbolem LD. I 8080A používal i symbolů MVI a MOV. Tato skupina instrukcí je rozšířena o přesuny s indexovou adresací.

3.3.2 Instrukce přesunu šestnáctibitových dat (LOAD)

Instrukce jsou určeny k plnění nebo ukládání obsahu vnitřních šestnáctibitových registrů nebo registrových párů BC, DE a HL.

3.3.3 Instrukce výměny (EXCHANGE)

Jedná se o zcela nové instrukce, které zajišťují vzájemnou výměnu obsahů hlavního a vedlejšího pole vnitřních registrů Z80-CPU. Instrukce EX AF,AF´ provádí vzájemnou výměnu akumulátorů a indikátorových registrů. Instrukce EXX pak výměnu obsahů zbylých pracovních registrů,tj. BC, DE, HL a BC´, DE´, HL´.

3.3.4 Aritmetické instrukce pro osmibitová data

Skupina rozšiřuje odpovídající instrukce mikroprocesoru I 8080A. Výjimečná je pouze instrukce pro vytvoření dekadického výsledku v akumulátoru po provedené operaci sčítání nebo odčítání. Instrukce DAA (DECIMAL ADJUST ACCUMULATOR) je rozšířena díky zavedení indikátoru N (NEGATIVE) i na výsledky rozdílových operací. Všechny instrukce této skupiny nastavují indikátor P/V na úroveň H nikoliv podle parity výsledku, ale podle aritmetického přeplnění.

3.3.5 Logické instrukce

Logické instrukce provádí logický součet, součin operandů, exclusive OR, inkrementaci a dekrementaci registrů. Logické operace jsou obohaceny o možnost indexového adresování.

3.3.6 Instrukce skoků (JUMP)

Běžné provádění programů je vzestupně po jednotlivých instrukcích. Pro porušení této posloupnosti je nutné použít skokovou instrukci, která vloží do programového čítače PC adresu instrukce, na které bude program pokračovat. Skokové instrukce jsou také nepodmíněné, kdy se skok provede v každém případě bez vlivu indikátorového registru nebo podmíněné, kdy se skok provede při splnění některé z podmínek. Před provedením skoku se testují

čtyři indikátorové bity S, Z, C, P, z kterých je možné vygenerovat osm větvících podmínek. Když je podmínka splněna, dojde k odskoku, není-li splněna, pokračuje se následující instrukcí. Mikroprocesor Z80-CPU provádí také relativní skoky, které mohou být opět nepodmíněné nebo podmíněné. Relativní skok JR (JUMP RELATIVE) je dlouhý dva byty. Instrukce pracuje tak, že přičte obsah druhého bytu k čítači instrukcí PC.

3.3.7 Instrukce rotací a posuvů

Zabezpečují posuv bitů registru nebo paměťového místa vpravo nebo vlevo s možností posuvu přes příznakový bit C. Instrukce souboru I 8080 byly rozšířeny u Z80-CPU o možnost uskutečňovat tyto operace s daty v registrech mikroprocesoru, ale i hlavní paměti adresované dvojicí registrů HL nebo relativně za pomoci indexových registrů IX, IY. Navíc byly doplněny o aritmetické a logické posuvy a rotace čtveřic bitů v šestnáctibitovém slově, jehož jedna polovina je ve střadači a druhá v hlavní paměti adresované registry HL.

3.3.8 Volání podprogramu

Těmito instrukcemi je možné opustit hlavní program a provést vyvolání podprogramu, případně podprogramu přerušení. Volání podprogramu řeší instrukce CALL nebo RST. Instrukce CALL pracuje také podmíněně na základě stavu indikátorového registru F.

3.3.9 Návrat z podprogramu

Mikroprocesor Z80-CPU je rozšířen o dvě instrukce nepodmíněného návratu:

- RETI (RETURN FROM INTERRUPT), jejíž funkce je zcela stejná jako u základní instrukce RET, tj. provede přesun obsahu zásobníku do programového čítače PC, avšak programovatelné podpůrné obvody stavebnice Z-80 jsou schopny ji identifikovat na datové sběrnici v cyklu M1, zrušit přijatý požadavek na přerušení a umožnit tak generaci případných dalších požadavků. Prakticky dochází ke shození signálu IEO na obvodě, který odbavil přerušení.

- RETN (RETURN FROM NON-MASKABLE INTERRUPT), která mimo typické návratové činnosti ještě provádí obnovení stavu klopného obvodu IFF1 (INTERRUPT FLIP-FLOP) z obvodu IFF2, a tím umožňuje řešit případné další přerušení. V době odbavení nemaskovatelného přerušení byl klopný obvod IFF1 přesunut do klopného obvodu IFF2 a vzápětí IFF1 nulován, aby nebylo možné nemaskovatelné přerušení opět přerušit.

3.3.10 Instrukce vstupu a výstupu

Dvanáct instrukcí vstupu/výstupu umožňuje přenos dat mezi periferními zařízeními adresovanými registrem C a zápisníkovými registry. Další způsob přenosu je mezi periferií adresovanou registrem C a hlavní pamětí adresovanou pomocí registru HL. Tento typ přenosu probíhá se současnou inkrementací nebo dekrementací HL a dekrementací čítače přenosové délky, který je realizován v registru B. Přenosy lze vést jednotlivě nebo s opakováním až do vynulování registru B. Uvažované instrukce umožňují přesuny bloků maximální délky 256 byte.

3.3.11 Instrukce blokových přenosů (BLOCK TRANSFER)

Instrukce blokových přenosů umožňují za pomocí jedné instrukce přesun bloku z nadefinovaného paměťového místa na jiné paměťové místo. Např. instrukce LDI (LOAD AND INCREMENT) přenese jednu slabiku dat z adresovaného místa paměti (adresování provádí párový registr HL) na adresu určenou registry DE. Instrukce potom inkrementuje oba adresní ukazatele a dekrementuje šestnáctibitový čítač v registrovém páru BC. Instrukce LDIR (LOAD INCREMENT AND REPEAT) cyklicky opakuje tuto operaci až do vynulování čítače BC. Instrukce LDD (LOAD AND DECREMENT) a instrukce LDDR (LOAD DECREMENT AND REPEAD) mají funkci obdobnou jako předcházející instrukce s tím rozdílem, že adresové ukazatele HL a DE jsou dekrementovány. Trvání instrukčního cyklu LDIR a LDDR může být až 1,4 s, proto je důležité, že k přerušení může docházet i při jejich provádění. Tyto instrukce jsou řešeny jako fiktivní programové smyčky. Při vzniku požadavku přerušení se při těchto instrukcích ukládá do zásobníku adresa téže instrukce a po návratu

z podprogramu přerušení se instrukce dokončí. Podprogram přerušení však nesmí změnit obsahy používaných registrů v instrukcích LDIR a LDDR.

3.3.12 Instrukce vyhledání v blocích dat (COMPARE)

U těchto čtyř instrukcí, blízkých instrukcím blokových přenosů, dochází k vyhledání shodného obsahu paměťového místa s obsahem střadače A. Paměťové místo je adresováno párovým registrem HL. Dílčí porovnání řeší instrukce CPI a CPD, opakované porovnání umožňuje instrukce CPIR, CPDR.

Instrukce CPI (COMPARE INCREMENT) slouží k prohledávání bloku dat, jehož počáteční adresa je v registru HL, délka bloku pak v registru BC. Bitová kombinace, která je porovnávána s každým bytem v bloku dat je uložena ve střadači A. Instrukce porovná obsah střadače A s obsahem paměťového místa odečtením obsahů registrů. Zároveň zvětší obsah registru HL a zmenší obsah registru BC o 1. V případě, že dojde ke komparaci, nastaví se podmínkový bit Z = 1 . V případě, že byl prohledán celý blok (BC = 0), je podmínkový bit nastaven P/V = 0. Bit P/V je indikátorem konce prohledávání bloku dat.

Instrukce CPIR (COMPARE INCREMENT REPEAD) provádí opakovaně instrukci CPI do té doby, dokud není nalezen konec prohledávaného bloku (BC = 0) nebo dokud se nenarazí na profil dat totožný s obsahem střadače A.

Instrukce CPD (COMPARE DECREMENT) a instrukce CPDR (COMPARE DECREMENT REPEAD) jsou řešeny obdobně jako výše uvedené instrukce pouze s tím rozdílem, že dochází k dekrementaci registru HL.

3.3.13 Instrukce bitových operací

Instrukce bitových operací nastavují, nulují nebo testují jednotlivé bity osmibitových operandů v zápisníkových registrech nebo v hlavní paměti, adresované párovým registrem HL nebo indexovými registry.

3.3.14 Řídicí instrukce

Skupina řídicích instrukcí je rozšířena o tři nové pro nastavení režimu přerušení IM0, IM1, IM2 (SET INTERRUPT MODE). Z původních instrukcí společných řadě I 8080A je využita instrukce povolení přerušení EI (ENABLE INTERRUPT), blokování přerušení DI (DISABLE INTERRUPT), které se týkají maskovatelného přerušení. Řídicí instrukce NOP (NO OPERATION) neprovádí žádnou činnost na registrech ani sběrnici. Instrukce HALT uvádí mikroprocesor do dynamického stopu, vo kterém dochází pouze k obnově stavu dynamických pamětí na základě vřazené instrukce NOP. Stav HALT se projeví nulovou úrovní výstupního signálu $\overline{\text{HALT}}$.

3.3.15 Instrukční soubor Z80-CPU

Tabulka 2

Použitá symbolika

B	délka instrukce (byte)
M	počet strojových cyklů
T	počet taktů

Použitá symbolika pro indikátory

-	indikátor nezměněn
1	indikátor nastaven
0	indikátor nulován
X	indikátor nedefinován
↕	indikátor nastaven podle výsledku operace
P	indikuje se parita
V	indikuje se přetečení
IFF	do indikátoru se přenese stav klopného obvodu IFF

Použitá symbolika pro operandy

s	r,n,(HL),(IX+d),(IY+d)
m	r,(HL),(IX+d),(IY+d)
CY	indikátor C
n	číslo rozsahu 1 Byte
e	číslo rozsahu (-126; 129)
d	doplněk rozsahu (-128; 127)
nn	číslo rozsahu 2 Byte

r,r	Registr
000	B
001	C
010	D
011	E
100	H
101	L
111	A

b	Bit
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

cc	Podmínka
000	NZ
001	Z
010	NC
011	C
100	PO
101	PE
110	P
111	M

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

dd	Registrový pár
00	BC
01	DE
10	HL
11	SP

qq	Registrový pár
00	BC
01	DE
10	HL
11	AF

ss	Registrový pár
00	BC
01	DE
10	HL
11	SP

pp	Registrový pár
00	BC
01	DE
10	IX
11	SP

rr	Registrový pár
00	BC
01	DE
10	IY
11	SP

B-počet bytov inštrukcie
M- počet strojových cyklov procesora
T - počet period (pri f=2,5 MHz T = 400ns)

Symbol inštrukce	Druh operace	PRÍZNAKY C	Z	P/V	S	N	H	76 543 210 oper. kód	BYTES B	M1 CYCLES M	PERIODS T	Ekvival. I 8080A
LD r,r´	r ← r´	–	–	–	–	–	–	01 r r´	1	1	4	áno
LD r,n	r ← n	–	–	–	–	–	–	00 r 110 n	2	2	7	áno
LD r,(HL)	r ← (HL)	–	–	–	–	–	–	01 r 110	1	2	7	áno
LD r,(IX+d)	r ← (IX+d)	–	–	–	–	–	–	11 011 101 01 r 110 d	3	5	19	
LD r,(IY+d)	r ← (IY+d)	–	–	–	–	–	–	11 111 101 01 r 110 d	3	5	19	
LD (HL),r	(HL) ← r	–	–	–	–	–	–	01 110 r	1	2	7	áno
LD (IX+d),r	(IX+d) ← r	–	–	–	–	–	–	11 011 101 01 110 r d	3	5	19	
LD (IY+d),r	(IY+d) ← r	–	–	–	–	–	–	11 111 101 01 110 r d	3	5	19	
LD (HL),n	(HL) ← n	–	–	–	–	–	–	00 110 110 n	2	3	10	áno
LD (IX+d),n	(IX+d) ← n	–	–	–	–	–	–	11 011 101 00 110 110 d n	4	5	19	
LD (IY+d),n	(IY+d) ← n	–	–	–	–	–	–	11 111 101 00 110 110 d n	4	5	19	

Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	B	M	T	Ekvival. I 8080A
									BYTES	MICYCL	PERIODS	
LD A,(BC)	A ← (BC)	–	–	–	–	–	–	00 001 010	1	2	7	ano
LD A,(DE)	A ← (DE)	–	–	–	–	–	–	00 011 010	1	2	7	ano
LD A,(nn)	A ← (nn)	–	–	–	–	–	–	00 111 010 n n	3	4	13	ano
LD (BC),A	(BC) ← A	–	–	–	–	–	–	00 000 010	1	2	7	ano
LD (DE),A	(DE) ← A	–	–	–	–	–	–	00 010 010	1	2	7	ano
LD (nn),A	(nn) ← A	–	–	–	–	–	–	00 110 010 n n	3	4	13	ano
LD A,I	A ← I	–	↕	IFF	↕	0	0	11 101 101 01 010 111	2	2	9	
LD A,R	A ← R	–	↕	IFF	↕	0	0	11 101 101 01 011 111	2	2	9	
LD I,A	I ← A	–	–	–	–	–	–	11 101 101 01 000 111	2	2	9	
LD R,A	R ← A	–	–	–	–	–	–	11 101 101 01 001 111	2	2	9	
LD dd,nn	dd ← nn	–	–	–	–	–	–	00 dd0 001 n n	3	3	10	
LD IX,nn	IX ← nn	–	–	–	–	–	–	11 011 101 00 100 001 n n	4	4	14	
LD IY,nn	IY ← nn	–	–	–	–	–	–	11 111 101 00 100 001 n n	4	4	14	

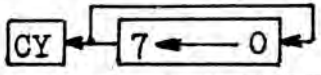
Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	BYTES B	STRO. M	PER. T	Ekvival. I 8080A
LD HL,(nn)	H ← (nn+1) L ← (nn)	–	–	–	–	–	–	00 101 010 n n	3	5	16	ano
LD dd,(nn)	dd_H ← (nn+1) dd_L ← (nn)	–	–	–	–	–	–	11 101 101 01 dd1 011 n n	4	6	20	
LD IX,(nn)	IX_H ← (nn+1) IX_L ← (nn)	–	–	–	–	–	–	11 011 101 00 101 010 n n	4	6	20	
LD IY,(nn)	IY_H ← (nn+1) IY_L ← (nn)	–	–	–	–	–	–	11 111 101 00 101 010 n n	4	6	20	
LD (nn),HL	(nn+1) ← H (nn) ← L	–	–	–	–	–	–	00 100 010 n n	3	5	16	ano
LD (nn),dd	(nn+1) ← dd_H (nn) ← dd_L	–	–	–	–	–	–	11 101 101 01 dd0 011 n n	4	6	20	
LD (nn),IX	(nn+1) ← IX_H (nn) ← IX_L	–	–	–	–	–	–	11 011 101 00 100 010 n n	4	6	20	
LD (nn),IY	(nn+1) ← IY_H (nn) ← IY_L	–	–	–	–	–	–	11 111 101 00 100 010 n n	4	6	20	
LD SP,HL	SP ← HL	–	–	–	–	–	–	11 111 001	1	1	6	ano

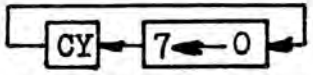
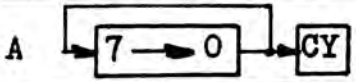
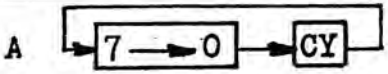
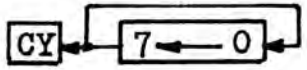
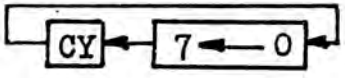
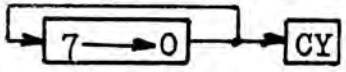
Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	BT	M1	T	Ekvival. I 8080A
LD SP,IX	SP ← IX	–	–	–	–	–	–	11 011 101 11 111 001	2	2	10	
LD SP,IY	SP ← IY	–	–	–	–	–	–	11 111 101 11 111 001	2	2	10	
PUSH qq	$(SP-2) \leftarrow qq_L$ $(SP-1) \leftarrow qq_H$	–	–	–	–	–	–	11 qq0 101	1	3	11	ano
PUSH IX	$(SP-2) \leftarrow IX_L$ $(SP-1) \leftarrow IX_H$	–	–	–	–	–	–	11 011 101 11 100 101	2	4	15	
PUSH IY	$(SP-2) \leftarrow IY_L$ $(SP-1) \leftarrow IY_H$	–	–	–	–	–	–	11 111 101 11 100 101	2	4	15	
POP qq	$qq_H \leftarrow (SP+1)$ $qq_L \leftarrow (SP)$	–	–	–	–	–	–	11 qq0 001	1	3	10	ano
POP IX	$IX_H \leftarrow (SP+1)$ $IX_L \leftarrow (SP)$	–	–	–	–	–	–	11 011 101 11 100 001	2	4	14	
POP IY	$IY_H \leftarrow (SP+1)$ $IY_L \leftarrow (SP)$	–	–	–	–	–	–	11 111 101 11 100 001	2	4	14	
EX DE,HL	DE ↔ HL	–	–	–	–	–	–	11 101 011	1	1	4	ano
EX AF,AF′	AF ↔ AF′	–	–	–	–	–	–	00 001 000	1	1	4	
EXX	BC ↔ BC′ DE ↔ DE′ HL ↔ HL′	–	–	–	–	–	–	11 011 001	1	1	4	
EX (SP),HL	H ↔ (SP+1) L ↔ (SP)	–	–	–	–	–	–	11 100 011	1	5	19	ano

Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	B BYT.	M M1	T PERIOD	Ekvival. I 8080A
CPD	A←(HL) HL←HL-1 BC←BC-1	–	↕	↕	↕	1	↕	11 101 101 10 101 001	2	4	16	
CPDR	A←(HL) HL←HL-1 BC←BC-1 REPEAT UNTIL A=(HL) ∨ BC=0	–	↕	↕	↕	1	↕	11 101 101 10 111 001	2 2	5 4	21 16	
ADD r	A←A+r	↕	↕	V	↕	0	↕	10 000 r	1	1	4	ano
ADD n	A←A+n	↕	↕	V	↕	0	↕	11 000 110 n	2	2	7	ano
ADD (HL)	A←A+(HL)	↕	↕	V	↕	0	↕	10 000 110	1	2	7	ano
ADD (IX+d)	A←A+(IX+d)	↕	↕	V	↕	0	↕	11 011 101 10 000 110 d	3	5	19	
ADD (IY+d)	A←A+(IY+d)	↕	↕	V	↕	0	↕	11 111 101 10 000 110 d	3	5	19	
ADC s	A←A+s+CY	↕	↕	V	↕	0	↕	001				ano
SUB s	A←A-s	↕	↕	V	↕	1	↕	010				ano
SBC s	A←A-s-CY	↕	↕	V	↕	1	↕	011				ano
AND s	A←A ∧ s	0	↕	P	↕	0	↕	100				ano
OR s	A←A ∨ s	0	↕	P	↕	0	↕	110				ano

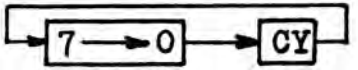
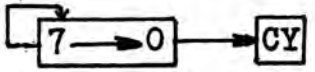
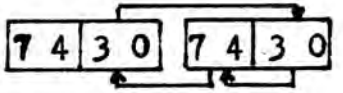
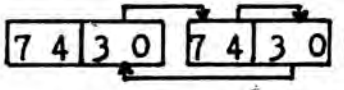
Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	B	M	T	Ekvival. I 8080A
EX (SP),IX	$IX_H \leftrightarrow (SP+1)$ $IX_L \leftrightarrow (SP)$	–	–	–	–	–	–	11 011 101 11 100 011	2	6	23	
EX (SP),IY	$IY_H \leftrightarrow (SP+1)$ $IY_L \leftrightarrow (SP)$	–	–	–	–	–	–	11 111 101 11 100 011	2	6	23	
LDI	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1	–	–	↕	–	0	0	11 101 101 10 100 000	2	4	16	
LDIR	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1 REPEAT UNTIL BC=0	–	–	0	–	0	0	11 101 101 10 110 000	2 2	5 4	21 16	
LDD	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1	–	–	↕	–	0	0	11 101 101 10 101 000	2	4	16	
LDDR	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1 REPEAT UNTIL BC=0	–	–	0	–	0	0	11 101 101 10 111 000	2 2	5 4	21 16	
CPI	A-(HL) HL ← HL+1 BC ← BC-1	–	↕	↕	↕	↕	↕	11 101 101 10 100 001	2	4	16	
CPIR	A-(HL) HL ← HL+1 BC ← BC-1 REPEAT UNTIL A=(HL) ∨ BC=0	–	↕	↕	↕	↕	↕	11 101 101 10 110 001	2 2	5 4	21 16	

Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	BYTES B	M1CYCL. M	PERIODS T	Ekvival. I 8080A
XOR s	A ← A ⊕ s	0	↕	P	↕	0	↕	101				ano
CP s	A — s	↕	↕	V	↕	1	↕	111				ano
INC r	r ← r+1	–	↕	V	↕	0	↕	00 r 100	1	1	4	ano
INC (HL)	(HL) ← (HL)+1	–	↕	V	↕	0	↕	00 110 100	1	3	11	ano
INC (IX+d)	(IX+d) ← (IX+d)+1	–	↕	V	↕	0	↕	11 011 101 00 110 100 d	3	6	23	
INC (IY+d)	(IY+d) ← (IY+d)+1	–	↕	V	↕	0	↕	11 111 101 00 110 100 d	3	6	23	
DEC m	m ← m-1	–	↕	V	↕	1	↕	101				ano
DAA	Oprava pro BCD kód	↕	↕	P	↕	–	↕	00 100 111	1	1	4	ano
CPL	A ← $\bar{A}$	–	–	–	–	1	1	00 101 111	1	1	4	ano
NEG	A ← 0-A	↕	↕	V	↕	1	↕	11 101 101 01 000 100	2	2	8	
CCF	CY ← $\overline{CY}$	↕	–	–	–	0	X	00 111 111	1	1	4	ano
SCF	CY ← 1	1	–	–	–	0	0	00 110 111	1	1	4	ano
NOP	Prázdná operace	–	–	–	–	–	–	00 000 000	1	1	4	ano
HALT	Zastavení	–	–	–	–	–	–	01 110 110	1	1	4	ano
DI	IFF ← 0	–	–	–	–	–	–	11 110 011	1	1	4	ano
EI	IFF ← 1	–	–	–	–	–	–	11 111 011	1	1	4	ano
IMØ	Přerušovací režim 0	–	–	–	–	–	–	11 101 101 01 000 110	2	2	8	

Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	B	M	T	Ekvival. I 8080A
IM1	Přerušovací režim 1	–	–	–	–	–	–	11 101 101 01 010 110	2	2	8	
IM2	Přerušovací režim 2	–	–	–	–	–	–	11 101 101 01 011 110	2	2	8	
ADD HL,ss	HL ← HL+ss	↕	–	–	–	0	X	00 ss1 001	1	3	11	ano
ADC HL,ss	HL ← HL+ss+CY	↕	↕	V	↕	0	X	11 101 101 01 ss1 010	2	4	15	
SBC HL,ss	HL ← HL-ss-CY	↕	↕	V	↕	1	X	11 101 101 01 ss0 010	2	4	15	
ADD IX,pp	IX ← IX+pp	↕	–	–	–	0	X	11 011 101 00 pp1 001	2	4	15	
ADD IY,rr	IY ← IY+rr	↕	–	–	–	0	X	11 111 101 00 rr1 001	2	4	15	
INC ss	ss ← ss+1	–	–	–	–	–	–	00 ss0 011	1	1	6	ano
INC IX	IX ← IX+1	–	–	–	–	–	–	11 011 101 00 100 011	2	2	10	
INC IY	IY ← IY+1	–	–	–	–	–	–	11 111 101 00 100 011	2	2	10	
DEC ss	ss ← ss-1	–	–	–	–	–	–	00 ss1 011	1	1	6	ano
DEC IX	IX ← IX-1	–	–	–	–	–	–	11 011 101 00 101 011	2	2	10	
DEC IY	IY ← IY-1	–	–	–	–	–	–	11 111 101 00 101 011	2	2	10	
RLCA		↕	–	–	–	0	0	00 000 111	1	1	4	ano

Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	B	M	T	Ekvival. I 8080A
RLA		↕	–	—	–	0	0	00 010 111	1	1	4	ano
RRCA		↕	–	—	–	0	0	00 000 111	1	1	4	ano
RRA		↕	–	–	–	0	0	00 011 111	1	1	4	ano
RLC r		↕	↕	P	↕	0	0	11 001 011 00 000 r	2	2	8	
RLC (HL)		↕	↕	P	↕	0	0	11 001 011 00 000 110	2	4	15	
RLC (IX+d)	 r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	11 011 101 11 001 011 d 00 000 110	4	6	23	
RLC (IY+d)		↕	↕	P	↕	0	0	11 111 101 11 001 011 d 00 000 110	4	6	23	
RL m	 r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	010				
RRC m	 r,(HL),(IX+d),(IX+d)	↕	↕	P	↕	0	0	001				

BYTES MCYCL. PERIODS

Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	B	M	T	Ekvival. I 8080A
RR m	 r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	011				
SLA m	r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	100				
SRA m	 r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	101				
SRL m	r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	111				
RLD		-	↕	P	↕	0	0	11 101 101 01 101 111	2	5	18	
RRD		-	↕	P	↕	0	0	11 101 101 01 100 111	2	5	18	
BIT b,r	$Z \leftarrow \overline{r}_b$	-	↕	X	X	0	1	11 001 011 01 b r	2	2	8	
BIT b,(HL)	$Z \leftarrow \overline{(HL)}_b$	-	↕	X	X	0	1	11 001 011 01 b 110	2	3	12	

Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	B (BYTES)	M (M.CYCLES)	T (PERIODS)	Ekvival. I 8080A
BIT b,(IX+d)	$Z \leftarrow (\overline{IX+d})_b$	–	↕	X	X	0	1	11 011 101 11 001 011 d 01 b 110	4	5	20	
BIT b,(IY+d)	$Z \leftarrow (\overline{IY+d})_b$	–	↕	X	X	0	1	11 111 101 11 001 011 d 01 b 110	4	5	20	
SET b,r	$r_b \leftarrow 1$	–	–	–	–	–	–	11 001 011 11 b r	2	2	8	
SET b,(HL)	$(HL)_b \leftarrow 1$	–	–	–	–	–	–	11 001 011 11 b 110	2	4	15	
SET b,(IX+d)	$(IX+d)_b \leftarrow 1$	–	–	–	–	–	–	11 011 101 11 001 011 d 11 b 110	4	6	23	
SET b,(IY+d)	$(IY+d)_b \leftarrow 1$	–	–	–	–	–	–	11 111 101 11 001 011 d 11 b 110	4	6	23	
RES b,m	$m_b \leftarrow 0$	–	–	–	–	–	–	10				
JP nn	PC ← nn	–	–	–	–	–	–	11 000 011 n n	3	3	10	ano
JP cc,nn	IF cc THEN PC ← nn	–	–	–	–	–	–	11 cc 010 n n	3	3	10	ano

Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	B	M	T	Ekvival. I 8080A
JR e	PC ← PC+e	–	–	–	–	–	–	00 011 000 e-2	2	3	12	
JR C,e	IF C=1 THEN PC ← PC+e	–	–	–	–	–	–	00 111 000 e-2	2 2	2 3	7 12	
JR NC,e	IF NC THEN PC ← PC+e	–	–	–	–	–	–	00 110 000 e-2	2	2	7	
JR Z,e	IF Z THEN PC ← PC+e	–	–	–	–	–	–	00 101 000 e-2	2 2	2 3	7 12	
JR NZ,e	IF NZ THEN PC ← PC+e	–	–	–	–	–	–	00 100 000 e-2	2 2	2 3	7 12	
JP (HL)	PC ← HL	–	–	–	–	–	–	11 101 001	1	1	4	ano
JP (IX)	PC ← IX	–	–	–	–	–	–	11 011 101 11 101 001	2	2	8	
JP (IY)	PC ← IY	–	–	–	–	–	–	11 111 101 11 101 001	2	2	8	
DJNZ e	B ← B-1 IF B≠0 THEN PC ← PC+e	–	–	–	–	–	–	00 010 000 e-2	2 2	2 3	8 13	
CALL nn	(SP-1) ← PC_H (SP-2) ← PC_L PC ← nn	–	–	–	–	–	–	11 001 101 n n	3	5	17	ano
CALL cc,nn	IF cc THEN CALL nn	–	–	–	–	–	–	11 cc 100 n n	3 3	3 5	10 17	ano
RET	PC_L ← (SP) PC_H ← (SP+1)	–	–	–	–	–	–	11 001 001	1	3	10	ano
RET cc	IF cc THEN RET	–	–	–	–	–	–	11 cc 000	1 1	1 3	5 11	ano

Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kod	BYTES B	M.CYCL. M	PERIODS T	Ekvival. I 8080A
RETI	Návrat z přerušení	–	–	–	–	–	–	11 101 101 01 001 101	2	4	14	
RETN	Návrat z nemaskovatelného přerušení	–	–	–	–	–	–	11 101 101 01 000 101	2	4	14	
RST p	(SP-1) ← PC_H (SP-2) ← PC_L PC_H ← 0 PC_L ← p	–	–	–	–	–	–	11 t 111	1	3	11	ano
IN A,(n)	A ← (n)	–	–	–	–	–	–	11 011 011 n	2	3	10	ano
IN r,(C)	r ← (C)	–	↕	P	↕	0	1	11 101 101 01 r 000	2	3	11	
INI	(HL) ← (C) B ← B-1 HL ← HL+1	–	↕	X	X	1	X	11 101 101 10 100 010	2	4	15	
INIR	(HL) ← (C) B ← B-1 HL ← HL+1 REPEAT UNTIL B=0	–	1	X	X	1	X	11 101 101 10 110 010	2 2	5 4	20 15	
IND	(HL) ← (C) B ← B-1 HL ← HL-1	–	↕	X	X	1	X	11 101 101 10 101 010	2	4	15	
INDR	(HL) ← (C) B ← B-1 HL ← HL-1 REPEAT UNTIL B=0	–	1	X	X	1	X	11 101 101 10 111 010	2 2	5 4	20 15	
OUT (n),A	(n) ← A	–	–	–	–	–	–	11 010 011 n	2	3	11	ano

Symbol instrukce	Druh operace	C	Z	P/V	S	N	H	76 543 210 oper. kód	B (BYTES)	M (MCYCL.)	T (PERIODS)	Ekvival. I 8080A
OUT (C),r	(C)← r	–	–	–	–	–	–	11 101 101 01 r 001	2	3	12	
OUTI	(C)← (HL) B← B-1 HL← HL+1	–	↕	X	X	1	X	11 101 101 10 100 011	2	4	15	
OTIR	(C)← (HL) B← B-1 HL← HL+1 REPEAT UNTIL B=0	–	1	X	X	1	X	11 101 101 10 110 011	2 2	5 4	20 15	
OUTD	(C)← (HL) B← B-1 HL← HL-1	–	↕	X	X	1	X	11 101 101 10 101 011	2	4	15	
OTDR	(C)← (HL) B← B-1 HL← HL-1 REPEAT UNTIL B=0	–	1	X	X	1	X	11 101 101 10 111 011	2 2	5 4	20 15	

Poznámka 1

Blokové instrukce s opakováním typu LDIR, LDDR, CPIR, CPDR, INIR, INDR, OTIR, OTDR mají rozdílný počet strojových cyklů M vzhledem ke stavu registru B nebo párového registru BC, kde je uložena informace o přenášené zóně bloku.

Při B ≠ 0 nebo BC ≠ 0 - 5M.

Při B = 0 nebo BC = 0 - 4M.

Poznámka 2

Blokové instrukce lze přerušit v jejich průběhu, protože jsou řešeny jako fiktivní programové smyčky. Je nutné zachovat obsah registrů BC, DE, HL. Při přerušení se do zásobníku ukládá adresa téže instrukce, a ne následující, jako tomu je u ostatních instrukcí.

3.3.16 Seznam instrukcí mikroprocesoru Z80-CPU podle operačních kódů

Tabulka 3 poskytuje přehled instrukcí řazených dle rostoucích hodnot operačních kódů. Délka instrukcí je jedno až čtyřbytová a jejich symbolika vychází z konvencí používaných v tabulce 2. Pro zajištění konkrétních údajů v operandové části instrukcí jsou použity hexadecimální hodnoty vyjádřené v následujícím přehledu.

N	číslo rozsahu 1 Byte (v tabulce uvedeno 20 H)
NN	číslo rozsahu 2 Byte (v tabulce uvedeno 8405 H)
d	doplněk rozsahu - 128; 127 (v tabulce uvedeno 05 H
DIS	číslo rozsahu - 126; 129 (v tabulce uvedeno 2E H)

Tabulka 3

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
00	NOP	12	LD (DE),A
018405	LD BC,NN	13	INC DE
02	LD (BC),A	14	INC D
03	INC BC	15	DEC D
04	INC B	1620	LD D,N
05	DEC B	17	RLA
0620	LD B,N	182E	JR DIS
07	RLCA	19	ADD HL,DE
08	EX AF,AF´	1A	LD A,(DE)
09	ADD HL,BC	1B	DEC DE
0A	LD A,(BC)	1C	INC E
0B	DEC BC	1D	DEC E
0C	INC C	1E20	LD E,N
0D	DEC C	1F	RRA
0E20	LD C,N	202E	JR NZ,DIS
0F	RRCA	218405	LD HL,NN
102E	DJNZ DIS	228405	LD (NN),HL
118405	LD DE,NN	23	INC HL

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
24	INC H	46	LD B,(HL)
25	DEC H	47	LD B,A
2620	LD H,N	48	LD C,B
27	DAA	49	LD C,C
282E	JR Z,DIS	4A	LD C,D
29	ADD HL,HL	4B	LD C,E
2A8405	LD HL,(NN)	4C	LD C,H
2B	DEC HL	4D	LD C,L
2C	INC L	4E	LD C,(HL)
2D	DEC L	4F	LD C,A
2E20	LD L,N	50	LD D,B
2F	CPL	51	LD D,C
302E	JR NC,DIS	52	LD D,D
318405	LD SP,NN	53	LD D,E
328405	LD (NN),A	54	LD D,H
33	INC SP	55	LD D,L
34	INC (HL)	56	LD D,(HL)
35	DEC (HL)	57	LD D,A
3620	LD (HL),N	58	LD E,B
37	SCF	59	LD E,C
382E	JR C,DIS	5A	LD E,D
39	ADD HL,SP	5B	LD E,E
3A8405	LD A,(NN)	5C	LD E,H
3B	DEC SP	5D	LD E,L
3C	INC A	5E	LD E,(HL)
3D	DEC A	5F	LD E,A
3E20	LD A,N	60	LD H,B
3F	CCF	61	LD H,C
40	LD B,B	62	LD H,D
41	LD B,C	63	LD H,E
42	LD B,D	64	LD H,H
43	LD B,E	65	LD H,L
44	LD B,H	66	LD H,(HL)
45	LD B,L	67	LD H,A

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
68	LD L,B	8A	ADC A,D
69	LD L,C	8B	ADC A,E
6A	LD L,D	8C	ADC A,H
6B	LD L,E	8D	ADC A,L
6C	LD L,H	8E	ADC A,(HL)
6D	LD L,L	8F	ADC A,A
6E	LD L,(HL)	90	SUB B
6F	LD L,A	91	SUB C
70	LD (HL),B	92	SUB D
71	LD (HL),C	93	SUB E
72	LD (HL),D	94	SUB H
73	LD (HL),E	95	SUB L
74	LD (HL),H	96	SUB (HL)
75	LD (HL),L	97	SUB A
76	HALT	98	SBC A,B
77	LD (HL),A	99	SBC A,C
78	LD A,B	9A	SBC A,D
79	LD A,C	9B	SBC A,E
7A	LD A,D	9C	SBC A,H
7B	LD A,E	9D	SBC A,L
7C	LD A,H	9E	SBC A,(HL)
7D	LD A,L	9F	SBC A,A
7E	LD A,(HL)	A0	AND B
7F	LD A,A	A1	AND C
80	ADD A,B	A2	AND D
81	ADD A,C	A3	AND E
82	ADD A,D	A4	AND H
83	ADD A,E	A5	AND L
84	ADD A,H	A6	AND (HL)
85	ADD A,L	A7	AND A
86	ADD A,(HL)	A8	XOR B
87	ADD A,A	A9	XOR C
88	ADC A,B	AA	XOR D
89	ADC A,C	AB	XOR E

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
AC	XOR H	CF	RST 8
AD	XOR L	D0	RET NC
AE	XOR (HL)	D1	POP DE
AF	XOR A	D28405	JP NC,NN
B0	OR B	D320	OUT (N),A
B1	OR C	D48405	CALL NC,NN
B2	OR D	D5	PUSH DE
B3	OR E	D620	SUB N
B4	OR H	D7	RST 10H
B5	OR L	D8	RET C
B6	OR (HL)	D9	EXX
B7	OR A	DA8405	JP C,NN
B8	CP B	DB20	IN A,(N)
B9	CP C	DC8405	CALL C,N
BA	CP D	DE20	SBC A,N
BB	CP E	DF	RST 18H
BC	CP H	E0	RET PO
BD	CP L	E1	POP HL
BE	CP (HL)	E28405	JP PO,NN
BF	CP A	E3	EX (SP),HL
C0	RET NZ	E48405	CALL PO,NN
C1	POP BC	E5	PUSH HL
C28405	JP NZ,NN	E620	AND N
C38405	JP NN	E7	RST 20H
C48405	CALL NZ,NN	E8	RET PE
C5	PUSH BC	E9	JP (HL)
C620	ADD A,N	EA8405	JE PE,NN
C7	RST 0	EC8405	CALL PE,NN
C8	RET Z	EE20	XOR N
C9	RET	EF	RST 28H
CA8405	JP Z,NN	F0	RET P
CC8405	CALL Z,NN	F1	POP AF
CD8405	CALL NN	F28405	JP P,NN
CE20	ADC A,N	F3	DI
		EB	EX DE,HL

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
F48405	CALL P,NN	CB17	RL A
F5	PUSH AF	CB18	RR B
F620	OR N	CB19	RR C
F7	RST 30H	CB1A	RR D
F8	RET M	CB1B	RR E
F9	LD SP,HL	CB1C	RR H
FA8405	JP M,NN	CB1D	RR L
FB	EI	CB1E	RR (HL)
FC8405	CALL M,NN	CB1F	RR A
FE20	CP N	CB20	SLA B
FF	RST 38H	CB21	SLA C
CB00	RLC B	CB22	SLA D
CB01	RLC C	CB23	SLA E
CB02	RLC D	CB24	SLA H
CB03	RLC E	CB25	SLA L
CB04	RLC H	CB26	SLA (HL)
CB05	RLC L	CB27	SLA A
CB06	RLC (HL)	CB28	SRA B
CB07	RLC A	CB29	SRA C
CB08	RRC B	CB2A	SRA D
CB09	RRC C	CB2B	SRA E
CB0A	RRC D	CB2C	SRA H
CB0B	RRC E	CB2D	SRA L
CB0C	RRC H	CB2E	SRA (HL)
CB0D	RRC L	CB2F	SRA A
CB0E	RRC (HL)	CB38	SRL B
CB0F	RRC A	CB39	SRL C
CB10	RL B	CB3A	SRL D
CB11	RL C	CB3B	SRL E
CB12	RL D	CB3C	SRL H
CB13	RL E	CB3D	SRL L
CB14	RL H	CB3E	SRL (HL)
CB15	RL L	CB3F	SRL A
CB16	RL (HL)	CB40	BIT 0,B

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
CB41	BIT 0,C	CB63	BIT 4,E
CB42	BIT 0,D	CB64	BIT 4,H
CB43	BIT 0,E	CB65	BIT 4,L
CB44	BIT 0,H	CB66	BIT 4,(HL)
CB45	BIT 0,L	CB67	BIT 4,A
CB46	BIT 0,(HL)	CB68	BIT 5,B
CB47	BIT 0,A	CB69	BIT 5,C
CB48	BIT 1,B	CB6A	BIT 5,D
CB49	BIT 1,C	CB6B	BIT 5,E
CB4A	BIT 1,D	CB6C	BIT 5,H
CB4B	BIT 1,E	CB6D	BIT 5,L
CB4C	BIT 1,H	CB6E	BIT 5,(HL)
CB4D	BIT 1,L	CB6F	BIT 5,A
CB4E	BIT 1,(HL)	CB70	BIT 6,B
CB4F	BIT 1,A	CB71	BIT 6,C
CB50	BIT 2,B	CB72	BIT 6,D
CB51	BIT 2,C	CB73	BIT 6,E
CB52	BIT 2,D	CB74	BIT 6,H
CB53	BIT 2,E	CB75	BIT 6,L
CB54	BIT 2 H	CB76	BIT 6,(HL)
CB55	BIT 2,L	CB77	BIT 6,A
CB56	BIT 2,(HL)	CB78	BIT 7,B
CB57	BIT 2,A	CB79	BIT 7,C
CB58	BIT 3,B	CB7A	BIT 7,D
CB59	BIT 3,C	CB7B	BIT 7,E
CB5A	BIT 3,D	CB7C	BIT 7,H
CB5B	BIT 3,E	CB7D	BIT 7,L
CB5C	BIT 3,H	CB7E	BIT 7,(HL)
CB5D	BIT 3,L	CB7F	BIT 7,A
CB5E	BIT 3,(HL)	CB80	RES 0,B
CB5F	BIT 3,A	CB81	RES 0,C
CB60	BIT 4,B	CB82	RES 0,D
CB61	BIT 4,C	CB83	RES 0,E
CB62	BIT 4,D	CB84	RES 0,H

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
CB85	RES 0,L	CBA7	RES 4,A
CB86	RES 0,(HL)	CBA8	RES 5,B
CB87	RES 0,A	CBA9	RES 5,C
CB88	RES 1,B	CBAA	RES 5,D
CB89	RES 1,C	CBAB	RES 5,E
CB8A	RES 1,D	CBAC	RES 5,H
CB8B	RES 1,E	CBAD	RES 5,L
CB8C	RES 1,H	CBAE	RES 5,(HL)
CB8D	RES 1,L	CBAF	RES 5,A
CB8E	RES 1,(HL)	CBB0	RES 6,B
CB8F	RES 1,A	CBB1	RES 6,C
CB90	RES 2,B	CBB2	RES 6,D
CB91	RES 2,C	CBB3	RES 6,E
CB92	RES 2,D	CBB4	RES 6,H
CB93	RES 2,E	CBB5	RES 6,L
CB94	RES 2,H	CBB6	RES 6,(HL)
CB95	RES 2,L	CBB7	RES 6,A
CB96	RES 2,(HL)	CBB8	RES 7,B
CB97	RES 2,A	CBB9	RES 7,C
CB98	RES 3,B	CBBA	RES 7,D
CB99	RES 3,C	CBBB	RES 7,E
CB9A	RES 3,D	CBBC	RES 7,H
CB9B	RES 3,E	CBBD	RES 7,L
CB9C	RES 3,H	CBBE	RES 7,(HL)
CB9D	RES 3,L	CBBF	RES 7,A
CB9E	RES 3,(HL)	CBC0	SET 0,B
CB9F	RES 3,A	CBC1	SET 0,C
CBA0	RES 4,B	CBC2	SET 0,D
CBA1	RES 4,C	CBC3	SET 0,E
CBA2	RES 4,D	CBC4	SET 0,H
CBA3	RES 4,E	CBC5	SET 0,L
CBA4	RES 4,H	CBC6	SET 0,(HL)
CBA5	RES 4,L	CBC7	SET 0,A
CBA6	RES 4,(HL)	CBC8	SET 1,B

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
CBC9	SET 1,C	CBEC	SET 5,H
CBCA	SET 1,D	CBED	SET 5,L
CBCB	SET 1,E	CBEE	SET 5,(HL)
CBCC	SET 1,H	CBEF	SET 5,A
CBCD	SET 1,L	CBF0	SET 6,B
CBCE	SET 1,(HL)	CBF1	SET 6,C
CBCF	SET 1,A	CBF2	SET 6,D
CBD0	SET 2,B	CBF3	SET 6,E
CBD1	SET 2,C	CBF4	SET 6,H
CBD2	SET 2,D	CBF5	SET 6,L
CBD3	SET 2,E	CBF6	SET 6,(HL)
CBD4	SET 2,H	CBF7	SET 6,A
CBD5	SET 2,L	CBF8	SET 7,B
CBD6	SET 2,(HL)	CBF9	SET 7,C
CBD7	SET 2,A	CBFA	SET 7,D
CBD8	SET 3,B	CBFB	SET 7,E
CBD9	SET 3,C	CBFC	SET 7,H
CBDA	SET 3,D	CBFD	SET 7,L
CBDB	SET 3,E	CBFE	SET 7,(HL)
CBDC	SET 3,H	CBFF	SET 7,A
CBDD	SET 3,L	DD09	ADD IX,BC
CBDE	SET 3,(HL)	DD19	ADD IX,DE
CBDF	SET 3,A	DD218405	LD IX,NN
CBE0	SET 4,B	DD228405	LD (NN),IX
CBE1	SET 4,C	DD23	INC IX
CBE2	SET 4,D	DD29	ADD IX,IX
CBE3	SET 4,E	DD2A8405	LD IX,(NN)
CBE5	SET 4,L	DD2B	DEC IX
CBE6	SET 4,(HL)	DD3405	INC (IX+d)
CBE7	SET 4,A	DD3505	DEC (IX+d)
CBE8	SET 5,B	DD360520	LD (IX+d),N
CBE9	SET 5,C	DD39	ADD IX,SP
CBEA	SET 5,D	DD4605	LD B,(IX+d)
CBEB	SET 5,E	DD4E05	LD C,(IX+d)
CBE4	SET 4,H		

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
DD5605	LD D,(IX+d)	DDCB0556	BIT 2,(IX+d)
DD5E05	LD E,(IX+d)	DDCB055E	BIT 3,(IX+d)
DD6605	LD H,(IX+d)	DDCB0566	BIT 4,(IX+d)
DD6E05	LD L,(IX+d)	DDCB056E	BIT 5,(IX+d)
DD7005	LD (IX+d),B	DDBC0576	BIT 6,(IX+d)
DD7105	LD (IX+d),C	DDCB057E	BIT 7,(IX+d)
DD7205	LD (IX+d),D	DDCB0586	RES 0,(IX+d)
DD7305	LD (IX+d),E	DDCB058E	RES 1,(IX+d)
DD7405	LD (IX+d),H	DDCB0596	RES 2,(IX+d)
DD7505	LD (IX+d),L	DDCB059E	RES 3,(IX+d)
DD7705	LD (IX+d),A	DDCB05A6	RES 4,(IX+d)
DD7E05	LD A,(IX+d)	DDCB05B6	RES 6,(IX+d)
DD8605	ADD A,(IX+d)	DDCB05BE	RES 7,(IX+d)
DD8E05	ADC A,(IX+d)	DDCB05C6	SET 0,(IX+d)
DD9605	SUB (IX+d)	DDCB05CE	SET 1,(IX+d)
DD9E05	SBC A,(IX+d)	DDCB05D6	SET 2,(IX+d)
DDA605	AND (IX+d)	DDCB05DE	SET 3,(IX+d)
DDAE05	XOR (IX+d)	DDCB05E6	SET 4,(IX+d)
DDB605	OR (IX+d)	DDCB05EE	SET 5,(IX+d)
DDBE05	CP (IX+d)	DDCB05F6	SET 6,(IX+d)
DDE1	POP IX	DDCB05FE	SET 7,(IX+d)
DDE3	EX (SP),IX	ED40	IN B,(C)
DDE5	PUSH IX	ED41	OUT (C),B
DDE9	JP (IX)	ED42	SBC HL,BC
DDF9	LD SP,IX	ED438405	LD (NN),BC
DDCB0506	RLC (IX+d)	ED44	NEG
DDCB050E	RRC (IX+d)	ED45	RETN
DDCB0516	RL (IX+d)	ED46	IM0
DDCB051E	RR (IX+d)	ED47	LD I,A
DDCB0526	SLA (IX+d)	ED48	IN C,(C)
DDCB052E	SRA (IX+d)	ED49	OUT (C),C
DDCB053E	SRL (IX+d)	ED4A	ADC HL,BC
DDCB0546	BIT 0,(IX+d)	ED4B8405	LD BC,(NN)
DDCB054E	BIT 1,(IX+d)	ED4D	RETI
DDCB05AE	RES 5,(IX+d)		

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
ED50	IN D,(C)	EDB1	CPIR
ED51	OUT (C),D	EDB2	INIR
ED52	SBC HL,DE	EDB3	OTIR
ED538405	LD (NN),DE	EDB8	LDDR
ED56	IM 1	EDB9	CPDR
ED57	LD A,I	EDBA	INDR
ED58	IN E,(C)	EDBB	OTDR
ED59	OUT (C),E	FD09	ADD IY,BC
ED5A	ADC HL,DE	FD19	ADD IY,DE
ED5B8405	LD DE,(NN)	FD218405	LD IY,NN
ED5E	IM 2	FD228405	LD (NN),IY
ED60	IN H,(C)	FD23	INC IY
ED61	OUT (C),H	FD29	ADD IY,IY
ED62	SBC HL,HL	FD2A8405	LD IY,(NN)
ED67	RRD	FD2B	DEC IY
ED68	IN L,(C)	FD3405	INC (IY+d)
ED69	OUT (C),L	FD3505	DEC (IY+d)
ED6A	ADC HL,HL	FD360520	LD (IY+d),N
ED6F	RLD	FD39	ADD IY,SP
ED72	SBC HL,SP	FD4605	LD B,(IY+d)
ED738405	LD (NN),SP	FD4E05	LD C,(IY+d)
ED78	IN A,(C)	FD5605	LD D,(IX+d)
ED79	OUT (C),A	FD5E05	LD E,(IY+d)
ED7A	ADC HL,SP	FD6605	LD H,(IY+d)
ED7B8405	LD SP,(NN)	FD6E05	LD L,(IY+d)
EDA0	LDI	FD7005	LD (IY+d),B
EDA1	CPI	FD7105	LD (IY+d),C
EDA2	INI	FD7205	LD (IY+d),D
EDA3	OUTI	FD7305	LD (IY+d),E
EDA8	LDD	FD7405	LD (IY+d),H
EDA9	CPD	FD7505	LD (IY+d),L
EDAA	IND	FD7705	LD (IY+d),A
EDAB	OUTD	FD7E05	LD A,(IY+d)
EDB0	LDIR	FD8605	ADD A,(IY+d)

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
FD8E05	ADD A,(IY+d)	FDCB05BE	RES 7,(IY+d)
FD9605	SUB (IY+d)	FDCB05C6	SET 0,(IY+d)
FD9E05	SBC A,(IY+d)	FDCB05CE	SET 1,(IY+d)
FDA605	AND (IY+d)	FDCB05D6	SET 2,(IY+d)
FDAE05	XOR (IY+d)	FDCB05DE	SET 3,(IY+d)
FDB605	OR (IY+d)	FDCB05E6	SET 4,(IY+d)
FDBE05	CP (IY+d)	FDCB05EE	SET 5,(IY+d)
FDE1	POP IY	FDCB05F6	SET 6,(IY+d)
FDE3	EX (SP),IY	FDCB05FE	SET 7,(IY+d)
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,(IX+d)		
FDCB055E	BIT 3,(IX+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)		

3.3.17 Abecední seznam instrukcí mikroprocesoru Z80-CPU

Tabulka 4 představuje seznam abecedně seřazených instrukcí. Je zde uveden symbolický tvar instrukce a strojový kód včetně operandové části. Symbolika vychází z konvencí použitých v kapitole 3.3.15 a 3.3.16.

Tabulka 4

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
8E	ADC A,(HL)	C620	ADD A,N
DD8E05	ADC A,(IX+d)	09	ADD HL,BC
FD8E05	ADC A,(IX+d)	19	ADD HL,DE
8F	ADC A,A	29	ADD HL,HL
88	ADC A,B	39	ADD HL,SP
89	ADC A,C	DD09	ADD IX,BC
8A	ADC A,D	DD19	ADD IX,DE
8B	ADC A,E	DD29	ADD IX,IX
8C	ADC A,H	DD39	ADD IX,SP
8D	ADC A,L	FD09	ADD IY,BC
CE20	ADC A,N	FD19	ADD IY,DE
ED4A	ADC HL,BC	FD29	ADD IY,IY
ED5A	ADC HL,DE	FD39	ADD IY,SP
ED6A	ADC HL,HL	A6	AND (HL)
ED7A	ADC HL,SP	DDA605	AND (IX+d)
86	ADD A,(HL)	FDA605	AND (IY+d)
DD8605	ADD A,(IX+d)	A7	AND A
FD8605	ADD A,(IY+d)	A0	AND B
87	ADD A,A	A1	AND C
80	ADD A,B	A2	AND D
81	ADD A,C	A3	AND E
82	ADD A,D	A4	AND H
83	ADD A,E	A5	AND L
84	ADD A,H	E620	AND N
85	ADD A,L	CB46	BIT 0,(HL)

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
DDCB0546	BIT 0,(IX+d)	CB59	BIT 3,C
FDCB0546	BIT 0,(IY+d)	CB5A	BIT 3,D
CB47	BIT 0,A	CB5B	BIT 3,E
CB40	BIT 0,B	CB5C	BIT 3,H
CB41	BIT 0,C	CB5D	BIT 3,L
CB42	BIT 0,D	CB66	BIT 4,(HL)
CB43	BIT 0,E	DDCB0566	BIT 4,(IX+d)
CB44	BIT 0,H	FDCB0566	BIT 4,(IY+d)
CB45	BIT 0,L	CB67	BIT 4,A
CB4E	BIT 1,(HL)	CB60	BIT 4,B
DDCB054E	BIT 1,(IX+d)	CB61	BIT 4,C
FDCB054E	BIT 1,(IY+d)	CB62	BIT 4,D
CB4F	BIT 1,A	CB63	BIT 4,E
BC48	BIT 1,B	CB64	BIT 4,H
CB49	BIT 1,C	CB65	BIT 4,L
CB4A	BIT 1,D	CB6E	BIT 5,(HL)
CB4B	BIT 1,E	DDCB056E	BIT 5,(IX+d)
CB4C	BIT 1,H	FDCB056E	BIT 5,(IY+d)
CB4D	BIT 1,L	CB6F	BIT 5,A
CB56	BIT 2,(HL)	CB68	BIT 5,B
DDCB0556	BIT 2,(IX+d)	CB69	BIT 5,C
FDCB0556	BIT 2,(IY+d)	CB6A	BIT 5,D
CB57	BIT 2,A	CB6B	BIT 5,E
CB50	BIT 2,B	CB6C	BIT 5,H
CB51	BIT 2,C	CB6D	BIT 5,L
CB52	BIT 2,D	CB76	BIT 6,(HL)
CB53	BIT 2,E	DDCB0576	BIT 6,(IX+d)
CB54	BIT 2,H	FDCB0576	BIT 6,(IY+d)
CB55	BIT 2,L	CB77	BIT 6,A
CB5E	BIT 3,(HL)	CB70	BIT 6,B
DDCB055E	BIT 3,(IX+d)	CB71	BIT 6,C
FDCB055E	BIT 3,(IY+d)	CB72	BIT 6,D
CB5F	BIT 3,A	CB73	BIT 6,E
CB58	BIT 3,B	CB74	BIT 6,H

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
CB75	BIT 6,L	EDA1	CPI
CB7E	BIT 7,(HL)	EDB1	CPIR
DDCB057E	BIT 7,(IX+d)	2F	CPL
FDCB057E	BIT 7,(IY+d)	27	DAA
CB7F	BIT 7,A	35	DEC (HL)
CB78	BIT 7,B	DD3505	DEC (IX+d)
CB79	BIT 7,C	FD3505	DEC (IY+d)
CB7A	BIT 7,D	3D	DEC A
CB7B	BIT 7,E	05	DEC B
CB7C	BIT 7,H	0B	DEC BC
CB7D	BIT 7,L	0D	DEC C
DC8405	CALL C,NN	15	DEC D
FC8405	CALL M,NN	1B	DEC DE
D48405	CALL NC,NN	1D	DEC E
CDB405	CALL NN	25	DEC H
C48405	CALL NZ,NN	2B	DEC HL
F48405	CALL P,NN	DD2B	DEX IX
EC8405	CALL PE,NN	FD2B	DEC IY
E48405	CALL PO,NN	2D	DEC L
CC8405	CALL Z,NN	3B	DEC SP
3F	CCF	F3	DI
BE	CP (HL)	102E	DJNZ DIS
DDBE05	CP (IX+d)	FB	EI
FDBE05	CP (IY+d)	E3	EX (SP),HL
BF	CP A	DDE3	EX (SP),IX
B8	CP B	FDE3	EX (SP),IY
B9	CP C	08	EX AF,AF´
BA	CP D	EB	EX DE,HL
BB	CP E	D9	EXX
BC	CP H	76	HALT
BD	CP L	ED46	IM0
FE20	CP N	ED56	IM1
EDA9	CPD	ED5E	IM2
EDB9	CPDR	ED78	IN A,(C)

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
DB20	IN A,(N)	C28405	JP NZ,NN
ED40	IN B,(C)	F28405	JP P,NN
ED48	IN C,(C)	EA8405	JP PE,NN
ED50	IN D,(C)	E28405	JP PO,NN
ED58	IN E,(C)	CA8405	JP Z,NN
ED60	IN H,(C)	382E	JR C,DIS
ED68	IN L,(C)	182E	JR DIS
34	INC (HL)	302E	JR NC,DIS
DD3405	INC (IX+d)	202E	JR NZ,DIS
FD3405	INC (IY+d)	282E	JR Z,DIS
3C	INC A	02	LD (BC),A
04	INC B	12	LD (DE),A
03	INC BC	77	LD (HL),A
0C	INC C	70	LD (HL),B
14	INC D	71	LD (HL),C
13	INC DE	72	LD (HL),D
1C	INC E	73	LD (HL),E
24	INC H	74	LD (HL),H
23	INC HL	75	LD (HL),L
DD23	INC IX	3620	LD (HL),N
FD23	INC IY	DD7705	LD (IX+d),A
2C	INC L	DD7005	LD (IX+d),B
33	INC SP	DD7105	LD (IX+d),C
EDAA	IND	DD7205	LD (IX+d),D
EDBA	INDR	DD7305	LD (IX+d),E
EDA2	INI	DD7405	LD (IX+d),H
EDB2	INIR	DD7505	LD (IX+d),L
E9	JP (HL)	DD360520	LD (IX+d),N
DDE9	JP (IX)	FD7705	LD (IY+d),A
FDE9	JP (IY)	FD7005	LD (IY+d),B
DA8405	JP C,NN	FD7105	LD (IY+d),C
FA8405	JP M,NN	FD7205	LD (IY+d),D
D28405	JP NC,NN	FD7305	LD (IY+d),E
C38405	JP NN	FD7405	LD (IY+d),H

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
FD7505	LD (IY+d),L	0620	LD B,N
FD360520	LD (IY+d),N	ED4B8405	LD BC,(NN)
328405	LD (NN),A	018405	LD BC,NN
ED438405	LD (NN),BC	4E	LD C,(HL)
ED538405	LD (NN),DE	DD4E05	LD C,(IX+ d)
228405	LD (NN),HL	FD4E05	LD C,(IY+d)
DD228405	LD (NN),IX	4F	LD C,A
FD228405	LD (NN),IY	48	LD C,B
ED738405	LD (NN),SP	49	LD C,C
0A	LD A,(BC)	4A	LD C,D
1A	LD A,(DE)	4B	LD C,E
7E	LD A,(HL)	4C	LD C,H
DD7E05	LD A,(IX+d)	4D	LD C,L
FD7E05	LD A,(IY+d)	0E20	LD C,N
3A8405	LD A,(NN)	56	LD D,(HL)
7F	LD A,A	DD5605	LD D,(IX+d)
78	LD A,B	FD5605	LD D,(IY+d)
79	LD A,C	57	LD D,A
7A	LD A,D	50	LD D,B
7B	LD A,E	51	LD D,C
7C	LD A,H	52	LD D,D
ED57	LD A,I	53	LD D,E
7D	LD A,L	54	LD D,H
3E20	LD A,N	55	LD D,L
46	LD B,(HL)	1620	LD D,N
DD4605	LD B,(IX+d)	ED5B8405	LD DE,(NN)
FD4605	LD B,(IY+d)	118405	LD DE,NN
47	LD B,A	5E	LD E,(HL)
40	LD B,B	DD5E05	LD E,(IX+d)
41	LD B,C	FD5E05	LD E,(IY+d)
42	LD B,D	5F	LD E,A
43	LD B,E	58	LD E,B
44	LD B,H,NN	59	LD E,C
45	LD B,L	5A	LD E,D

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
5B	LD E,E	F9	LD SP,HL
5C	LD E,H	DDF9	LD SP,IX
5D	LD E,L	FDF9	LD SP,IY
1E20	LD E,N	318405	LD SP,NN
66	LD H,(HL)	EDA8	LDD
DD6605	LD H,(IX+d)	EDB8	LDDR
FD6606	LD H,(IY+d)	EDA0	LDI
67	LD H,A	EDB0	LDIR
60	LD H,B	ED44	NEG
61	LD H,C	00	NOP
62	LD H,D	B6	OR (HL)
63	LD H,E	DDB605	OR (IX+d)
64	LD H,H	FDB605	OR (IY+d)
65	LD H,L	B7	OR A
2620	LD H,N	B0	OR B
2A8405	LD HL,(NN)	B1	OR C
218405	LD HL,NN	B2	OR D
ED47	LD I,A	B3	OR E
DD2A8405	LD IX,(NN)	B4	OR H
DD218405	LD IX,NN	B5	OR L
FD2A8405	LD IY,(NN)	F620	OR N
FD218405	LD IY,NN	EDBB	OTDR
6E	LD L,(HL)	EDB3	OTIR
DD6E05	LD L,(IX+d)	ED79	OUT (C),A
FD6E05	LD L,(IY+d)	ED41	OUT (C),B
6F	LD L,A	ED49	OUT (C),C
68	LD L,B	ED51	OUT (C),D
69	LD L,C	ED59	OUT (C),E
6A	LD L,D	ED61	OUT (C),H
6B	LD L,E	ED69	OUT (C),L
6C	LD L,H	D320	OUT (N),A
6D	LD L,L	EDAB	OUTD
2E20	LD L,N	EDA3	OUTI
ED7B8405	LD SP,(NN)	F1	POP AF

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
C1	POP BC	CB97	RES 2,A
D1	POP DE	CB90	RES 2,B
E1	POP HL	CB91	RES 2,C
DDE1	POP IX	CB92	RES 2,D
FDE1	POP IY	CB93	RES 2,E
F5	PUSH AF	CB94	RES 2,H
C5	PUSH BC	CB95	RES 2,L
D5	PUSH DE	CB9E	RES 3,(HL)
E5	PUSH HL	DDCB059E	RES 3,(IX+d)
DDE5	PUSH IX	FDCB059E	RES 3,(IY+d)
FDE5	PUSH IY	CB9F	RES 3,A
CB86	RES 0,(HL)	CB98	RES 3,B
DDCB0586	RES 0,(IX+d)	CB99	RES 3,C
FDCB0586	RES 0,(IY+d)	CB9A	RES 3,D
CB87	RES 0,A	CB9B	RES 3,E
CB80	RES 0,B	CB9C	RES 3,H
CB81	RES 0,C	CB9D	RES 3,L
CB82	RES 0,D	CBA6	RES 4,(HL)
CB83	RES 0,E	DDCB05A6	RES 4,(IX+d)
CB84	RES 0,H	FDCB05A6	RES 4,(IY+d)
CB85	RES 0,L	CBA7	RES 4,A
CB8E	RES 1,(HL)	CBA0	RES 4,B
DDCB058E	RES 1,(IX+d)	CBA1	RES 4,C
FDCB058E	RES 1,(IY+d)	CBA2	RES 4,D
CB8F	RES 1,A	CBA3	RES 4,E
CB88	RES 1,B	CBA4	RES 4,H
CB89	RES 1,C	CBA5	RES 4,L
CB8A	RES 1,D	CBAE	RES 5,(HL)
CB8B	RES 1,E	DDCB05AE	RES 5,(IX+d)
CB8C	RES 1,H	FDCB05AE	RES 5,(IY+d)
CB8D	RES 1,L	CBAF	RES 5,A
CB96	RES 2,(HL)	CBA8	RES 5,B
DDCB0596	RES 2,(IX+d)	CBA9	RES 5,C
FDCB0596	RES 2,(IY+d)	CBAA	RES 5,D

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
CBAB	RES 5,E	CB16	RL (HL)
CBAC	RES 5,H	DDCB0516	RL (IX+d)
CBAD	RES 5,L	FDCB0516	RL (IY+d)
CBB6	RES 6,(HL)	CB17	RL A
DDCB05B6	RES 6,(IX+d)	CB10	RL B
FDCB05B6	RES 6,(IY+d)	CB11	RL C
CBB7	RES 6,A	CB12	RL D
CBB0	RES 6,B	CB13	RL E
CBB1	RES 6,C	CB14	RL H
CBB2	RES 6,D	CB15	RL L
CBB3	RES 6,E	17	RLA
CBB4	RES 6,H	CB06	RLC (HL)
CBB5	RES 6,L	DDCB0506	RLC (IX+d)
CBBE	RES 7,(HL)	FDCB0506	RLC (IY+d)
DDCB05BE	RES 7,(IX+d)	CB07	RLC A
FDCB05BE	RES 7,(IY+d)	CB00	RLC B
CBBF	RES 7,A	CB01	RLC C
CBB8	RES 7,B	CB02	RLC D
CBB9	RES 7,C	CB03	RLC E
CBBA	RES 7,D	CB04	RLC H
CBBB	RES 7,E	CB05	RLC L
CBBC	RES 7,H	07	RLCA
CBBD	RES 7,L	ED6F	RLD
C9	RET	CB1E	RR (HL)
D8	RET C	DDCB051E	RR (IX+d)
F8	RET M	FDCB051E	RR (IY+d)
D0	RET NC	CB1F	RR A
C0	RET NZ	CB18	RR B
F0	RET P	CB19	RR C
E8	RET PE	CB1A	RR D
E0	RET PO	CB1B	RR E
C8	RET Z	CB1C	RR H
ED4D	RETI	CB1D	RR L
ED45	RETN	1F	RRA

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
CB0E	RRC (HL)	ED72	SBC HL,SP
DDCB050E	RRC (IX+d)	37	SCF
FDCB050E	RRC (IY+d)	CBC6	SET 0,(HL)
CB0F	RRC A	DDCB05C6	SET 0,(IX+d)
CB08	RRC B	FDCB05C6	SET 0,(IY+d)
CB09	RRC C	CBC7	SET 0,A
CB0A	RRC D	CBC0	SET 0,B
CB0B	RRC E	CBC1	SET 0,C
CB0C	RRC H	CBC2	SET 0,D
CB0D	RRC L	CBC3	SET 0,E
0F	RRCA	CBC4	SET 0,H
ED67	RRD	CBC5	SET 0,L
C7	RST 0	CBCE	SET 1,(HL)
D7	RST 10H	DDCB05CE	SET 1,(IX+d)
DF	RST 18H	FDCB05CE	SET 1,(IY+d)
E7	RST 20H	CBCF	SET 1,A
EF	RST 28H	CBC8	SET 1,B
F7	RST 30H	CBC9	SET 1,C
FF	RST 38H	CBCA	SET 1,D
CF	RST 8	CBCB	SET 1,E
9E	SBC A,(HL)	CBCC	SET 1,H
DD9E05	SBC A,(IX+d)	CBCD	SET 1,L
FD9E05	SBC A,(IY+d)	CBD6	SET 2,(HL)
9F	SBC A,A	DDCB05D6	SET 2,(IX+d)
98	SBC A,B	FDCB05D6	SET 2,(IY+d)
99	SBC A,C	CBD7	SET 2,A
9A	SBC A,D	CBD0	SET 2,B
9B	SBC A,E	CBD1	SET 2,C
9C	SBC A,H	CBD2	SET 2,D
9D	SBC A,L	CBD3	SET 2,E
DE20	SBC A,N	CBD4	SET 2,H
ED42	SBC HL,BC	CBD5	SET 2,L
ED52	SBC HL,DE	CBD8	SET 3,B
ED62	SBC HL,HL	CBDE	SET 3,(HL)

Strojový kód	Symbolický tvar	Strojový kód	Symbolický tvar
DDCB05DE	SET 3,(IX+d)	CBF2	SET 6,D
FDCB05DE	SET 3,(IY+d)	CBF3	SET 6,E
CBDF	SET 3,A	CBF4	SET 6,H
CBD9	SET 3,C	CBF5	SET 6,L
CBDA	SET 3,D	CBFE	SET 7,(HL)
CBDB	SET 3,E	DDCB05FE	SET 7,(IX+d)
CBDC	SET 3,H	FDCB05FE	SET 7,(IY+d)
CBDD	SET 3,L	CBFF	SET 7,A
CBE6	SET 4,(HL)	CBF8	SET 7,B
DDCB05E6	SET 4,(IX+d)	CBF9	SET 7,C
FDCB05E6	SET 4,(IY+d)	CBFA	SET 7,D
CBE7	SET 4,A	CBFB	SET 7,E
CBE0	SET 4,B	CBFC	SET 7,H
CBE1	SET 4,C	CBFD	SET 7,L
CBE2	SET 4,D	CB26	SLA (HL)
CBE3	SET 4,E	DDCB0526	SLA (IX+d)
CBE4	SET 4,H	FDCB0526	SLA (IY+d)
CBE5	SET 4,L	CB27	SLA A
CBEE	SET 5,(HL)	CB20	SLA B
DDCB05EE	SET 5,(IX+d)	CB21	SLA C
FDCB05EE	SET 5,(IY+d)	CB22	SLA D
CBEF	SET 5,A	CB23	SLA E
CBE8	SET 5,B	CB24	SLA H
CBE9	SET 5,C	CB25	SLA L
CBEA	SET 5,D	CB2E	SRA (HL)
CBEB	SET 5,E	DDCB052E	SRA (IX+d)
CBEC	SET 5,H	FDCB052E	SRA (IY+d)
CBED	SET 5,L	CB2F	SRA A
CBF6	SET 6,(HL)	CB28	SRA B
DDCB05F6	SET 6,(IX+d)	CB29	SRA C
FDCB05F6	SET 6,(IY+d)	CB2A	SRA D
CBF7	SET 6,A	CB2B	SRA E
CBF0	SET 6,B	CB2C	SRA H
CBF1	SET 6,C	CB2D	SRA L

Strojový kod	Symbolický tvar
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

3.4 Rozbor instrukcí souboru Z-80

Přes značný rozsah instrukčního souboru řady Z-80 jsou v kapitole 3.4 rozebrány veškeré instrukce tohoto souboru.

3.4.1 Instrukce výměny registrů

Po vynulování pracuje mikroprocesor Z80-CPU vždy s hlavní skupinou registrů. Přechod k vedlejší skupině registrů zabezpečují instrukce typu EX a EXX.

3.4.1.1 Výměna

EX a, b.	a.←→b.

Instrukce EX zabezpečí vzájemnou výměnu dvou lokalizovaných míst, které jsou určeny operandy a. a b..

Nastavení indikátorů: nezměněno

Kombinace operandů jsou v tabulce 5.

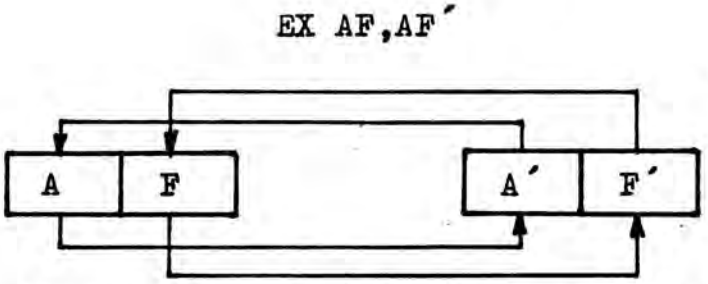
Kombinace operandů v instrukci EX — Tabulka 5

a./b.	HL	AF´	IX	IY
DE	—			
AF		—		
(SP)	—		—	—

Poznámka:

Možné kombinace vzájemných registrových výměn jsou v tabulce 5 označeny pomlčkou.

Příklad: Vzájemná výměna párových registrů AF a AF´(viz obr. 55).



Obr. 55 Vzájemná výměna párových registrů AF a AF´

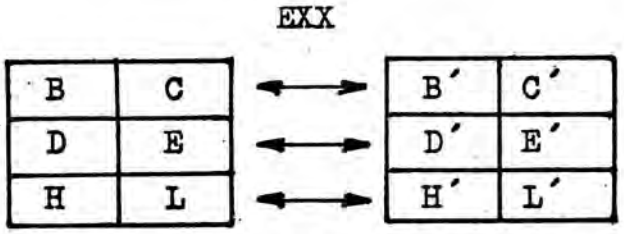
3.4.1.2 Výměna skupiny registrů

EXX	BC↔BC´; DE↔DE´; HL↔HL´

Instrukce EXX zabezpečí vzájemnou výměnu hlavních registrů BC, DE, HL za vedlejší registry BC´, DE´, HL´.

Nastavení indikátorů: nezměněno

Příklad: Vzájemná výměna hlavních a vedlejších registrů mikroprocesoru (viz obr. 56).



Obr. 56 Vzájemná výměna hlavních a vedlejších registrů

3.4.2 Instrukce blokových přenosů

Skupina blokových přenosů je zcela nová a slouží k přenášení bloku dat z lokalizovaného paměťového místa na místo jiné. Zahrnuje 4 instrukce LDI, LDIR, LDD a LDDR.

3.4.2.1 Blokový přenos s inkrementací bez opakování

LDI

(DE) ← (HL); DE ← DE + 1; HL ← HL + 1;
BC ← BC - 1

Instrukce zabezpečí přesun jednoho bytu dat z paměťové buňky, dané obsahem párového registru HL na paměťovou buňku, danou obsahem párového registru DE. Dále se provede inkrementace párového registru DE a HL a dekrementace párového registru BC. V párovém registru BC je možné sledovat celkový počet přenášených bytů.

Nastavení indikátorů:

S	nezměněn
Z	nezměněn
H	nulován
P/V	nastaven, pakliže je BC - 1 ≠ 0
N	nulován
C	nezměněn

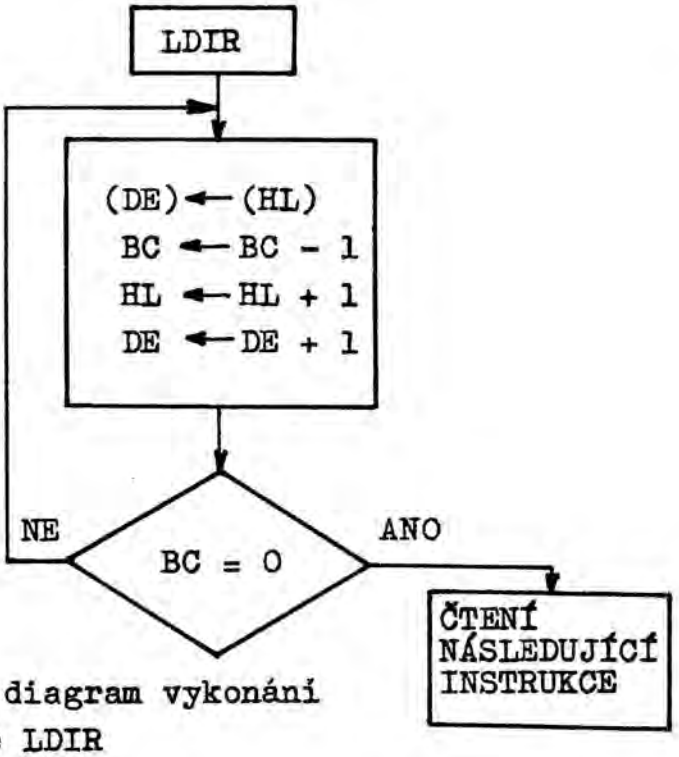
3.4.2.2 Blokový přenos s inkrementací a opakováním

LDIR

(DE) ← (HL); DE ← DE + 1; HL ← HL + 1;
BC ← BC - 1; opakování přenosu až do
stavu BC = 0

Instrukce LDIR vychází z práce instrukce LDI s tím rozdílem, že přenos vykonává opakovaně až do vyčerpání obsahu párového registru BC. LDIR přenese obsah paměťové buňky, jejíž adresa je v párovém registru HL na paměťovou buňku, jejíž adresa je v registrovém páru DE. Instrukce inkrementuje párový registr HL, DE a dekrementuje párový registr BC, který hlídá délku přenášené zóny. Přenosový cyklus opakuje do té doby, dokud je obsah registrového páru BC rozdílný od nuly. Po vyčerpání přenosové délky BC = 0 mikroprocesor pokračuje následující instrukcí v programu. Pakliže je před provedením instrukce nastaven párový registr BC na nulu, instrukce LDIR bude přenášet zónu 64 kB. Princip provedení instrukce je na obr. 57.

Nastavení indikátorů: S nezměněn
Z nezměněn
H nulován
P/V nulován
N nulován
C nezměněn



Obr. 57 Vývojový diagram vykonání instrukce LDIR

3.4.2.3 Blokový přenos s dekrementací bez opakování

LDD (DE) ← (HL); DE ← DE - 1; HL ← HL - 1;
BC ← BC - 1

Instrukce přenáší obsah paměťového místa, jehož adresa je uložena v párovém registru HL na paměťové místo, jehož adresa je v párovém registru DE. Instrukce provádí dekrementaci párového registru HL, DE a BC.

Nastavení indikátorů: S nezměněn
Z nezměněn
H nulován

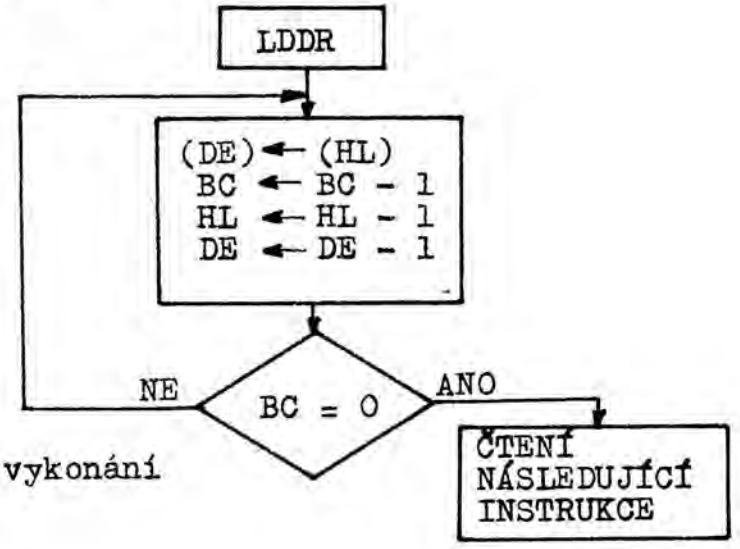
P/V nastaven, pakliže je BC - 1 ≠ 0
N nulován
C nezměněn

3.4.2.4 Blokový přenos s dekrementací a opakováním

LDDR (DE) ← (HL); DE ← DE - 1; HL ← HL - 1; BC ← BC - 1; opakování přenosu až do stavu BC = 0

Instrukce LDDR vychází z práce instrukce LDD, přenos je ale opakovaný až do vyčerpání obsahu párového registru BC. LDDR provádí přenos obsahu paměťové buňky, jejíž adresu obsahuje párový registr HL na paměťovou buňku, jejíž adresu obsahuje párový registr DE. Instrukce dále provádí dekrementaci párového registru DE, HL a BC. V párovém registru BC je registrována přenášená délka zóny dat. Přenos se opakuje do té doby, dokud obsah párového registru BC není roven nule. Pakliže obsah párového registru BC je roven nule, mikroprocesor začíná pracovat na následující instrukci programu. Princip vykonání instrukce je na obr. 58.

Nastavení indikátorů: S nezměněn
Z nezměněn
H nulován
P/V nulován
N nulován
C nezměněn



Obr. 58 Vývojový diagram vykonání instrukce LDDR

3.4.3 Instrukce vyhledávání v blocích dat

Čtyři instrukce této skupiny vyhledávají byte v paměti, který se svým obsahem shoduje s akumulačním registrem mikroprocesoru. Jedná se o instrukce CPI, CPIR, CPD, CPDR.

3.4.3.1 Blokové vyhledávání s inkrementací bez opakování

CPI — A - (HL); HL ← HL + 1; BC ← BC - 1

Instrukce CPI zabezpečí odečtení obsahu paměťové buňky, jejíž adresa je obsažena v párovém registru HL od registru A. Při shodě obsahů paměťového místa a akumulátoru je nastaven indikátorový bit Z = 1. Při práci instrukce se dále provádí inkrementace párového registru HL a dekrementace párového registru BC.

Nastavení indikátorů:

- S — nastaven, pakliže je výsledek záporný
- Z — nastaven, pakliže je výsledek nula, tj. A = (HL)
- H — nastaven, pakliže je přenos mezi 3. a 4. bitem
- P/V — nastaven, pakliže je BC - 1 ≠ 0
- N — nastaven
- C — nezměněn

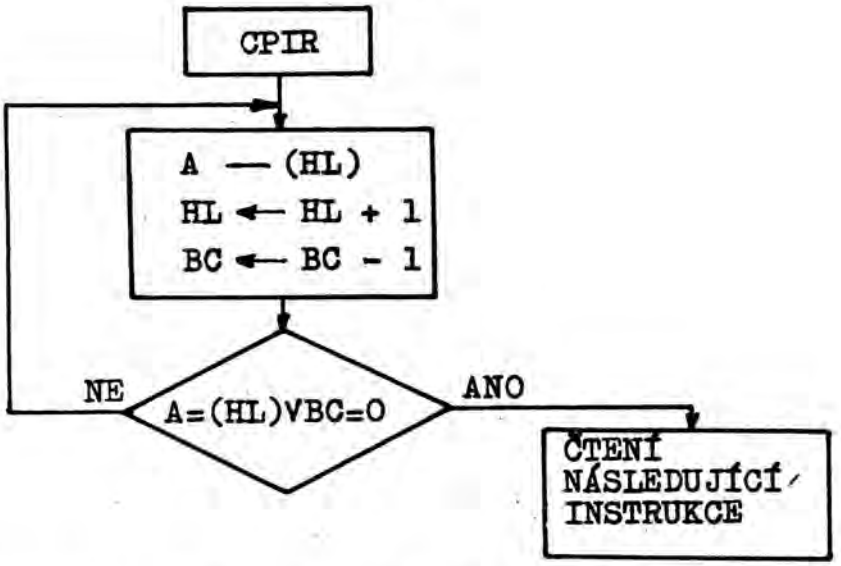
3.4.3.2 Blokové vyhledávání s inkrementací a opakováním

CPIR — A - (HL); HL ← HL + 1; BC ← BC - 1;
opakování vyhledávání až do stavu
BC = 0 ∨ A = (HL)

Instrukce CPIR pracuje na principu instrukce CPI, vyhledávání shodného obsahu je ale opakované až do stavu BC = 0 ∨ A=(HL). CPIR zabezpečí odečtení obsahu paměťové buňky, jejíž adresu obsahuje párový registr HL od obsahu akumulátoru A. Instrukce dále inkrementuje registrový pár HL a dekrementuje registrový pár BC. Pakliže je obsah párového registru BC různý od nuly a zároveň výsledek posledního odečtení není rovný nule, činnost se

cyklicky opakuje. Jestliže je obsah párového registru BC roven nule nebo je-li výsledek odečtení roven nule, pokračuje se následující instrukcí v programu. Provedení instrukce je zachyceno na vývojovém diagramu (viz obr. 59).

Nastavení indikátorů:
- S nastaven, pakliže je výsledek záporný
- Z nastaven, pakliže je výsledek nula, tj. A = (HL)
- H nastaven, pakliže je přenos mezi 3. a 4. bitem
- P/V nastaven, pakliže je BC - 1 ≠ 0
- N nastaven
- C nezměněn



Obr. 59 Vývojový diagram provedení instrukce CPIR

3.4.3.3 Blokové vyhledávání s dekrementací bez opakování

CPD A - (HL); HL ← HL - 1; BC ← BC - 1

Instrukce CPD zabezpečí odečtení paměťové buňky, jejíž adresa je v párovém registru HL od obsahu akumulačního registru A.

Při shodě obsahů paměťového místa a akumulátoru je nastaven indikátorový bit Z = 1. Instrukce dále provádí dekrementaci párového registru HL a BC.

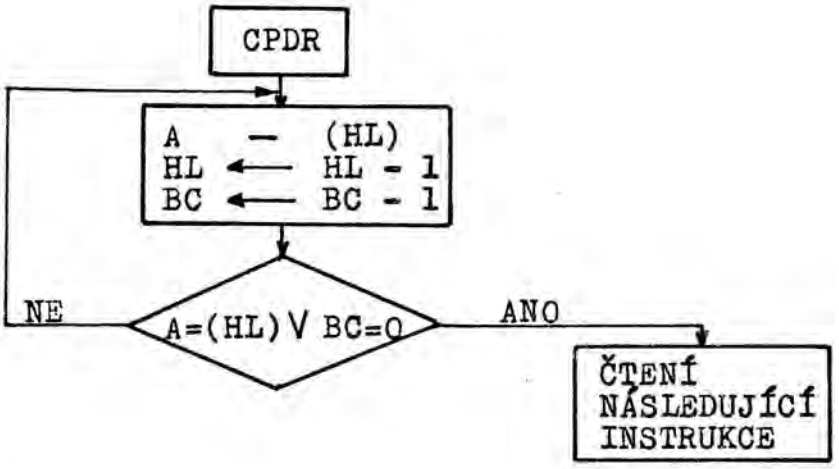
Nastavení indikátorů:
- S nastaven, pakliže je výsledek záporný
- Z nastaven, pakliže je výsledek nula, tj. A = (HL)
- H nastaven, pakliže je přenos mezi 3. a 4. bitem
- P/V nastaven, pakliže je BC - 1 ≠ 0
- N nastaven
- C nezměněn

3.4.3.4 Blokové vyhledávání s dekrementací a opakováním

CPDR — A - (HL); HL ← HL - 1; BC ← BC - 1;
opakování vyhledávání až do stavu
BC = 0 ∨ A = (HL)

Instrukce CPDR pracuje stejně jako instrukce CPD s tím rozdílem, že vyhledávání je opakované až do stavu BC = 0 ∨ A = (HL). CPDR zabezpečí odečtení paměťové buňky, jejíž adresu obsahuje párový registr HL od akumulačního registru A. Další popis činnosti je zcela shodný s instrukcí CPIR s tím rozdílem, že párový registr HL je dekrementován. Princip provedení instrukce je na obr. 60.

Nastavení indikátorů:
- S nastaven, pakliže je výsledek záporný
- Z nastaven, pakliže je výsledek nula, tj. A = (HL)
- H nastaven, pakliže je přenos mezi 3. a 4. bitem
- P/V nastaven, pakliže je BC - 1 ≠ 0
- N nastaven
- C nezměněn



Obr. 60 Vývojový diagram provedení instrukce CPDR

3.4.4 Instrukce pro přesun dat

Instrukce pro přesun dat provádí přesuny dat mezi registry mikroprocesoru, mezi registry a pamětí, případně mezi registry a zásobníkem.

3.4.4.1 Přesun dat

LD a., b.	a. ← b.

Instrukce LD zabezpečí přesun jednoho nebo dvou bytů dat z lokalizovaného místa operandem b. na cílové místo lokalizované operandem a.. Instrukce, které přesouvají do paměti informaci 2 byte, pracují tak, že nižší byte šestnáctibitového slova ukládají na nižší adresu a vyšší byte šestnáctibitového slova na vyšší adresu paměti.

Nastavení indikátoru: nezměněno
Kombinace operandu jsou v tabulce 6.

3.4.4.2 Čtení zásobníkové paměti

POP a.	$a._L \leftarrow (SP)$; $a._H \leftarrow (SP+1)$; $SP \leftarrow SP+2$

a. ← b.	(BC)	(DE)	(HL)	(IX+d)	(IY+d)	(nn)	A	B	C	D	E	H	L	I	R	n	IX	IY	SP	BC	DE	HL	nn
(BC)							–																
(DE)							–																
(HL)							–	–	–	–	–	–	–			–							
(IX+d)							–	–	–	–	–	–	–			–							
(IY+d)							–	–	–	–	–	–	–			–							
(nn)							–										–	–	–	–	–	–	
A	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–							
B			–	–	–		–	–	–	–	–	–	–			–							
C			–	–	–		–	–	–	–	–	–	–			–							
D			–	–	–		–	–	–	–	–	–	–			–							
E			–	–	–		–	–	–	–	–	–	–			–							
H			–	–	–		–	–	–	–	–	–	–			–							
L			–	–	–		–	–	–	–	–	–	–			–							
I							–																
R							–																
IX						–																	–
IY						–																	–
SP						–											–	–				–	–
BC						–																	–
DE						–																	–
HL						–																	–

Možné kombinace přesunů v LD instrukci jsou označeny pomlčkou. Tabulka 6

Instrukce POP postupně přesouvá dva byte z vrcholu zásobníku do index registru nebo do registrového páru. POP nejprve přenese obsah zásobníku (nižší byte), který je lokalizován SP registrem, pak inkrementuje SP a zase přenáší obsah zásobníku (vyšší byte). Ukazatel zásobníkové paměti SP je opět inkrementován, aby ukazoval po provedení instrukce na nový vrchol zásobníku.

Kombinace operandů: IX, IY, AF, BC, DE, HL

Nastavení indikátorů: nezměněno.

3.4.4.3 Zápis do zásobníkové paměti

PUSH a. (SP-1)←a.$_H$; (SP-2)←a.$_L$; SP←SP-2

Tato instrukce provádí postupně přesun index registru nebo párového registru na vrchol zásobníkové paměti. Provedení instrukce se zahajuje dekrementací ukazatele zásobníkové paměti, následuje zápis vyššího bytu do zásobníku, opět se dekrementuje SP a zapisuje se nižší byte do zásobníkové paměti. Registr SP po provedení instrukce adresuje poslední zapisovaný byte zásobníkové paměti.

Kombinace operandů: IX, IY, AF, BC, DE, HL

Nastavení indikátorů nezměněno.

3.4.5 Aritmetické instrukce

Aritmetické instrukce pracují s osmibitovým nebo šestnáctibitovým registrem, registrovým párem a paměťovým místem. Instrukce DAA koriguje výsledek aritmetické operace (sčítání, odčítání) do BCD kódu.

3.4.5.1 Aritmetické sčítání

ADD a.,b. a.←a.+b.

Operace ADD sčítá dva operandy a. a b.. Výsledek provedeného součtu ukládá na místo operandu a..

Kombinace operandů jsou v tabulce 7.

Nastavení indikátorů při osmibitové operaci ADD:

S nastaven, pakliže je výsledek záporný
Z nastaven, pakliže je výsledek nula
H nastaven, pakliže je přenos mezi 3. a 4. bitem
P/V nastaven v případě přetečení
N nulován
C nastaven v případě přenosu ze znaménkového bitu

Nastavení indikátorů při šestnáctibitové operaci ADD:

S nezměněn
Z nezměněn
H nastaven, pakliže je přenos z 11. bitu
P/V nezměněn
N nulován
C nastaven v případě přenosu ze znaménkového bitu

3.4.5.2 Aritmetické sčítání s přenosem

ADC a.,b.	a.← a.+b.+CY

Instrukce ADC sečte operand a. s operandem b. a s indikátorovým bitem CY. Výsledek operace je uložen na místě operandu a..

Varianty operandů,použité v operaci ADC,jsou v tabulce 8.

Nastavení indikátorů:

S nastaven, pakliže je výsledek záporný
Z nastaven, pakliže je výsledek nula
H nastaven, jestliže došlo k přenosu mezi 3. a 4. bitem u osmibitové operace, anebo mezi 11. a 12. bitem u šestnáctibitové operace
P/V nastaven v případě přetečení
N nulován
C nastaven v případě přenosu ze znaménkového bitu

Kombinace operandů instrukce ADD Tabulka 7

a./b.	IX	IY	DE	HL	BC	SP	(HL)	(IX+d)	(IY+d)	A	B	C	D	E	H	L	n
A							—	—	—	—	—	—	—	—	—	—	—
HL			—	—	—	—											
IX	—		—		—	—											
IY		—	—		—	—											

Kombinace operandů instrukce ADC, SBC Tabulka 8

a./b.	n	(HL)	(IX+d)	(IY+d)	A	B	C	D	E	H	L	BC	DE	HL	SP
A	—	—	—	—	—	—	—	—	—	—	—				
HL												—	—	—	—

3.4.5.3 Aritmetické odčítání

SUB a. A←A-a.

Instrukce odčítá od akumulátoru hodnotu operandu a.. Výsledek operace se ukládá opět do akumulátoru.

Operandové varianty: A, B, C, D, E, H, L, n, (HL), (IX+d), (IY+d)

Nastavení indikátorů:
- S nastaven, jestliže je výsledek záporný
- Z nastaven, jestliže je výsledek nula
- H nastaven, když došlo k přenosu mezi 3. a 4. bitem
- P/V nastaven v případě přetečení
- N nastaven
- C nastaven při přenosu ze znaménkového bitu

3.4.5.4 Aritmetické odčítání s přenosem

SBC a.,b. a.←a.-b.-CY

Instrukce odčítá operand b. a indikátorový bit CY od operandu a.. Výsledek provedení operace se uloží na místo operandu a..

Operandové varianty jsou v tabulce 8.

Nastavení indikátorů:
- S nastaven, jestliže je výsledek záporný
- Z nastaven, jestliže je výsledek nula
- H nastaven, když nastal přenos mezi 3. a 4. bitem u osmibitové operace, anebo mezi 11. a 12. bitem u šestnáctibitové operace

P/V nastaven v případě přetečení
N nastaven
C nastaven při přenosu ze znaménkového bitu

3.4.5.5 Dekadická úprava akumulátoru

DAA

Instrukce DAA zabezpečí opravu výsledku uloženého v akumulátoru A po provedené aritmetické operaci v binárním kódu tak, aby korigovaný výsledek odpovídal operaci v BCD kódu. Korekce je prováděna na základě zjištění stavu indikátorových bitů C a H před provedením instrukce DAA a dále je testována hodnota nižších a vyšších řádů akumulátoru před provedením instrukce DAA. Způsob korigování výsledku na úrovni akumulátoru je v tabulce 9.

Nastavení indikátorů:
S nastaven, jestliže je znaménkový bit akumulátoru 1
Z nastaven, jestliže je obsah akumulátoru nula
H nedefinován
P/V nastaven, jestliže akumulátor obsahuje po operaci sudou paritu
N nezměněn
C viz tabulka 9

Příklad: Sečti dvě dekadická čísla instrukcí ADD, např. 17D a 54D.

```
0001  0111  17D
0101  0100  54D
0110  1011  6BH
```

Výsledek vychází hexadecimální, korekci na dekadický, tj. v BCD kódu, provede instrukce DAA.

```
0110  1011  6BH
0000  0110  06H
0111  0001  71D
```

Způsob provádění korekce výsledku instrukcí DAA — Tabulka 9

Instrukce	C před DAA	Obsah A4÷A7	H před DAA	Obsah A0÷A3	Přičtená hodnota k A instrukcí DAA	C po DAA
ADD ADC INC	0	0 ÷ 9	0	0 ÷ 9	0 0	0
	0	0 ÷ 8	0	A ÷ F	0 6	0
	0	0 ÷ 9	1	0 ÷ 3	0 6	0
	0	A ÷ F	0	0 ÷ 9	6 0	1
	0	9 ÷ F	0	A ÷ F	6 6	1
	0	A ÷ F	1	0 ÷ 3	6 6	1
	1	0 ÷ 2	0	0 ÷ 9	6 0	1
	1	0 ÷ 2	0	A ÷ F	6 6	1
	1	0 ÷ 3	1	0 ÷ 3	6 6	1
SUB SBC DEC NEG	0	0 ÷ 9	0	0 ÷ 9	0 0	0
	0	0 ÷ 8	1	6 ÷ F	F A	0
	1	7 ÷ F	0	0 ÷ 9	A 0	1
	1	6 ÷ F	1	6 ÷ F	9 A	1

3.4.6 Logické instrukce

Instrukce logického součinu, logického součtu, exkluzivního součtu a srovnání pracují s osmibitovým registrem nebo paměťovým místem. Operace inkrementace a dekrementace lze provádět nejen na osmibitovém registru a paměťovém místě, ale i na šestnáctibitovém registru. Instrukce negace a komplementu lze řešit výhradně na akumulátoru.

3.4.6.1 Logický součin

`AND a.` $A \leftarrow A \wedge a.$

Instrukce AND zabezpečí logický součin sobě odpovídajících bitů obsahu akumulátoru A s operandem a.. Výsledek provedení operace se uloží do akumulátoru.

Kombinace operandu a.: A, B, C, D, E, H, L, n, (HL), (IX+d), (IY+d)

Nastavení indikátorů:

S	nastaven, jestliže je výsledek záporný
Z	nastaven, jestliže je výsledek nulový
H	nastaven
P/V	nastaven v případě sudé parity
N	nulován
C	nulován

Příklad: Zajistěte snulování bitů A0 ÷ A3 akumulátoru, bity A4 ÷ A7 mají zůstat nezměněny. Obsah akumulátoru je 47H.

```
AND F0
```

Po provedení instrukce AND F0 dojde k naplnění akumulátoru hodnotou 40H.

`0100 0111` $\wedge$ `1111 0000` = `0100 0000`

3.4.6.2 Logický součet

OR a. $A \leftarrow A \vee a.$

Instrukce OR řeší logický součet sobě odpovídajících bitů akumulátoru A a operandu a.. Výsledek provedené operace se uloží do akumulátoru.

Kombinace operandu a.: A, B, C, D, E, H, L, n, (HL), (IX+d), (IY+d)

Nastavení indikátorů:
- S nastaven, jestliže je výsledek záporný
- Z nastaven, jestliže je výsledek nulový
- H nastaven
- P/V nastaven v případě sudé parity
- N nulován
- C nulován

3.4.6.3 Exkluzivní součet

XOR a. $A \leftarrow A \oplus a.$

Instrukce XOR zabezpečí exkluzivní součet sobě odpovídajících bitů akumulátoru A a operandu a.. Výsledek exkluzivního součtu (součet modulo 2) se uloží do akumulátoru.

Kombinace operandu a.: A, B, C, D, E, H, L, n, (HL), (IX+d), (IY+d)

Nastavení indikátorů:
- S nastaven, jestliže je výsledek záporný
- Z nastaven, jestliže je výsledek nulový
- H nastaven
- P/V nastaven v případě sudé parity
- N nulován
- C nulován

Příklad: Proveďte exkluzivní součet XOR (HL), jestliže A = 91H, HL = 5 400H a obsah paměti lokalizované párovým registrem HL je 30H.

$$\boxed{1001\ 0001} \oplus \boxed{0111\ 0000} = \boxed{1010\ 0001}$$

Výsledek v akumulátoru po provedené operaci je A1H a z indikátorových bitů budedo jedničky nastaven pouze bit S.

3.4.6.4 Srovnání

CP a.	A - a.

Instrukce srovnání CP provádí odečtení obsahu operandu a. od akumulátoru A s tím, že výsledek provedení operace nepřepisuje žádný z registrů mikroprocesoru. Proto nás instrukce informuje o svém provedení pomocí indikátorového registru.

Kombinace operandu a.: A, B, C, D, E, H, L, n, (HL), (IX+d), (IY+d)

Nastavení indikátorů:

S	nastaven, jestliže je výsledek záporný
Z	nastaven, jestliže je výsledek nulový
H	nastaven, jestliže je přenos mezi 3. a 4. bitem
P/V	nastaven v případě přetečení
N	nastaven
C	nastaven při přenosu ze znaménkového bitu

3.4.6.5 Inkrementace

INC a.	a. ← a. + 1

Instrukce přičítá jedničku k a. operandu. Výsledek operace se ukládá na místo operandu a..

Kombinace operandu: A, B, C, D, E, H, L, (HL), (IX+d), (IY+d), IX, IY, BC, DE, HL, SP

Nastavení indikátorů při osmibitové operaci:

S nastaven, jestliže je výsledek záporný
Z nastaven, jestliže je výsledek nulový
H nastaven, jestliže je přenos mezi 3. a 4. bitem
P/V nastaven při přetečení
N nulován
C nezměněn

Nastavení indikátorů při šestnáctibitové operaci:

původní stav indikátorových bitů je nezměněn

3.4.6.6 Dekrementace

DEC a. a. ← a. - 1

Instrukce odčítá jedničku od operandu a., výsledek operace se ukládá na místo operandu a..

Kombinace operandu: A, B, C, D, E, H, L, (HL), (IX+d), (IY+d), IX, IY, BC, DE, HL, SP

Nastavení indikátorů při osmibitové operaci:

S nastaven, jestliže je výsledek záporný
Z nastaven, jestliže je výsledek nulový
H nastaven, jestliže je přenos mezi 3. a 4. bitem
P/V nastaven při přetečení
N nastaven
C nezměněn

Nastavení indikátorů při šestnáctibitové operaci:

původní stav indikátorových bitů je nezměněn

3.4.6.7 Komplement

CPL $A \leftarrow \overline{A}$

Tato instrukce pouze provádí jedničkový doplněk akumulátoru A.

Nastavení indikátorů:
- S nezměněn
- Z nezměněn
- H nastaven
- P/V nezměněn
- N nastaven
- C nezměněn

Příklad: Původní obsah akumulátoru A je 0110 0011. Po provedení instrukce CPL bude jeho obsah 1001 1100.

3.4.6.8 Negace

NEG $A \leftarrow 0 - A$

Instrukce NEG provádí odečtení obsahu akumulátoru od nuly.

Nastavení indikátorů:
- S nastaven, jestliže je výsledek záporný
- Z nastaven, jestliže je výsledek nulový
- H nastaven, jestliže je přenos mezi 3. a 4. bitem
- P/V nastaven, jestliže je obsah akumulátoru před operací 80H
- N nastaven
- C nastaven, jestliže obsah akumulátoru před operací není nula

Příklad: Obsah akumulátoru je 0011 1010. Po provedení instrukce NEG je obsah akumulátoru 1100 0110.

3.4.7 Řídicí instrukce

Řídicí instrukce řeší nastavení jednoho ze tří přerušovacích režimů MODE 0,1,2, zastavení práce programu, povolení a blokování zpracování přerušení. Dále umožňují pracovat s indikátorovým bitem CY a vkládat do programu prázdnou operaci.

3.4.7.1 Programové nastavení režimu přerušení - MODE 0, 1, 2

IM 0

IM 1

IM 2

Souhrnně se instrukce IM 0 ÷ IM 2 používají k programovému nastavení režimu maskovatelných přerušení 0 ÷ 2. Mikroprocesor po vykonání instrukce IM 0 ÷ IM 2 je nastaven do jednoho z přerušovacích režimů MODE 0 ÷ 2.

3.4.7.2 Zastavení

HALT

Instrukce HALT zastavuje činnost mikroprocesoru a kromě toho zajišťuje systematické vkládání instrukce NOP až do přijetí přerušení, vynulování Z80-CPU nebo žádosti o DMA přenos. Vložená instrukce NOP zajišťuje obnovu dynamických pamětí.

Nastavení indikátorů: nezměněno

3.4.7.3 Povolení přerušení

EI IFF ← 1

Instrukce EI nastavuje klopné obvody povolení přerušení IFF1 a IFF2. Maskovatelný požadavek přerušení může být akceptován po dokončení následující instrukce po instrukci EI.

Nastavení indikátorů: nezměněno

3.4.7.4 Blokování přerušení

DI IFF ← 0

Instrukce DI naopak vynuluje klopné obvody povolení přerušení IFF1 a IFF2, a tím se zablokuje možnost přijetí maskovatelného přerušení (vstupní signál $\overline{INT}$). Požadavek maskovatelného přerušení lze přijmout až během následující instrukce po instrukci EI.

3.4.7.5 Inverze indikátorového bitu C

CCF CY ← $\overline{CY}$

Instrukce zabezpečí inverzi indikátorového bitu C.

Nastavení indikátorů:

S	nezměněn
Z	nezměněn
H	přebírá předchozí stav C bitu
P/V	nezměněn
N	nulován
C	invertován

3.4.7.6 Nastavení indikátorového bitu C

SCF CY ← 1

Instrukce SCF nastavuje do jedničky indikátorový bit C.

Nastavení indikátorů:

S	nezměněn
Z	nezměněn
H	nulován
P/V	nezměněn
N	nulován
C	nastaven

3.4.7.7 Prázdná operace

NOP

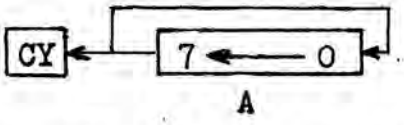
Instrukce NOP nevykonává žádnou činnost. V programu prezentuje pouze čas potřebný na své načtení z paměti do mikroprocesoru.

3.4.8 Instrukce rotací a posuvů

Instrukce umožňují posuv nebo rotaci s jednotlivými bity osmibitového registru nebo paměťového místa. Posuvy i rotace lze řešit vlevo i vpravo bez účasti nebo s účastí indikátorového bitu C. Rotační operace lze uplatnit také na skupinách čtyř bitů osmibitových informací akumulátoru a paměťového místa.

3.4.8.1 Rotace akumulátoru vlevo a do CY

RLCA

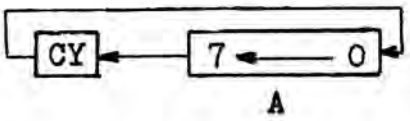


Instrukce RLCA provádí rotaci akumulátoru A směrem doleva a do CY. Přitom se přenese nejvyšší bit akumulátoru do uvolněného nultého bitu a zároveň do indikátorového bitu CY.

Nastavení indikátorů:
- S nezměněn
- Z nezměněn
- H nulován
- P/V nezměněn
- N nulován
- C obsahuje nejvyšší bit akumulátoru

3.4.8.2 Rotace akumulátoru vlevo přes CY

RLA

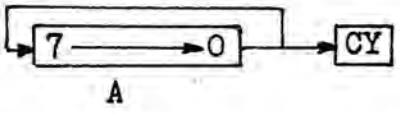


Rotace akumulátoru u instrukce RLA probíhá vlevo přes indikátorový bit CY. Nejvyšší bit akumulátoru se přenese do CY a původní obsah CY se přesune do nultého bitu akumulátoru.

Nastavení indikátorů:
- S nezměněn
- Z nezměněn
- H nulován
- P/V nezměněn
- N nulován
- C obsahuje nejvyšší bit akumulátoru

3.4.8.3 Rotace akumulátoru vpravo a do CY

RRCA

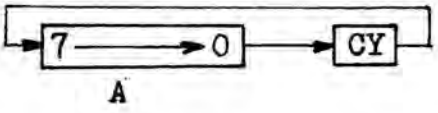


Rotace akumulátoru probíhá doprava a do CY. Nultý bit se přesune jednak do nejvyššího bitu akumulátoru a jednak do indikátorového bitu CY.

Nastavení indikátorů:
S nezměněn
Z nezměněn
H nulován
P/V nezměněn
N nulován
C obsahuje nultý bit akumulátoru

3.4.8.4 Rotace akumulátoru vpravo přes CY

RRA

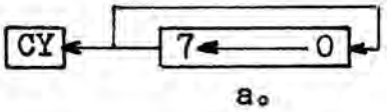


Instrukce RRA řeší rotaci akumulátoru vpravo přes indikátorový bit CY. Nultý bit akumulátoru se přesouvá do CY a původní obsah CY se přesouvá do nejvyššího bitu akumulátoru.

Nastavení indikátorů:
S nezměněn
Z nezměněn
H nulován
P/V nezměněn
N nulován
C obsahuje nultý bit akumulátoru

3.4.8.5 Rotace operandu vlevo a do CY

RLC a.



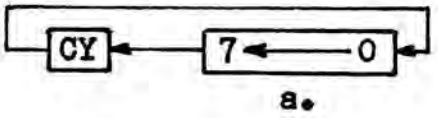
Instrukce RLC provádí rotaci operandu a. vlevo a do CY. Nejvyšší bit operandu se přenese do nultého bitu a zároveň do indikátoru CY.

Kombinace operandu a.: A, B, C, D, E, H, L, (HL), (IX+d), (IY+d)

Nastavení indikátorů:
- S nastaven, jestliže je výsledek záporný
- Z nastaven, jestliže je výsledek nulový
- H nulován
- P/V nastaven v případě sudé parity
- N nulován
- C obsahuje nejvyšší bit operandu

3.4.8.6 Rotace operandu vlevo přes CY

RL a.



a.

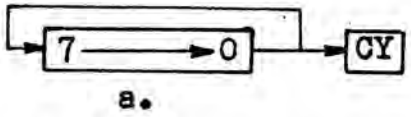
Instrukce RL rotuje s operandem a. vlevo přes indikátorový bit CY. Přitom se hodnota nejvyššího bitu operandu přesune do indikátoru CY a původní stav CY se přesune do nultého bitu operandu.

Kombinace operandu a.: A, B, C, D, E, H, L, (HL), (IX+d), (IY+d)

Nastavení indikátorů:
- S nastaven, jestliže je výsledek záporný
- Z nastaven, jestliže je výsledek nulový
- H nulován
- P/V nastaven v případě sudé parity
- N nulován
- C obsahuje nejvyšší bit operandu

3.4.8.7 Rotace operandu vpravo a do CY

RRC a.



a.

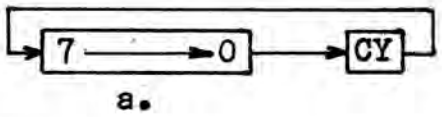
Instrukce RRC rotuje s operandem a. vpravo a do CY. Nultý bit operandu se přesune do nejvyššího bitu a zároveň do indikátorového bitu CY.

Kombinace operandu a.: A, B, C, D, E, H, L, (HL), (IX+d), (IY+d)

Nastavení indikátorů:
- S nastaven, jestliže je výsledek záporný
- Z nastaven, jestliže je výsledek nulový
- H nulován
- P/V nastaven v případě sudé parity
- N nulován
- C obsahuje nultý bit operandu

3.4.8.8 Rotace operandu vpravo přes CY

RR a.



a.

Instrukce RR rotuje s operandem a. vpravo přes indikátorový bit CY. Přitom se hodnota nultého bitu přesune do indikátorového bitu CY a původní stav CY se přesune do nejvyššího bitu operandu.

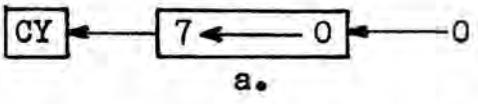
Kombinace operandu a.: A, B, C, D, E, H, L, (HL), (IX+d), (IY+d)

Nastavení indikátorů:
- S nastaven, jestliže je výsledek záporný
- Z nastaven, jestliže je výsledek nulový

H nulován
P/V nastaven v případě sudé parity
N nulován
C obsahuje nultý bit operandu

3.4.8.9 Aritmetický posuv vlevo

SLA a.



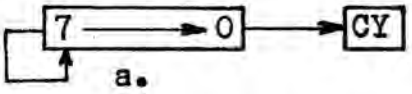
Aritmetický posuv operandu u instrukce SLA probíhá vlevo. Nejvyšším bitem operandu se naplní indikátorový bit CY a do nultého bitu operandu se přenese nula.

Kombinace operandu a.: A, B, C, D, E, H, L, (HL), (IX+d), (IY+d)

Nastavení indikátorů: S nastaven, jestliže je výsledek záporný
Z nastaven, jestliže je výsledek nulový
H nulován
P/V nastaven v případě sudé parity
N nulován
C obsahuje nejvyšší bit operandu

3.4.8.10 Aritmetický posuv vpravo

SRA a.



Aritmetický posuv operandu u instrukce SRA probíhá vpravo. Nultý bit operandu naplní indikátorový bit CY a stav nejvyššího bitu operandu se zachová.

Kombinace operandu a.: A, B, C, D, E, H, L, (HL), (IX+d), (IY+d)

Nastavení indikátorů: S nastaven, jestliže je výsledek záporný

Z nastaven, jestliže je výsledek nulový
H nulován
P/V nastaven v případě sudé parity
N nulován
C obsahuje nultý bit operandu

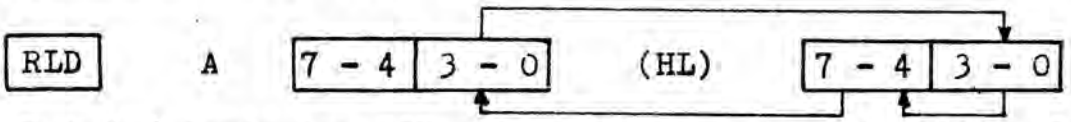
3.4.8.11 Posuv vpravo

Instrukce SRL provádí posuv operandu a. vpravo. Nultý bit operandu naplní indikátorový bit CY a nejvyšší bit operandu je naplněn nulou.

Kombinace operandu a.: A, B, C, D, E, H, L, (HL), (IX+d), (IY+d)

Nastavení indikátorů:
S nastaven, jestliže je výsledek záporný
Z nastaven, jestliže je výsledek nulový
N nulován
P/V nastaven v případě sudé parity
N nulován
C obsahuje nultý bit operandu

3.4.8.12 Rotace čtveřic bitů vlevo



Rotační instrukce RLD zabezpečí rotaci mezi nižší polovinou byte akumulátoru A a obsahem paměťové buňky, jejíž adresa je obsažena v párovém registru HL. Nižší polovina akumulátoru je přenesena do nižší poloviny bytu paměťové buňky, nižší polovina bytu paměťové buňky je přenesena do vyšší poloviny bytu buňky a vyšší polovina bytu paměťové buňky je přenesena do nižší poloviny bytu akumulátoru A.

Nastavení indikátorů: S nastaven, jestliže je obsah akumulátoru záporný
Z nastaven, jestliže je obsah akumulátoru nula
H nulován
P/V nastaven, jestliže má akumulátor sudou paritu
N nulován
C nezměněn

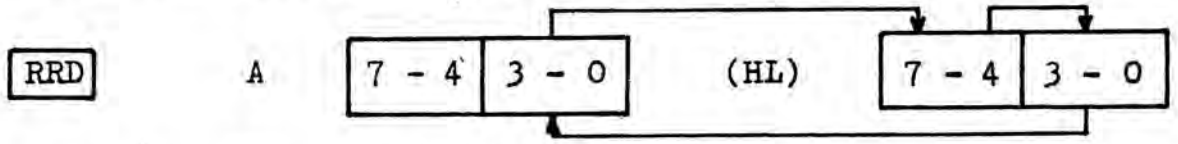
Příklad:

A	24	HL	0500	0500	40

Následuje stav akumulátoru A, párového registru HL a paměťového místa určeného registrem HL po provedení instrukce RLD.

A	24	HL	0500	0500	04

3.4.8.13 Rotace čtveřic bitů vpravo



Instrukce RRD pracuje stejně jako předchozí,s tím rozdílem, že rotace probíhá opačným směrem. RRD zabezpečí rotaci mezi nižší polovinou bytu akumulátoru A a obsahem paměťové buňky, jejíž adresa je obsažena v párovém registru HL. Nižší polovina bytu akumulátoru je přenesena do vyšší poloviny bytu paměťového místa, vyšší polovina bytu paměťového místa je přesunuta do nižší poloviny bytu paměťového místa a nižší polovina bytu paměťového místa je zanesena do nižší poloviny bytu akumulátoru.

Nastavení indikátorů: S nastaven, jestliže je obsah akumulátoru záporný
Z nastaven, jestliže je obsah akumulátoru nula

H nulován
P/V nastaven, jestliže má akumulátor sudou paritu
N nulován
C nezměněn

Příklad: V příkladu jsou použity stejné obsahy akumulátoru a paměťového místa jako v kapitole 3.4.8.12.

A	24	HL	0500	0500	40

Následuje stav akumulátoru A, párového registru HL a paměťového místa určeného registrem HL po provedení instrukce RRD.

A	20	HL	0500	0500	44

3.4.9 Instrukce bitových operací

Instrukce bitových operací umožňují nastavovat, nulovat nebo testovat jakýkoliv bit registru nebo paměti, lokalizovaný obsahem párového registru HL nebo adresovaný indexovaně.

3.4.9.1 Nastavení bitu

SET b,a. $\quad a._b \leftarrow 1$

Instrukce SET zabezpečí nastavení bitu b operandu a.. Bit b je možné lokalizovat od nejnižšího po nejvyšší bit, tj. 0÷7 a operand a. jako (HL), (IX+d), (IY+d), A, B, C, D, E, H, L.

Nastavení indikátorů: nezměněno

3.4.9.2 Nulování bitu

RES b,a. $\quad a._b \leftarrow 0$

Instrukce RES zabezpečí nulování bitu b operandu a.. Opět platí veškeré možné kombinace pro b = 0 ÷ 7 a operand a., tj. (HL), (IX+d), (IY+d), A, B, C, D, E, H, L.

Nastavení indikátoru: nezměněno

3.4.9.3 Testování bitu

BIT b,a.	$Z \leftarrow \overline{a.}_b$

Instrukce BIT zabezpečí přenos komplementu bitu b operandu a. do indikátorového bitu Z. Zase platí veškeré možné kombinace pro b = 0 ÷ 7 a operand a., tj. (HL), (IX+d), (IY+d), A, B, C, D, E, H, L.

Nastavení indikátorů:

S	nedefinován
Z	obsahuje komplement testovaného bitu
H	nastaven
P/V	nedefinován
N	nulován
C	nezměněn

3.4.10 Skokové instrukce

Skokové instrukce provádí změnu obsahu programového čítače PC mikroprocesoru, a tím zabezpečí skok v programu. Skokové instrukce lze rozdělit na nepodmíněné a podmíněné, tj. podmíněné stavu indikátorového registru.

3.4.10.1 Nepodmíněný skok

JP a.	PC ← a.

Instrukce nepodmíněného skoku JP provede skok na paměťové místo definované operandem a. tak, že přenese operand do šestnáctibitového registru PC.

Kombinace operandu a.: nn, (HL), (IX), (IY)

Nastavení indikátorů: nezměněno

Poznámka

Šestnáctibitový registr (HL), (IX), (IY), uzavřený v kulatých závorkách, definuje přímo svým obsahem místo pokračování programu.

3.4.10.2 Podmíněný skok

JP cc, nn

jestliže je podmínka cc splněna, pak se provede skok PC←nn, při nesplněné podmínce cc program pokračuje následující instrukcí.

Indikátorové podmínky cc jsou shrnuty v tabulce 10.

Nastavení indikátorů: nezměněno

Indikátorové podmínky cc podmíněných instrukcí typu JP, CALL, RET — Tabulka 10

cc	Indikátor	Stav indikátoru	Stev splňující podmínku
NZ	Z	nulován	výsledek nenulový
Z	Z	nastaven	výsledek nulový
NC	C	nulován	bez přenosu
C	C	nastaven	s přenosem
PO	P/V	nulován	lichá parita
PE	P/V	nastaven	sudá parita
P	S	nulován	výsledek nezáporný
M	S	nastaven	výsledek záporný

3.4.10.3 Relativní nepodmíněný skok

JR e

PC←PC + e

Instrukce JR zabezpečí relativní skok na adresu paměti, která je dána součtem PC + e. Hodnota e má rozsah jednoho bytu operandu.

Nastavení indikátorů: nezměněno

3.4.10.4 Relativní podmíněný skok

JR cc,e

jestliže je podmínka cc splněna, pak se provede relativní skok PC←PC + e, při nesplněné podmínce cc program pokračuje následující instrukcí.

Indikátorové podmínky cc jsou shrnuty v tabulce 11.

Nastavení indikátorů: nezměněno

Indikátorové podmínky cc instrukce JR Tabulka 11

cc	Indikátor	Stav indikátoru	Stav splňující podmínku
C	C	nastaven	s přenosem
NC	C	nulován	bez přenosu
Z	Z	nastaven	výsledek nulový
NZ	Z	nulován	výsledek nenulový

3.4.10.5 Relativní skok podmíněný čítačem

DJNZ e

B ← B - 1
při B ≠ 0 se provede relativní skok
PC ← PC + e; při B = 0 program pokračuje následující instrukcí

Instrukce DJNZ nejprve zabezpečí dekrementaci B registru, a pakliže je jeho obsah po dekrementaci různý od nuly, provede se relativní skok PC←PC + e. V případě nulového registru B pokračuje program následující instrukcí.

Nastavení indikátorů: nezměněno

3.4.11 Instrukce volání podprogramu nebo podprogramu přerušení

Pro volání podprogramu lze použít nejen nepodmíněnou instrukci CALL nn, ale také podmíněné volání instrukcí CALL cc,nn. Podmíněné volání se uskuteční při splnění podmínky definované v operačním kódu instrukce. V podprogramech přerušení se nejčastěji využívá volací instrukce RST.

3.4.11.1 Nepodmíněné volání podprogramu

CALL nn (SP-1) ← PC_H; (SP-2) ← PC_L; SP ← SP-2; PC ← nn

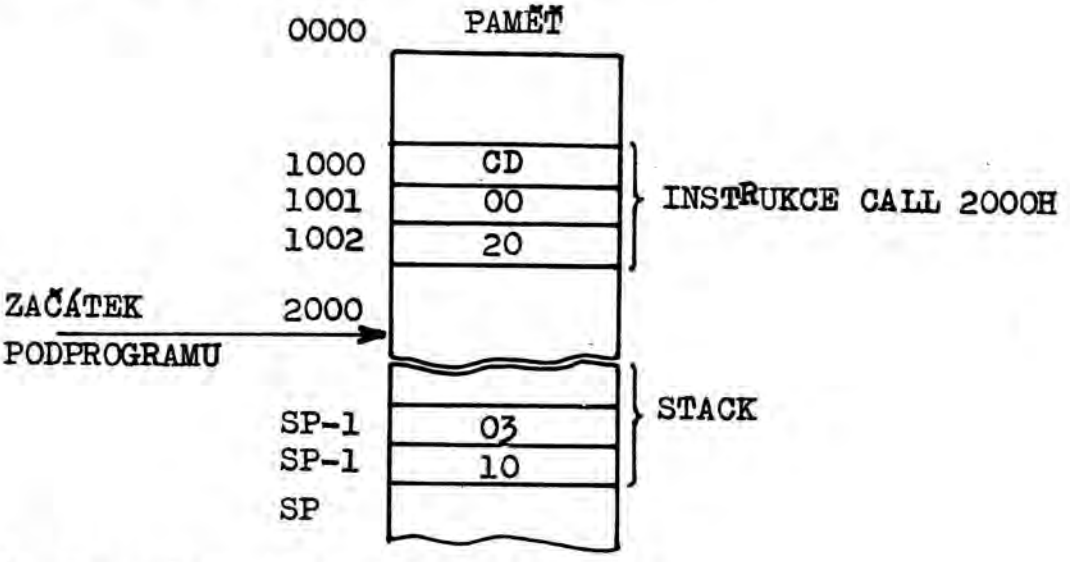
Instrukce CALL zabezpečí volání podprogramu. Návratová adresa hlavního programu, tj. obsah PC, se uloží do zásobníku tak, jako by se ukládaly vyšší a nižší řády PC instrukcí PUSH a do programového čítače se zavede adresa daná dvoubytovým operandem nn.

Nastavení indikátorů: nezměněno

Příklad: Instrukce CALL má provést, volání podprogramu, který začíná na adrese 2000H. Instrukce CALL leží v paměti na adrese 1000 + 1002H.

CALL 2000H

Obr. 61 zachycuje uložení návratové adresy do zásobníku a přechod na adresu 2000H.



Obr. 61 Provedení instrukce CALL 2000H

3.4.11.2 Podmíněné volání podprogramu

CALL cc,nn	jestliže je podmínka cc splněna, pak se provede (SP - 1) ← PC_H; (SP - 2) ← PC_L; SP ← SP - 2; PC ← nn, při nesplněné podmínce cc se pokračuje následující instrukcí.

Podmíněné instrukce CALL provádějí volání podprogramu, jestliže je splněna podmínka, která je definována operandem cc. V opačném případě se pokračuje následující instrukcí v programu. Návratová adresa se při volání podprogramu uloží na vrchol zásobníku. Tato zásada je stejná jako u nepodmíněné instrukce CALL. Podmínka je dána stavem indikátorových bitů Z, C, S, P/V. Přehled podmínek je v tabulce 10.

3.4.11.3 Instrukce RESTART

RST p $(SP - 1) \leftarrow PC_H$; $(SP - 2) \leftarrow PC_L$; $SP \leftarrow SP - 2$; $PC_H \leftarrow 0$; $PC_L \leftarrow p$

Instrukce RST zabezpečí nejprve uložení obsahu PC do zásobníku tak, jako by se ukládaly vyšší a nižší řády PC instrukcí PUSH. Potom se do PC zavede dvoubytová volací adresa podprogramu. Vyšší řády PC jsou vždy nulové, nižší řády PC nabývají hodnoty P. Instrukce má celkem osm strojových podob a využívá ji např. přerušovací systém Z80-CPU pracující v MODE 0.

Nastavení indikátorů: nezměněno

Poznámka

Hodnota p = 00H, 08H, 10H, 18H, 20H, 28H, 30H, 38H.

3.4.12 Instrukce návratu z podprogramu nebo podprogramu přerušení

Instrukce této skupiny zajišťují nepodmíněný nebo podmíněný návrat z podprogramu, případně podprogramu přerušení.

3.4.12.1 Nepodmíněný návrat z podprogramu

RET $PC_L \leftarrow (SP)$; $PC_H \leftarrow (SP + 1)$; $SP \leftarrow SP + 2$

Instrukce RET provádí návrat z podprogramu do hlavního programu. Návratovou adresu instrukce čerpá z vrcholu zásobníku podle stejných zásad jako instrukce POP.

Nastavení indikátorů: nezměněno

3.4.12.2 Podmíněný návrat z podprogramu

RET cc jestliže je podmínka cc splněna, pák se provede $PC_L \leftarrow (SP)$; $PC_H \leftarrow (SP + 1)$; $SP \leftarrow SP + 2$

Podmíněný návrat z podprogramu nastane při splněné indikátorové podmínce cc. Podmínky jsou obsaženy v tabulce 10. Při nesplněné podmínce podprogram pokračuje následující instrukcí.

Nastavení indikátorů: nezměněno

3.4.12.3 Návrat z přerušení v režimu 2

RETI $PC_L \leftarrow (SP)$; $PC_H \leftarrow (SP + 1)$; $SP \leftarrow SP + 2$

Instrukce provádí jednak skok na adresu danou obsahem vrcholu zásobníku a dále oznamuje perifernímu zařízení, jehož přerušení bylo přijato, že končí jeho obsluha. Na periferním zařízení přechází výstupní signál IEO na úroveň logické jedničky. Instrukce neprovádí uvolnění přerušení, které bylo zamaskováno vlastním přijetím přerušení. Tudíž je zapotřebí před instrukcí RETI uvolnit přerušení instrukcí EI.

Nastavení indikátorů: nezměněno

3.4.12.4 Návrat z nemaskovatelného přerušení

RETN $PC_L \leftarrow (SP)$; $PC_H \leftarrow (SP + 1)$; $SP \leftarrow SP + 2$

RETN je instrukce návratu z nemaskovatelného přerušení. Provádí skok na adresu danou vrcholem zásobníku. Instrukce dále uvede klopný obvod povolení přerušení IFF1 do stavu, jaký byl před přijetím nemaskovatelného přerušení. Tuto činnost zabezpečí přesunem klopného obvodu IFF2 do klopného obvodu IFF1.

Nastavení indikátorů: nezměněno

3.4.13 Instrukce vstupů a výstupů

Instrukce řídí styk s periferními zařízeními. Dvě původní instrukce typu IN a OUT společné řadě I 8080, zabezpečující přenos jednoho bytu z periferie do akumulátoru nebo opačným směrem, rozšiřuje řada Z80 o 22 nových instrukcí. Instrukce IN r,(C) adresuje periferii obsahem registru C a bytový přenos z periferie může být veden do volitelného vnitřního registru mikroprocesoru. Instrukce OUT (C),r přenáší bytovou informaci ve směru opačném (volitelný vnitřní registr - periferie).

Zcela nové instrukce vstup/výstupního charakteru pracují blokově. Maximální přenášená zóna je 256 bytů, přenos je veden mezi periferií a operační pamětí nebo ve směru opačném.

3.4.13.1 Instrukce vstupu

IN a.,b.	a. ← b.

Instrukce IN zabezpečí přenos jednoho bytu dat z periferního zařízení, adresovaného operandem b. do volitelného registru operandem a.

Nastavení indikátorů: při kombinaci IN a.,(n) nezměněno
při kombinaci operandů IN a.,(C) je nastavení indikátorů následující

- S nastaven, jestliže je hodnota vstupujících dat záporná
- Z nastaven, jestliže jsou vstupní data nulová
- H nulová
- P/V nastaven v případě sudé parity
- N nulován
- C nezměněn

Programovatelné varianty operandů jsou v tabulce 12.

Varianty operandů instrukce IN a OUT Tabulka 12

b. / a.	A	B	C	D	E	H	L
(n)	-						
(C)	-	-	-	-	-	-	-

(n) - symbol znamená, že periferie je adresována druhým bytem instrukce

(C)- symbol znamená, že periferie je adresována obsahem C registru mikroprocesoru

3.4.13.2 Blokový vstup s inkrementací bez opakování

INI	(HL) ← (C); B ← B - 1; HL ← HL + 1

Vstupní instrukce INI přenáší jeden byte z periferie o adrese, která je uložená v registru C na paměťovou buňku, jejíž adresa je uložená v párovém registru HL. Potom se provede dekrementace registru B a inkrementace registru HL.

Nastavení indikátorů: S nedefinován
Z nastaven, jestliže je B - 1 = 0
H nedefinován
P/V nedefinován
N nastaven
C nezměněn

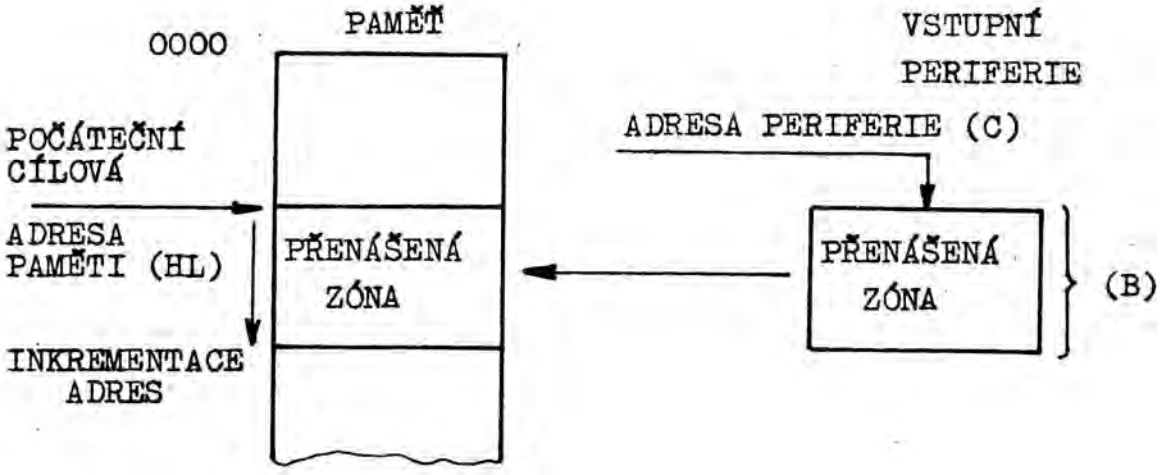
3.4.13.3 Blokový vstup s inkrementací a opakováním

INIR (HL) ← (C); B ← B - 1; HL ← HL + 1;
opakování přenosu do stavu B = 0

Instrukce INIR řeší přenos z periferie o adrese lokalizované reg. C na paměťovou buňku, lokalizovanou párovým registrem HL. Instrukce dekrementuje B registr a inkrementuje párový registr HL. Přenosy provádí tak dlouho, dokud B ≠ 0. Načtené byty jsou uloženy vzestupně v operační paměti na adresy definované obsahem párového registru HL. Když obsah registru B dosáhne nuly, Z80-CPU pokračuje následující instrukcí. Poněvadž test na nulovost registru B je prováděn až po dekrementaci tohoto registru, má za následek úvodní nastavení registru B na nulu přenos 256 bytů.

Nastavení indikátorů: S nedefinován
Z nastaven
H nedefinován
P/V nedefinován
N nastaven
C nezměněn

Způsob blokového přenosu vedený instrukcí INIR zachycuje obr. 62.



Obr. 62 Blokový přenos vedený instrukcí INIR

3.4.13.4 Blokový vstup s dekrementací bez opakování

IND (HL) ← (C); B ← B - 1; HL ← HL - 1

Vstupní instrukce IND přenáší jeden byte z periferie o adrese uložené v registru C na paměťovou buňku, jejíž adresa je v párovém registru HL. Potom se provede dekrementace registru B a dekrementace párového registru HL. Instrukce IND se od instrukce INI liší pouze dekrementací obsahu registrového páru HL.

Nastavení indikátorů:
- S nedefinován
- Z nastaven, jestliže je B - 1 = 0
- H nedefinován
- P/V nedefinován
- N nastaven
- C nezměněn

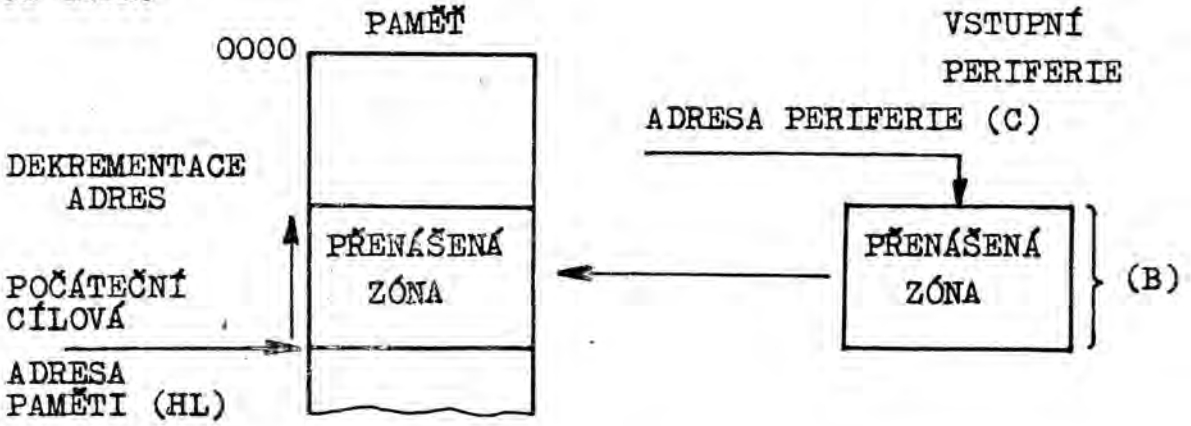
3.4.13.5 Blokový vstup s dekrementací a opakováním

INDR (HL) ← (C); B ← B - 1; HL ← HL - 1;
opakování přenosu až do stavu B = 0

Vstupní instrukce, obdobná instrukci INIR, liší se jen dekrementací registrového páru HL, a tudíž načtené byty jsou v operační paměti uloženy sestupně.

Nastavení indikátorů: S nedefinován
Z nastaven
H nedefinován
P/V nedefinován
N nastaven
C nezměněn

Na obr. 63 je vidět způsob vedení blokového přenosu instrukcí INDR.



Obr. 63 Blokový přenos vedený instrukcí INDR

3.4.13.6 Instrukce výstupu

OUT a.,b. a. ← b.

Instrukce OUT zabezpečí přenos jednoho byte do periferie adresované operandem a. z registru, který je lokalizován operandem b..

Programovatelné varianty operandů byly uvedeny v tab. 12.

Nastavení indikátorů: nezměněno

3.4.13.7 Blokový výstup s inkrementací bez opakování

OUTI (C) ← (HL); B ← B - 1; HL ← HL + 1

Výstupní instrukce OUTI přenese jeden byte z paměťové buňky, jejíž adresu obsahuje párový registr HL na periferní zařízení, jehož adresu obsahuje registr C. Potom se dekrementuje registr B a inkrementuje párový registr HL.

Nastavení indikátorů:
- S nedefinován
- Z nastaven, jestliže je B - 1 = 0
- H nedefinován
- P/V nedefinován
- N nastaven
- C nezměněn

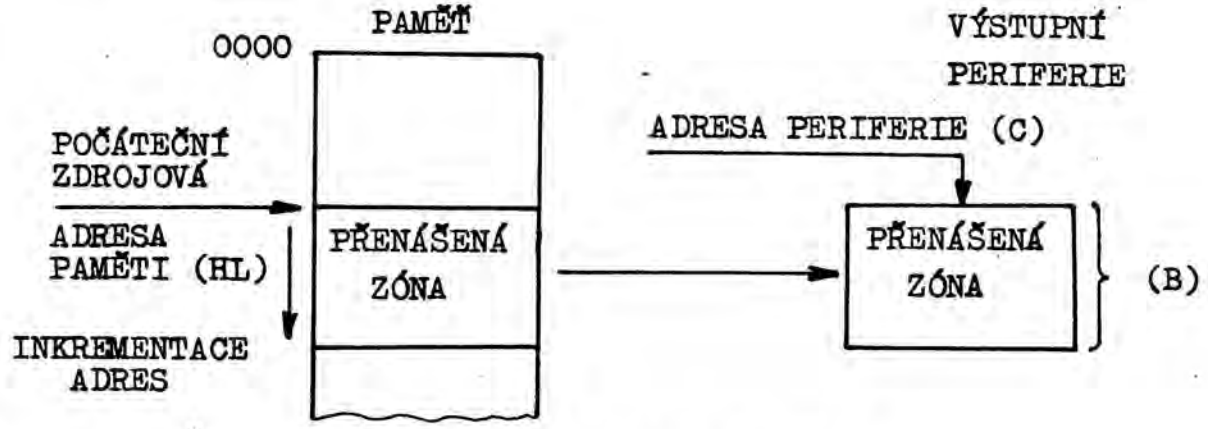
3.4.13.8 Blokový výstup s inkrementací a opakováním

OTIR (C) ← (HL); B ← B - 1; HL ← HL + 1;
opakování přenosu až do stavu B = 0

Instrukce OTIR řeší přenos jednoho bytu dat z paměťové buňky, jejíž adresa je dána obsahem párového registru HL na periferii, jejíž adresu obsahuje registr C. Potom se provede dekrementace registru B a inkrementace párového registru HL. Tato činnost se opakuje tak dlouho, dokud obsah registru B je různý od nuly. Při stavu B = 0 Z80-CPU pokračuje následující instrukcí. Vyčítání informací z paměti probíhá vzestupně po adresách lokalizovaných obsahem párového registru HL.

Nastavení indikátorů:
- S nedefinován
- Z nastaven
- H nedefinován
- P/V nedefinován
- N nastaven
- C nezměněn

Způsob vedení blokového přenosu instrukcí OTIR zachycuje obr. 64.



Obr. 64 Blokový přenos vedený instrukcí OTIR

3.4.139 Blokový výstup s dekrementací bez opakování

OUTD (C) ← (HL); B ← B - 1; HL ← HL - 1

Výstupní instrukce OUTD řeší vyčtení jednoho bytu dat z paměťové buňky, jejíž adresu obsahuje párový registr HL na periferii, jejíž adresu obsahuje registr C. Potom se provádí dekrementace registru B a dekrementace párového registru HL. Instrukce OUTD se od instrukce OUTI odlišuje jen dekrementací obsahu registrového páru HL.

Nastavení indikátorů:

S	nedefinován
Z	nastaven, jestliže je B - 1 = 0
H	nedefinován
P/V	nedefinován
N	nastaven
C	nezměněn

3.4.1310 Blokový výstup s dekrementací a opakováním

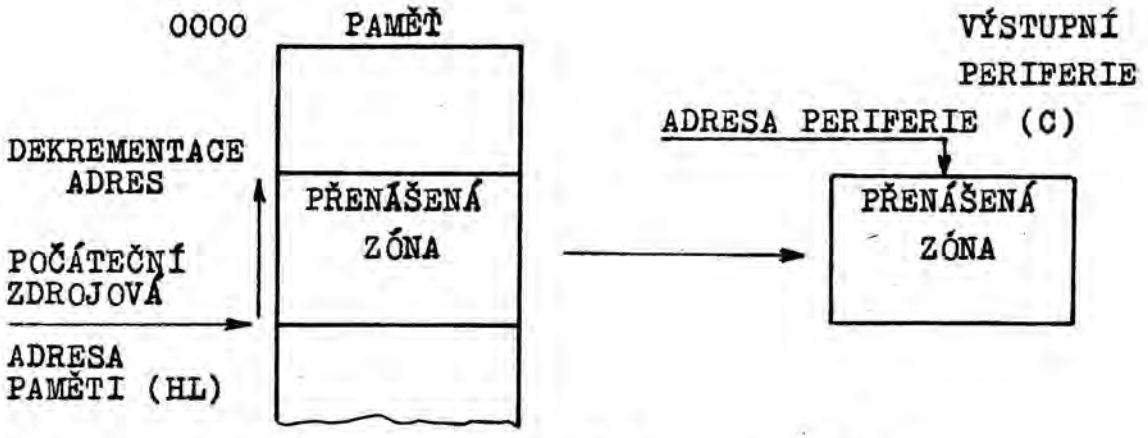
OTDR (C) ← (HL); B ← B - 1; HL ← HL - 1;
opakování přenosu až do stavu B = 0

Výstupní instrukce blokového přenosu OTDR je obdobou instrukce OTIR. Rozdíl je jen v tom, že nyní dochází k sestupnému vyčítání buněk operační paměti.

Nastavení indikátorů:

S	nedefinován
Z	nastaven
H	nedefinován
P/V	nedefinován
N	nastaven
C	nezměněn

Na obr. 65 je vidět způsob vedení blokového přenosu instrukcí OTDR.



Obr. 65 Blokový přenos vedený instrukcí OTDR

4. PROGRAMOVATELNÉ PODPŮRNÉ OBVODY ŘADY Z-80

Podpůrné obvody řady Z-80 jsou konstruovány tak, aby umožňovaly pružnou spolupráci procesoru Z80-CPU s periferními jednotkami V/V. Tyto prvky řady Z-80 obsahují vnitřní registry pro zachycení programových neboli řídicích slov (CONTROL WORD). Řídicí slova definují vlastnosti programovatelného obvodu v další fázi činnosti. Zadávání řídicích slov je vedeno přes konkrétní adresy nebo probíhá v pevné sekvenci zajišťované repeatovou výstupní instrukcí OTIR nebo OTDR.

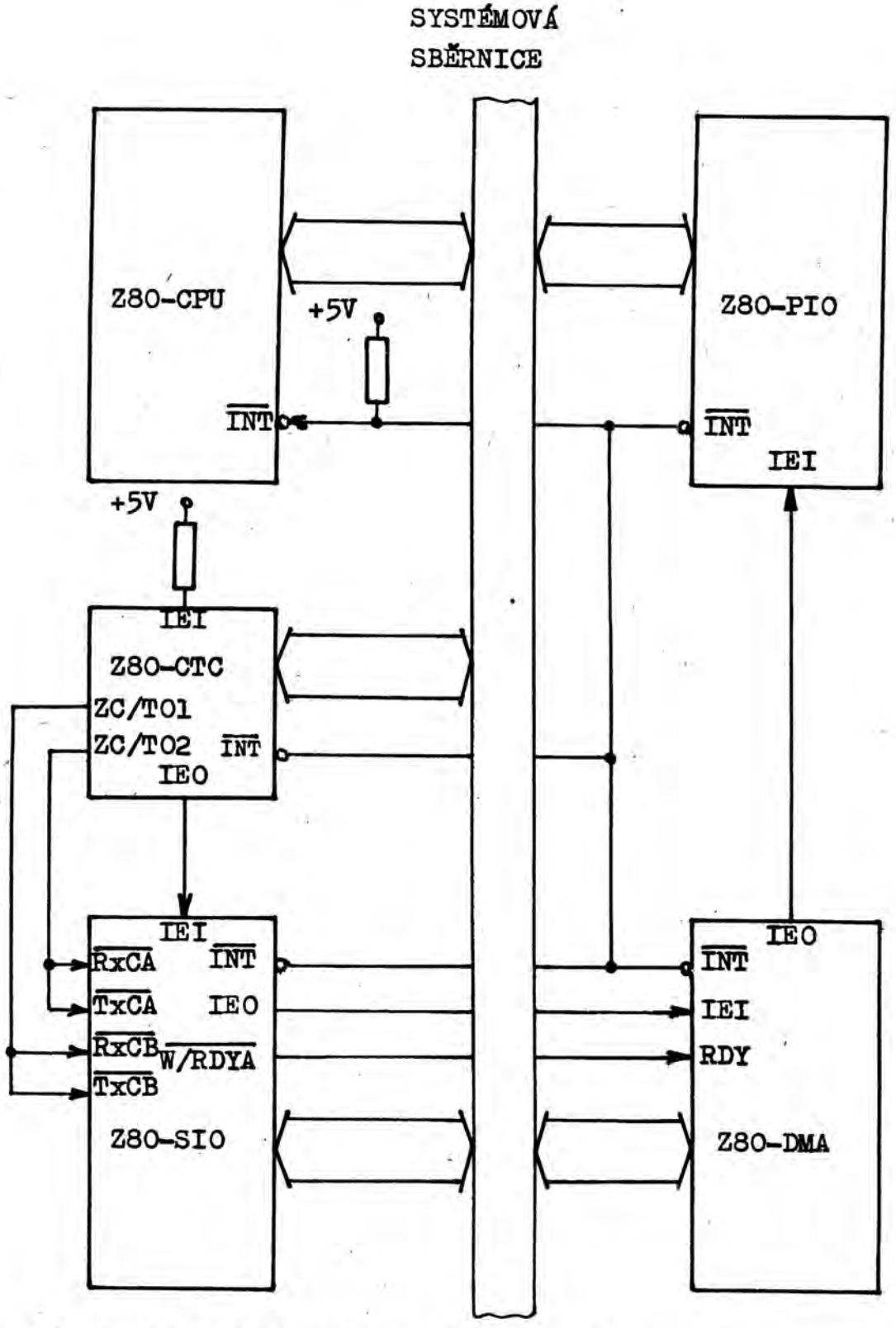
Z hlediska přerušení obvody pracují v MODE 2, v kterém vytváří sériové řetězce priorit, a tím přímo přiřazují prioritu V/V zařízení dle své polohy v řetězci. V režimu 2 pak není zapotřebí přídavný hardware na ošetření přerušení.

Systém je vybaven novými instrukcemi, které optimalizují zpracování přerušení a řeší blokové přenosy informací mezi periferií a pamětí nebo pamětí a periferií. Zástupci řady Z-80:

Z80-PIO PARALLEL INPUT/OUTPUT CONTROLLER
Z80-SIO SERIAL INPUT/OUTPUT CONTROLLER
Z80-DART DUAL ASYNCHRONOUS RECEIVE/TRANSMITTER
Z80-CTC COUNTER/TIMER CONTROLLER
Z80-DMA DIRECT MEMORY ACCESS CONTROLLER

Celá škála podpůrných integrovaných obvodů je dovážena z NDR(UB855D, UB856D, UB857D, UB858D, UB8563D). Z přehledu je vidět, že stavebnice sestává jen z pěti podpůrných obvodů, což by mohlo vést k mylné představě o omezených možnostech této řady. Obvody jsou vysoce universální a mohou být doplněny v případě potřeby standardizovanými prvky pro mikropočítače jiných výrobců např. obvody firmy Intel.

Všechny obvody jsou vyrobeny technologií N-MOS s jednotným napájením + 5 V. Vytvoření sériového řetězce priorit obvodů řady Z-80 je na obr. 66.



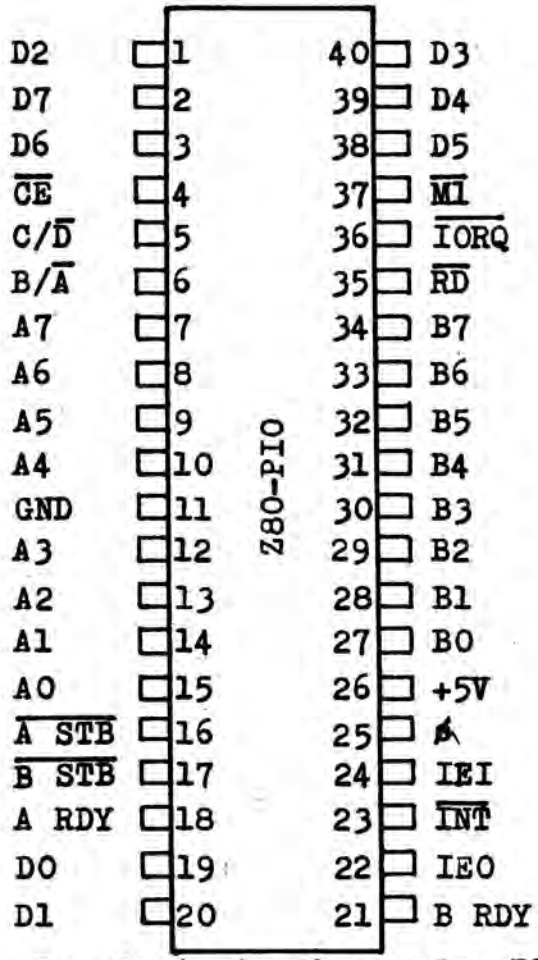
Obr. 66 Typická konfigurace systému Z-80

4.1 Z80-PIO

LSI obvod Z80_PIO (PARALLEL INPUT/OUTPUT CONTROLLER) tvoří vazební člen mezi mikroprocesorem a periferií, pracující paralelním způsobem např. snímač, děrovač děrné pásky, tiskárna, klávesnice atd. Předávání informací mezi periferií a procesorem probíhá pod řízením přerušovacího systému v MODE 2.

Obvod je dvoubránový, programovatelný do čtyř režimů práce (MODE 0 ÷ 3). Brány mají pevně přiřazenou prioritu, tj. brána A má vyšší prioritu jak brána B. Každá brána generuje samostatný vektor přerušení. Řízení čipu je jednofázovými hodinami, napájecí napětí + 5 V, pouzdro DIL je 40 vývodové.

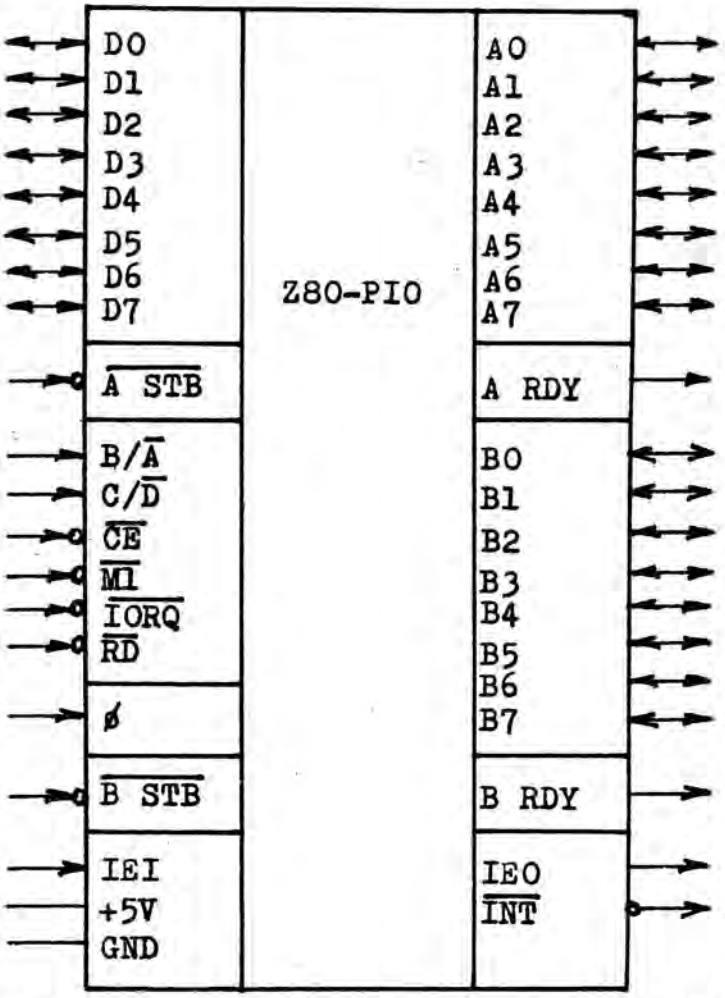
Zapojení vývodů pouzdra je na obr. 67.



Obr. 67 Zapojení vývodů pouzdra Z80-PIO

4.1.1 Popis vývodů pouzdra

Na obr. 68 je uvedena schematická značka stykového obvodu Z80-PIO.



Obr. 68 Schematická značka obvodu Z80-PIO

D0 ÷ D7 datová sběrnice	třístavový vstup/výstup. Sběrnice slouží k přenosu řídicích a datových slov mezi procesorem a obvodem Z80-PIO
B/$\overline{A}$ výběr kanálu	vstupní signál aktivující bránu A nebo B. Úrovní H se aktivuje brána B, úrovní L se aktivuje brána A.

C/$\overline{D}$ výběr typu přenosu	vstupní signál. Úrovní na vstupu C/$\overline{D}$ se udává, jaká informace je předána z mikroprocesoru do obvodu PIO. Úroveň H říká, že po datové sběrnici jsou přenášena řídicí slova, úroveň L identifikuje přenos dat.
$\overline{M1}$ prvý strojový cyklus	vstupní signál mající tyto významy: - synchronizuje funkce bloku přerušení $\overline{M1}$ = 0 a $\overline{IORQ}$ = 0 - cyklus $\overline{INTA}$ - nuluje obvod PIO tj. nastaven vstupní režim pro oba kanály, pokud se vyskytne signál $\overline{M1}$ bez aktivních signálů $\overline{RD}$ a $\overline{IORQ}$.
$\overline{RD}$ čtení	vstupní signál. Společně se signály $\overline{IORQ}$, B/$\overline{A}$, C/$\overline{D}$ a $\overline{CE}$ signalizuje čtení dat do Z80-CPU. Pakliže signál $\overline{RD}$ není aktivní a další signály aktivní jsou, znamená to, že data jsou do Z80-PIO zapisována a nikoliv čtena ze Z80-PIO.
$\overline{IORQ}$ žádost na V/V zařízení	vstupní signál. $\overline{IORQ}$ je generován při provádění vstup/výstup instrukcí, pakliže je připojen na stejnojmenný vodič řídicí sběrnice mikroprocesoru. Je-li $\overline{IORQ}$ = 0 a $\overline{M1}$ = 0, jde o přerušovací cyklus. Obecně se signálu $\overline{IORQ}$ užívá spolu se signály B/$\overline{A}$, C/$\overline{D}$, $\overline{RD}$ a $\overline{CE}$ k identifikaci typu a směru přenosu mezi mikroprocesorem a stykovým obvodem.
$\overline{CE}$ výběr obvodu	vstupní signál. Nulová úroveň tohoto signálu aktivuje Z80-PIO.
IEI povolení generování přerušení	vstupní signál, používaný pro vytváření sériových řetězců priorit.
IEO povolení generování přerušení	výstupní signál. IEO je rovněž použit pro vytváření prioritních řetězců.

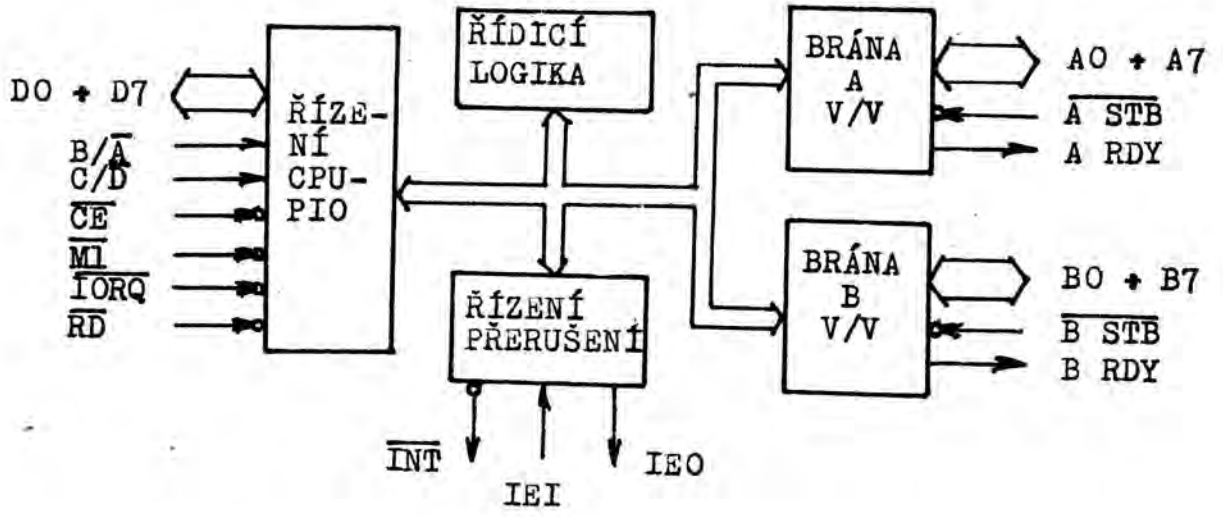
$\overline{INT}$ přerušení	výstupní signál na otevřeném kolektoru. Signálem $\overline{INT}$ signalizuje obvod PIO mikroprocesoru žádost o přerušení.
ø systémové hodiny	vstup pro připojení systémových hodin. Provádí interní synchronizaci signálů.
A0 ÷ A7 vodiče brány A	třístavová vstupní/výstupní brána. Vodiče A0 ÷ A7 slouží k přenosu informací mezi bránou A a okolím.
A RDY řídicí signál brány A	výstupní signál. Využití signálu je závislé na režimu práce (MODE), v kterém příslušná brána pracuje: - MODE 0 - výstupní režim - A RDY udává, že ve výstupním registru jsou připravena data k převzetí a tato data jsou též přítomna na vodičích brány. - MODE 1 - vstupní režim - A RDY udává, že vstupní registr brány A je prázdný a proto je schopen akceptovat data. - MODE 2 - obousměrný režim - A RDY má stejný význam jako v MODE 0, s tím rozdílem, že data na výstupech brány A jsou dostupná až při nulovém signálu $\overline{A\ STB}$. - MODE 3 - řídicí režim - při tomto režimu není signál A RDY vůbec použit.
$\overline{A\ STB}$ řídicí signál brány A	vstupní signál. Význam signálu je opět závislý na režimu, ve kterém brána A pracuje: - MODE 0 - výstupní režim - náběžnou hranou tohoto signálu oznamuje periferie stykovému obvodu, že převzala data z brány A a uvolnila výstupní buffer pro další informaci. - MODE 1 - vstupní režim - při aktivní úrovni na vstupu $\overline{A\ STB}$ se přepisují data ze vstupů A0 ÷ A7 do vstupního bufferu brány A.

	- MODE 2 - obousměrný režim - v době aktivního signálu $\overline{\text{A STB}}$ jsou na vodičích brány A data, která byla zapsána do výstupního registru signálem A RDY. Padací hranou je potvrzeno převzetí dat. - MODE 3 - řídicí režim - tento režim nevyužívá řídícího vodiče $\overline{\text{A STB}}$.
B0 ÷ B7 vodiče brány B	třístavová vstupní/výstupní brána. Brána B plní stejnou funkci jako brána A.
B RDY řídicí signál brány B	výstupní signál. Jediný rozdíl řídicího signálu B RDY od signálu A RDY je v tom, že v obousměrném režimu je využit k indikaci, že vstupní registr brány A je prázdný, a tím je připraven převzít data z okolí.
$\overline{\text{B STB}}$ řídicí signál brány B	vstupní signál. Jediný rozdíl řídicího signálu $\overline{\text{B STB}}$ od signálu $\overline{\text{A STB}}$ je v tom, že v obousměrném režimu se používá k vložení dat do vstupního registru brány A z vnějšku.

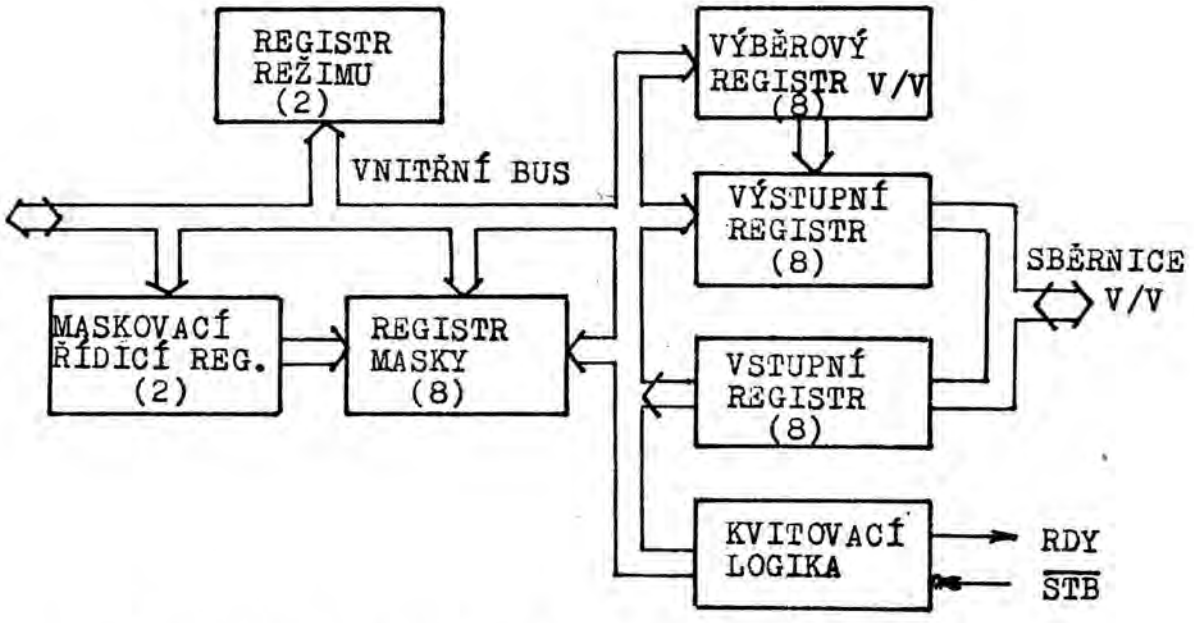
4.1.2 Architektura obvodu

Vnitřní strukturu obvodu lze rozdělit do pěti funkčních částí, které jsou vzájemně propojeny vnitřní sběrnicí. Blok řízení CPU-PIO zabezpečuje styk s mikroprocesorem klasickými sběrnicovými signály. Blok řízení přerušení obsahuje dva vyrovnávací registry vektorů přerušení pro kanál A a B a sekvenční logiku pro odbavení přerušení. Styk s periferiemi řídí blok brány A V/V a brány B V/V, které jsou vybaveny obousměrnými osmibitovými kanály a po jednom vstupním a výstupním řídícím signálu typu HANDSHAKING. Práci těchto celků koordinuje blok řídicí logiky. Jednotlivé bloky zachycuje obr. 69.

Obecný rozbor brány V/V je na obr. 70. Každý blok brány V/V obsahuje sadu registrů, a to vyrovnávací výstupní registr a vyrovnávací vstupní registr. Způsoby plnění a vyčítání těchto registrů definuje jednak kvitovací logika typu HANDSHAKING a jednak režim práce (MODE 0÷3), definovaný programově ve dvoubitovém registru



Obr. 69 Blokové schéma Z80-PIO



Obr. 70 Struktura brány V/V obvodu Z80-PIO

režimu. Režim 3 pro svou funkci ještě využívá dvoubitový maskovací řídicí registr, osmibitový registr masky a osmibitový výběrový registr V/V. Význam těchto registrů bude objasněn v souvislosti s popisem režimu 3.

4.1.3 Režimy práce

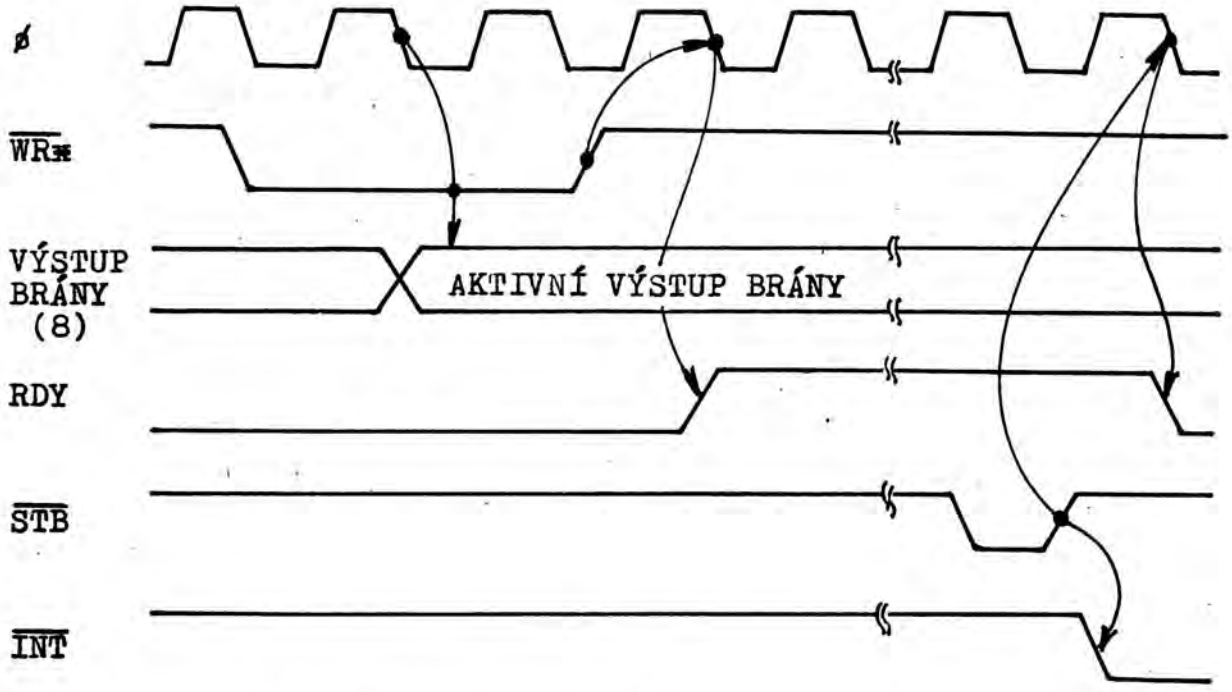
Signály, generované v jednotlivých režimech práce, jsou vnitřně synchronizovány vstupním hodinovým signálem ϕ, který přichází od mikroprocesoru Z80-CPU. Stykový obvod Z80-PIO může pracovat ve čtyřech režimech práce (MODE 0÷3).

MODE 0 - výstupní režim (OUTPUT MODE)

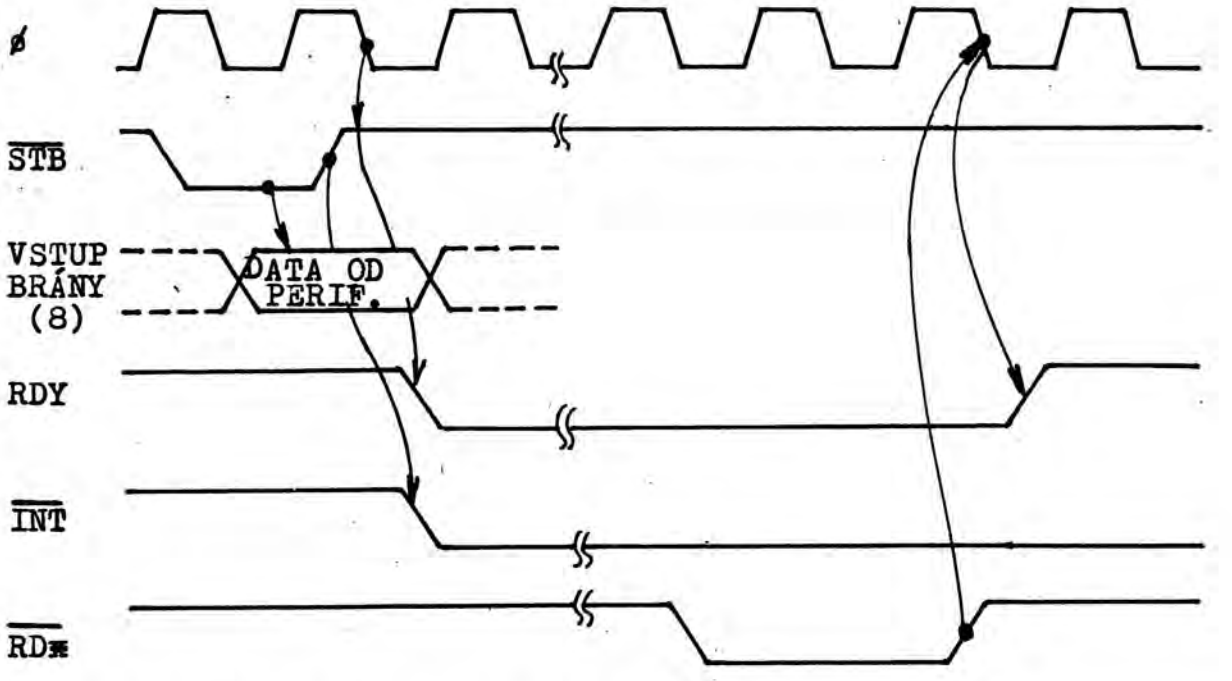
Zápisový signál $\overline{WR}$ je odvozen od neaktivního signálu $\overline{RD}$ za přítomnosti aktivních signálů $\overline{CE}$, C/$\overline{D}$, $\overline{IORQ}$. Splněním tohoto součinu dojde k zápisu dat do výstupního registru s první padací hranou hodin. Následující padací hrana hodin ϕ uvolňuje datové slovo na výstupech určených pro připojení periferie. Konec zápisového cyklu s padací hranou hodin ϕ aktivuje signál RDY, který informuje periferii o přítomnosti dat k odebrání. Shození signálu RDY je podmíněno padací hranou hodin po ukončení signálu $\overline{STB}$, kterým periferie oznamuje odebrání bytu dat ze stykového obvodu. Zároveň od náběžné hrany signálu $\overline{STB}$ je generováno přerušení na výstupu $\overline{INT}$, pakliže je vznik tohoto přerušení povolen řídicím slovem. Výstupní vodiče brány se přepisují až další datovou informací ze zápisového cyklu mikroprocesoru. Aktivita signálu $\overline{INT}$ je ukončena přerušovacím cyklem ($\overline{IORQ}$.$\overline{M1}$), kterým se přenáší vektor přerušení do Z80-CPU. Časový diagram výstupního režimu je na obr. 71.

MODE 1 - vstupní režim (INPUT MODE)

Signál $\overline{STB}$ zapisuje data od vstupní periferie do vstupního registru brány. Příchod signálu $\overline{STB}$ byl vyvolán jedničkovým signálem RDY, kterým obvod Z80-PIO informuje periferii o uprázdnění svého vstupního registru. Náběžná hrana signálu $\overline{STB}$ vyvolá $\overline{INT}$, kterým je žádán mikroprocesor o odebrání znaku ze stykového obvodu. Po ukončení signálu $\overline{STB}$ a následující padací hraně hodin dochází k přechodu signálu RDY na nulu, čímž je hlášeno naplnění



Obr. 71 MODE 0 - výstupní režim



Obr. 72 MODE 1 - vstupní režim

vstupního registru. Po převzetí dat mikroprocesorem za pomoci $\overline{RD}$ tj. aktivní signál $\overline{RD}$, $\overline{CE}$, C/$\overline{D}$, $\overline{IORQ}$ se s následující padací hranou hodin ϕ aktivuje signál RDY. **Signál $\overline{INT}$ se vrací do neaktivní úrovně až na začátku ošetření požadavku přerušení ($\overline{M1}$.$\overline{IORQ}$) obr.72.**

MODE 2 - obousměrný režim (BIDIRECT MODE)

Tento režim lze aplikovat pouze u kanálu A a jeho časový diagram je na obr. 73. Obousměrný režim je průnikem vstupního a výstupního režimu. Řídicí signály B RDY a $\overline{B\ STB}$ jsou využity pro řízení vstupu informace od periferie, kdežto signály A RDY a $\overline{A\ STB}$ řídí výstup informace z obvodu Z80-PIO směrem k periferii. Přerušovací signál $\overline{INT}$ je společný pro oba směry přenosu. Při práci v MODE 2 lze využít kanálu B pouze v MODE 3, poněvadž výše uvedený MODE 2 pracuje se všemi řídicími signály obou kanálů. Zásadní rozdíl práce v MODE 2 oproti MODE 0 je v tom, že výstupní data jsou na výstupech A0 ÷ A7 přítomna po dobu aktivního signálu $\overline{A\ STB}$. Vstup dat do vstupního registru zajišťuje signál $\overline{B\ STB}$. Vodiče typu A RDY a B RDY definují obsazení vstupního a výstupního registru informací. Přerušovací signál $\overline{INT}$ je jednak generován po odebrání dat z výstupního registru signálem $\overline{A\ STB}$ a jednak po naplnění vstupního registru zařízení signálu $\overline{B\ STB}$.

MODE 3 - bitový režim (BIT MODE)

Bitový režim lze uplatnit u kanálu A nebo B. V režimu 3 zcela ztrácejí význam řídicí signály typu $\overline{STB}$ a RDY. Výstupní řídicí signál RDY je v tomto režimu nastaven na nulu, pakliže není aplikován MODE 2, který by využil signály typu RDY pro řídicí funkce. V bitovém režimu je možno pracovat s jednotlivými bity, skupinami bitů nebo celými kanály jako se vstupními či výstupními porty. Orientaci portů a formy práce definují programovací slova. Okamžik čtení vstupní informace lze ošetřit přes přerušení, které nastane, jsou-li splněny logické podmínky definované osmibitovou maskou a dvoubitovým registrem řízení masky. Žádost o přerušení je generována při změně výsledku logického výrazu z 0 na 1. Čtení brány lze provádět kdykoliv signálem $\overline{RD}$. Porty orientované jako výstupy jsou přečteny ze svého výstupního registru a z portů orientovaných jako vstupy je sejmut jejich okamžitý stav,

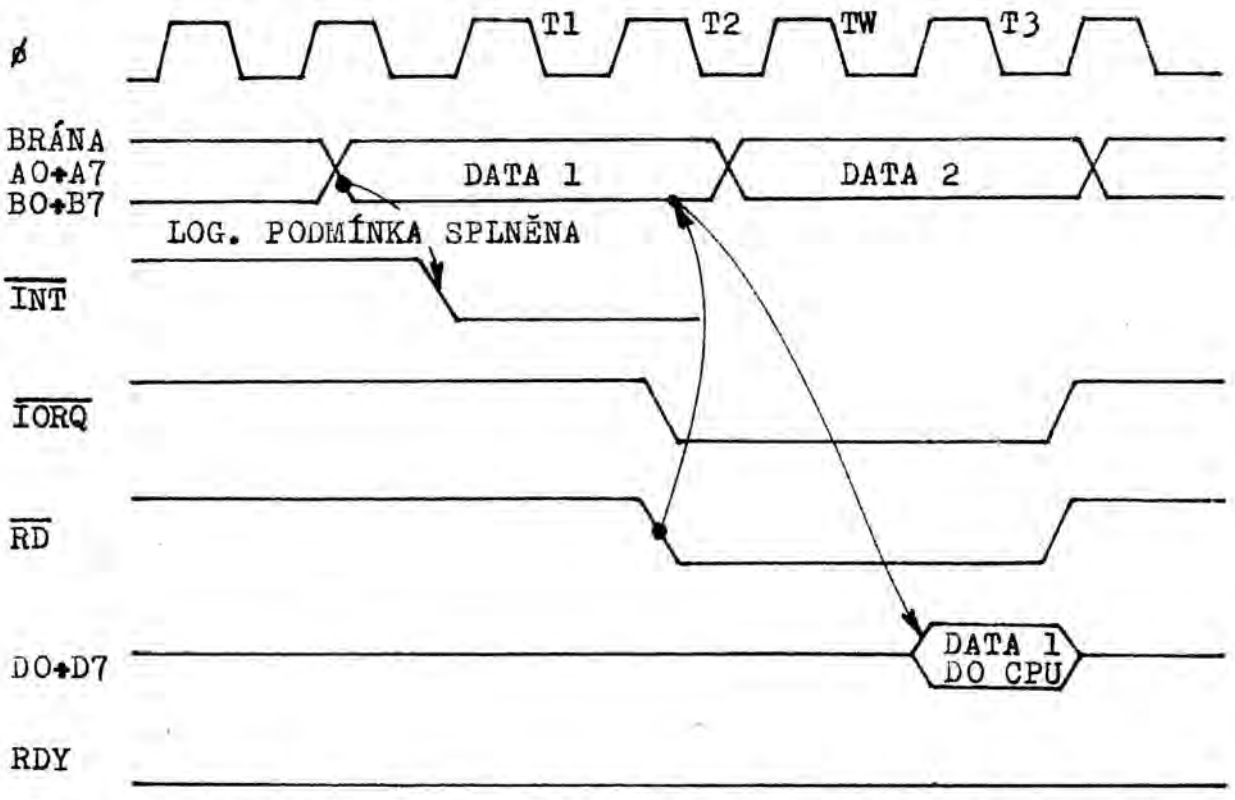
Obr. 73 MODE 2 - obousměrný režim

nacházející se před příchodem padací hrany signálu $\overline{RD}$. Časový diagram je zachycen na obr. 74 pro čtení informací z portů orientovaných jako vstupy.

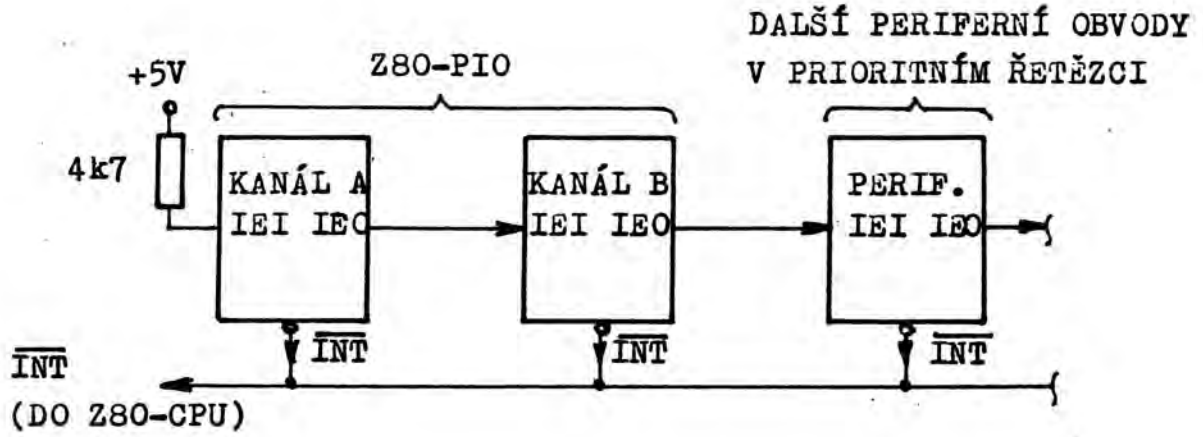
Odbavení přerušení (INTERRUPT)

Programovatelný obvod Z80-PIO, tak jako i ostatní prvky této řady, mohou plně využít přerušovacího systému pracujícího v MODE 2, který se nastavuje instrukcí IM2 na úrovni Z80-CPU.

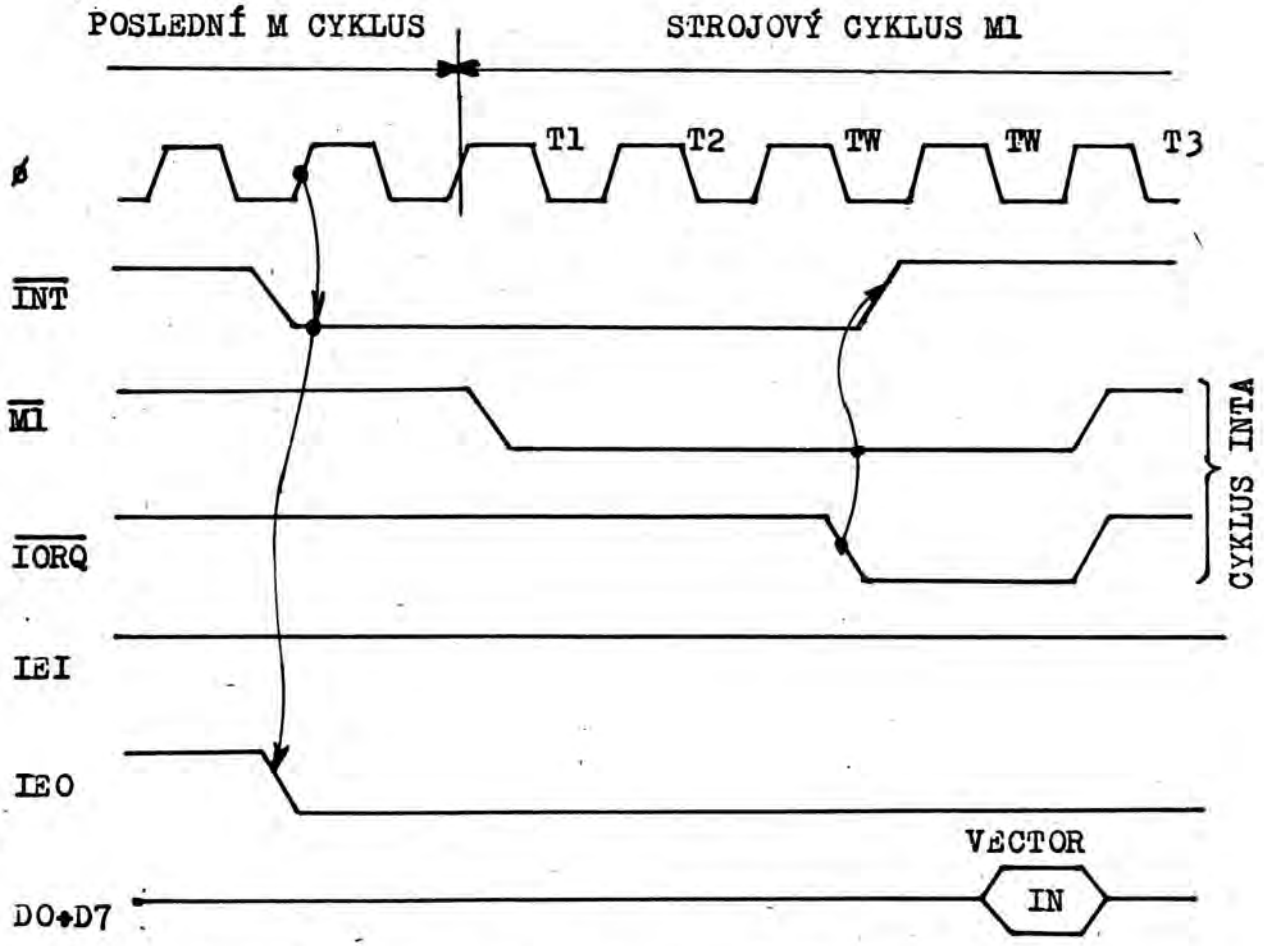
Žádost o přerušení na vstupu $\overline{INT}$ mikroprocesoru je akceptována v posledním taktu posledního strojového cyklu prováděné instrukce. Signál IEO je shozen do nuly od přerušení $\overline{INT}$. Nulová úroveň signálů $\overline{M1}$ a $\overline{IORQ}$ vyvolává vektor přerušení na datovou sběrnici. Výstupní signál IEO setrvá v nule až do příchodu dvoubytové instrukce RETI (ED4D), kterou je ukončena obsluha přerušení vyvolaného obvodem Z80-PIO. Instrukce RETI je přímo dekódována v podpůrném obvodě. Sériový prioritní řetězec je na obr. 45. Časové průběhy požadavku přerušení a návratu z přerušení zachycují obr. 76 a obr. 77.



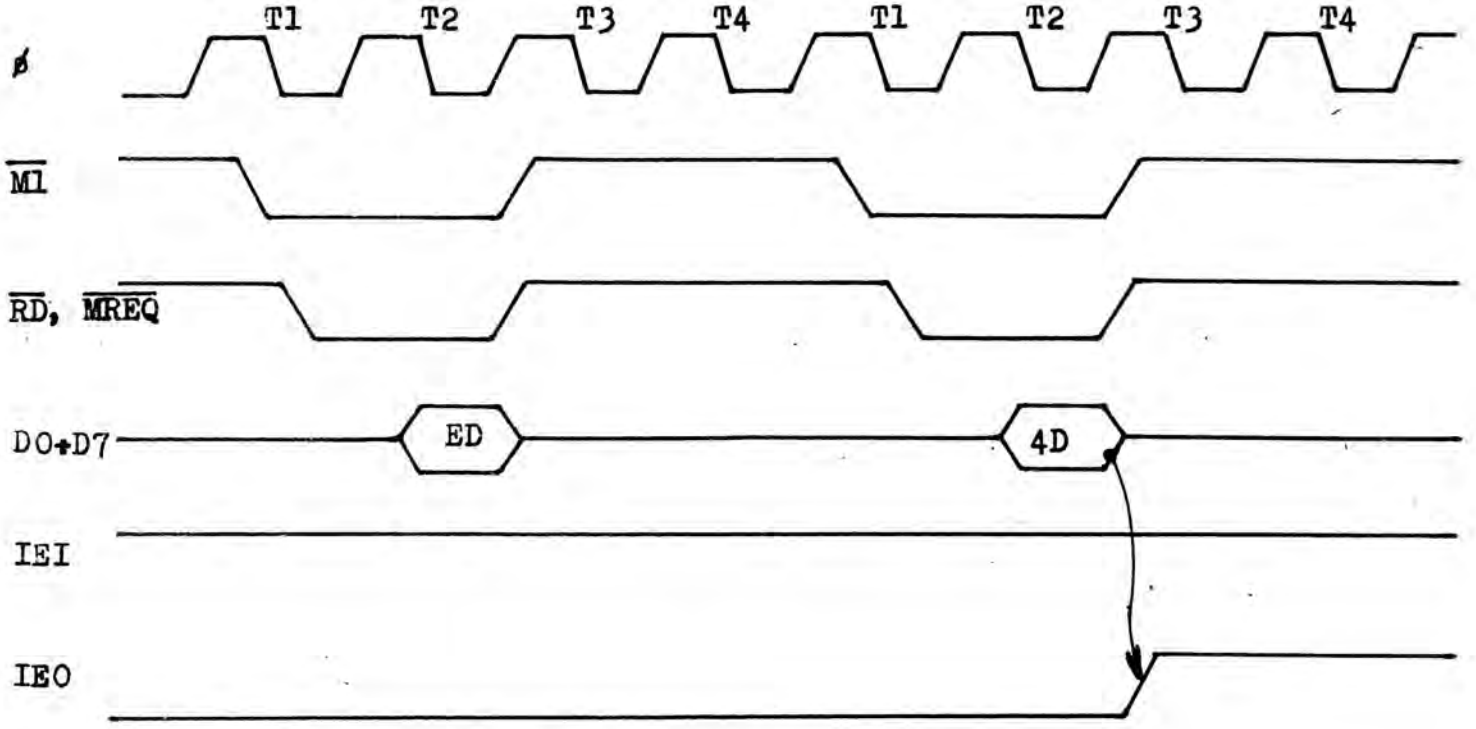
Obr. 74 MODE 3 - bitový režim (čtení portů do mikroprocesoru)



Obr. 75 Sériový prioritní řetězec kanálů Z80-PIO a dalších obvodů



Obr. 76 Časový průběh cyklu požadavku přerušení



Obr. 77 Návrat z podprogramu přerušení

4.1.4 Programování Z80-PIO

Automatické nulování obvodu Z80-PIO nastává po připojení napájení. Resetový stav je definován:

- nastavením MODE 1 tj. vstupní režim
- maskovací registry obou kanálů provádí inhibici všech bitů
- výstupy bran jsou ve třetím stavu a jejich řídicí vodiče se nachází v neaktivním stavu
- vyrovnávací registry bran jsou nulovány
- klopné obvody povolení přerušení jsou nulovány
- stav vektoru přerušení není resetem ovlivněn

Ve stavu reset obvod Z80-PIO setrvá až do příjmu řídicích slov. Programování obvodu je provedeno po polovinách signálem $B/\overline{A}$ za stavu signálu $C/\overline{D} = 1$. Jednotlivá řídicí slova jsou adresována bitovou hodnotou řídicích slov a jejich význam je následující.

Formát vektoru přerušení

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

vektor přerušení — adresa řídicího slova vektoru přerušení

Bity V1 ÷ V7 definují nižší řády vektoru přerušení, vyšší řády vektoru přerušení určuje stav registru I mikroprocesoru.

Formát režimového řídicího slova

D7	D6	D5	D4	D3	D2	D1	D0
M1	M0	X	X	1	1	1	1

režim (MODE) — adresa režimového řídicího slova

Adresa tohoto typu řídicího slova je definována bity D0 ÷ D3. Pozice D4, D5 nejsou dekódovány a nejvyšší bity D6, D7 určují režim práce obvodu a plní dvoubitový registr režimu.

Tabulka 13 ukazuje režim práce na základě obsazení bitu M0, M1 režimového řídicího slova.

Volba režimu práce Z80-PIO Tabulka 13

M1	M0	MODE	REŽIM
0	0	OUTPUT	VÝSTUPNÍ
0	1	INPUT	VSTUPNÍ
1	0	BIDIRECT	OBOUSMĚRNÝ
1	1	BIT	BITOVÝ

Je-li nastaven bitový režim, pak následující řídicí slovo určuje, která linka je vstupní a která výstupní.

Formát řídicího slova pro orientaci linek v režimu 3

D7	D6	D5	D4	D3	D2	D1	D0
I/07	I/06	I/05	I/04	I/03	I/02	I/01	I/00
INPUT/OUTPUT							

I/O = 0 - příslušný port je výstupem
I/O = 1 - příslušný port je vstupem

Programové slovo obsazuje osmibitový výběrový registr V/V.

Formát přerušovacího řídicího slova

Pozice D5 a D6 přerušovacího řídicího slova obsazuje dvoubitový maskovací řídící registr.

D7	D6	D5	D4	D3	D2	D1	D0
EI	AND/$\overline{OR}$	H/$\overline{L}$	M	0	1	1	1

bity D4 + D6 mají význam pouze v režimu 3, ostatní režimy je ignorují

adresa přerušovacího řídicího slova

D7 je využit ve všech režimech práce

EI - povolení přerušení

EI = 1 - přerušení povoleno

EI = 0 - přerušení blokováno

AND/$\overline{OR}$ - součinová nebo součtová funkce pro vznik přerušení

AND/$\overline{OR}$ = 1 - všechny odmaskované porty musí být aktivní

AND/$\overline{OR}$ = 0 - pouze některý z odmaskovaných portů musí být aktivní

H/$\overline{L}$ - definování aktivní logické úrovně portů

H/$\overline{L}$ = 1 - jedničková aktivní úroveň

H/$\overline{L}$ = 0 - nulová aktivní úroveň

M - maskovací bit

M = 1 - maska

M = 0 - odmaskováno

Je-li nastaven M = 1 musí následovat řídicí slovo masky.

Bity D4, D5 a D6 definují v režimu 3 podmínky pro generaci přerušovacího signálu $\overline{INT}$. Za stavu D6 = 1 je uplatněna funkce AND nemaskovatelných bitů kanálu. Logickou aktivitu těchto nemaskovatelných bitů programuje bit H/$\overline{L}$. Pozice D6 = 0 definuje logickou funkci OR nemaskovatelných bitů. Bity, které nemají být vtaženy do funkce AND nebo OR, lze maskovat samostatným řídicím slovem, které musí bezprostředně následovat za výše uvedeným řídicím slovem. V přerušovacím řídicím slově musí být ale nastaven bit M = 1, tj. příznak masky.

Formát řídicího slova masky

Profil řídicího slova masky je uložen v osmibitovém registru masky.

D7	D6	D5	D4	D3	D2	D1	D0
MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0

maskování jednotlivých bitů brány

MB0 ÷ MB7 - maskovací bit odpovídající portu A0 ÷ A7 nebo B0 ÷ B7 dle stavu signálu B/$\overline{A}$.

MB = 1 - maska
MB = 0 - odmaskováno
Nulové hodnoty v řídicím slově masky označují konkrétní bity brány, které mohou vyvolat přerušení.

Pro definování bitu povolujícího přerušení je možné využít zjednodušeného řídicího slova se zcela jinou adresou. Nejvyšší bit EI povoluje nebo blokuje přerušení v naprogramovaném režimu práce. Použitím tohoto slova se nemění ostatní podmínky pro vznik žádosti přerušení.

D7	D6	D5	D4	D3	D2	D1	D0
EI	X	X	X	0	0	1	1

povolení přerušení

adresa zjednodušeného přerušovacího slova

EI = 1 - přerušení povoleno
EI = 0 - přerušení blokováno

4.2 Z80-SIO

Dvoukanálový programovatelný sériový vstup/výstupní obvod Z80-SIO zajišťuje přenos v asynchronním režimu,případně v synchronním režimu v protokolech MONOSYNC, BISYNC, HDLC, SDLC. Synchronní přenos lze vést s vnitřní nebo vnější synchronizací. Veškeré synchronní protokoly je možné zabezpečit CRC polynomem. Základem řadiče SIO je sérioparalelní a paralelněsériový převodník.

Výrobce dodává čtyři modifikace pouzdra pod označením Z80-SIO/0, Z80-SIO/1, Z80-SIO/2 a Z80-SIO/9.

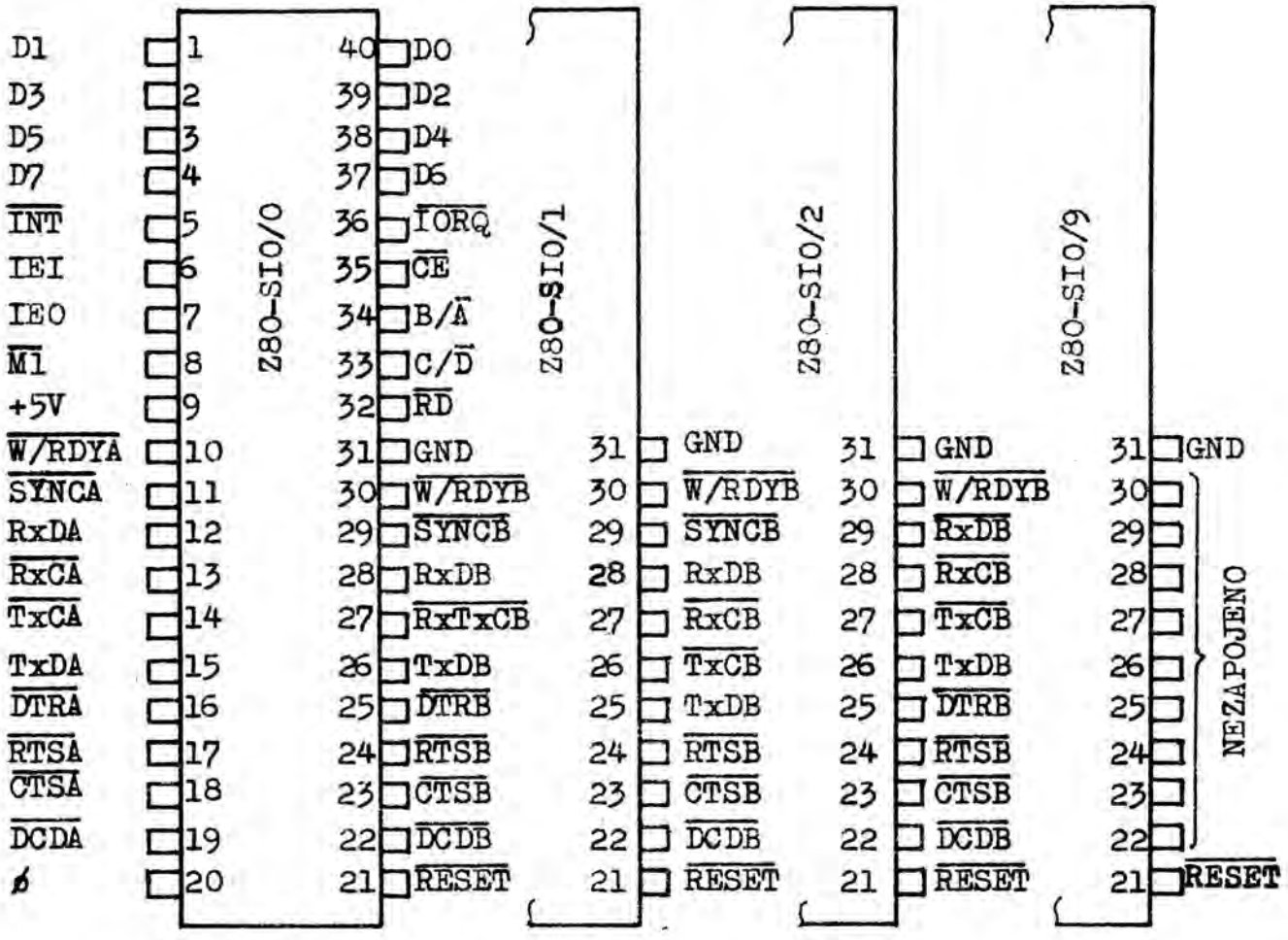
Typ Z80-SIO/0 má společné hodiny $\overline{\text{RxTxCB}}$ kanálu B pro vysílací a přijímací cestu.

Typ Z80-SIO/1 neobsahuje modemový výstupní signál kanálu B $\overline{\text{DTRB}}$.

Typ Z80-SIO/2 nemá vyveden synchronizační vodič $\overline{\text{SYNCB}}$ pro zajištění vnější synchronizace kanálu B.

Typ Z80-SIO/9 nemá zapojeny vývody kanálu B.

Zapojení čtyř modifikací pouzder Z80-SIO je na obr. 78. Ekvivalent vyráběný v NDR pod označením UB856D odpovídá typu Z80-SIO/0.



Obr. 78 Vyráběné varianty pouzder Z80-SIO

Stručná charakteristika pouzdra:

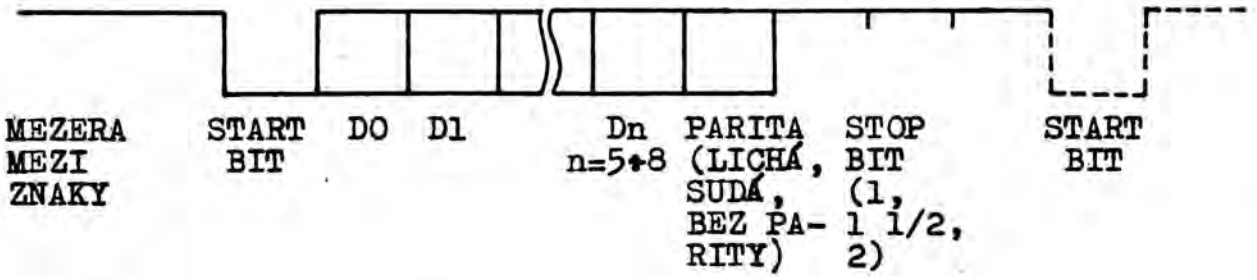
- dva duplexní sériové kanály
- vysílací cesta obsahuje 2 vyrovnávací registry znaků
- přijímací cesta obsahuje 4 vyrovnávací registry znaků
- přenos znaků v synchronních a asynchronních režimech s rychlostí 0 ÷ 550 kb/s při systémových hodinách 2,5 MHz u Z80-SIO
- přenos znaků v synchronních a asynchronních režimech s rychlostí 0 ÷ 880 kb/s při systémových hodinách 4 MHz u Z80A-SIO

- 8 řídicích registrů
- 3 stavové registry
- každý kanál disponuje 4 řídicími signály pro zajištění funkce modemu
- generování přerušení v rámci kaskádního sériového řetězce s dodáním vektoru přerušení
- čtení stavu sériového kanálu a stavu modemu
- možnost generování požadavku na zahájení DMA přenosu
- samostatné řízení modemů pro kanál A a B
- možnost vytváření sériových řetězců z obvodů Z80-SIO
- asynchronní režim:
  - přenos 5, 6, 7, 8 bitových znaků
  - parita lichá, sudá nebo bez parity
  - 1, 1,5, 2 stop bity
  - frekvence hodin vůči přenosové frekvenci 1, 16, 32, 64:1
  - na příjmu indikace chyb parity, formátu přenosu a přeběhu tj. PARITA ERROR, FRAMING ERROR a OVERRUN ERROR
  - indikace a generování BREAK signálu
  - možnost programování pomocného výstupu $\overline{\text{SYNC}}$
- synchronní režim s vnější synchronizací:
  - synchronizace zajištěna vstupním signálem $\overline{\text{SYNC}}$
  - kontrola přenosu CRC polynomem
  - frekvence hodin vůči přenosové frekvenci 1 : 1
- synchronní režim s vnitřní synchronizací (MONOSYNC nebo BISYNC):
  - 1 nebo 2 synchronizační znaky
  - generování a kontrola přenosu CRC polynomem
  - automatické vkládání synchronizačních znaků do vysílací cesty
  - frekvence hodin vůči přenosové frekvenci 1 : 1
- synchronní režim s bitově orientovaným protokolem (HDLC a IBM SDLC):
  - synchronizace přenosu pomocí počátečního FLAG formátu
  - automatické vkládání a výmaz nul v přenosu na úrovni Z80-SIO
  - automatické vysílání příznaku FLAG na začátku a konci bitově orientovaného protokolu
  - generování adresy ve vysílání
  - testování příznaku FLAG a adresy v přijímací cestě
  - generování a testování CRC polynomu při přenosu

4.2.1 Přenosové protokoly

Asynchronní přenos (ASYNCHRONOUS DATA LINK PROTOCOL)

Používá se pro zabezpečení znakově orientovaného přenosu, např. pro terminály, telefonní sítě. Přenos každého znaku je na začátku synchronizován jedním start bitem, konec přenášeného znaku se uzavírá 1, 1,5, 2 stop bity. Znak může nabývat formátu 5 ÷ 8 bitové informace a lze jej zajistit naprogramováním liché nebo sudé parity, případně paritu zcela potlačit. Na obr. 79 je formát asynchronního znaku.

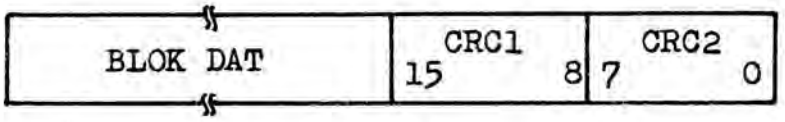


Obr. 79 Formát asynchronního přenosu

Vyslání zprávy nastane po dodání znaku z mikroprocesoru, je-li vysílací cesta uvolněna nulovým signálem $\overline{CTS}$ nebo programově. Podobně přijímací cestu lze uvolnit programově nebo vstupním modemovým signálem $\overline{DCD}$. Detekce rozběhového tzv. start bitu aktivního do nuly v přijímací cestě probíhá v polovině jeho délky. Tímto způsobem se značně potlačuje rušení na příjmu. Přijímací cesta obsahuje 4 vyrovnávací registry pro přijímaná data, a tím má Z80-CPU větší časovou reservu pro započetí obsluhy přerušení. Chyba parity a chyba přeběhu je pamatována v přijímací cestě od doby vzniku až po program. znulování. Chyba formátu přenosu, tj. počtu stop bitů, se automaticky přepisuje **každým dalším přijmutým znakem.**

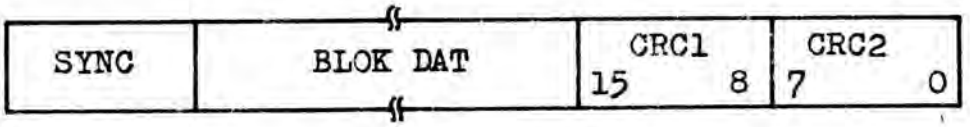
Synchronní přenos (SYNCHRONOUS DATA LINK PROTOCOL)

a) Synchronní přenos s vnější synchronizací
Synchronizace přenosu je zajištěna pouze aktivitou vstupního signálu $\overline{\text{SYNC}}$ viz obr. 80. Přenos zabezpečuje CRC polynom. Synchronizační znaky nejsou generovány ani detekovány.



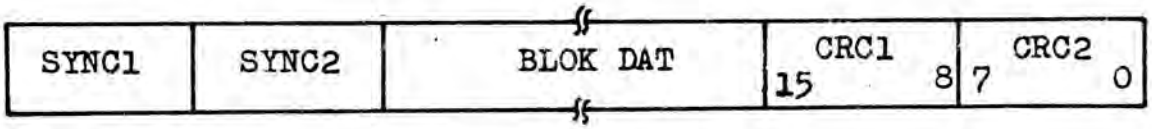
Obr. 80 Formát synchronního přenosu s vnější synchronizací

b) Synchronní přenos s vnitřní synchronizací - MONOSYNC (MONO SYNCHRONOUS DATA LINK PROTOCOL)
Přenos je na začátku přenášeného bloku dat synchronizován jedním osmibitovým synchro znakem. Kontrolu přenosu provádí CRC polynom za blokem znaků viz obr. 81.



Obr. 81 Synchronní přenos - MONOSYNC

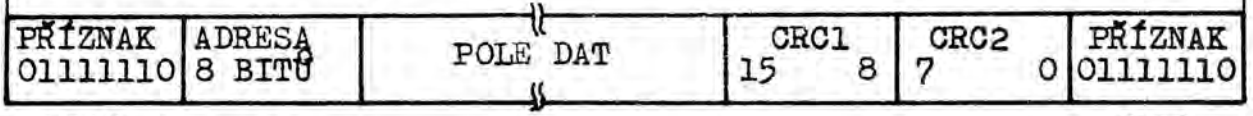
c) Synchronní přenos s vnitřní synchronizací - BISYNC (BINARY SYNCHRONOUS DATA LINK PROTOCOL)
Odlišnost tohoto protokolu oproti MONOSYNC přenosu je pouze v zabezpečení začátku přenosu dvěma osmibitovými synchronizačními znaky viz obr. 82.



Obr. 82 Synchronní přenos - BISYNC

d) Synchronní přenos SDLC a HDLC (SYNCHRONOUS DATA LINK CONTROL AND HIGH LEVEL DATA LINK CONTROL)
Protokol HDLC byl normalizován mezinárodní normalizační komisí ISO/CCITT a slouží k spojení dle normy ISO X.25. Protokol SDLC byl vyvinut firmou IBM pro realizaci přenosových sítí. Oba protokoly jsou bitově orientované, kde synchronizaci zajišťuje na začátku FLAG znak, tj. příznak o profilu 01111110, za kterým následuje osmibitová adresa, pole dat, šestnáctibitový kontrolní součet CRC a opět FLAG znak stejného profilu jako na začátku zprávy, viz obr. 83. Programovatelný řadič těchto protokolů nabízí také firma Intel pod označením I 8273.

RÁMEC - základní přenosová jednotka dat u SDLC a HDLC přenosu



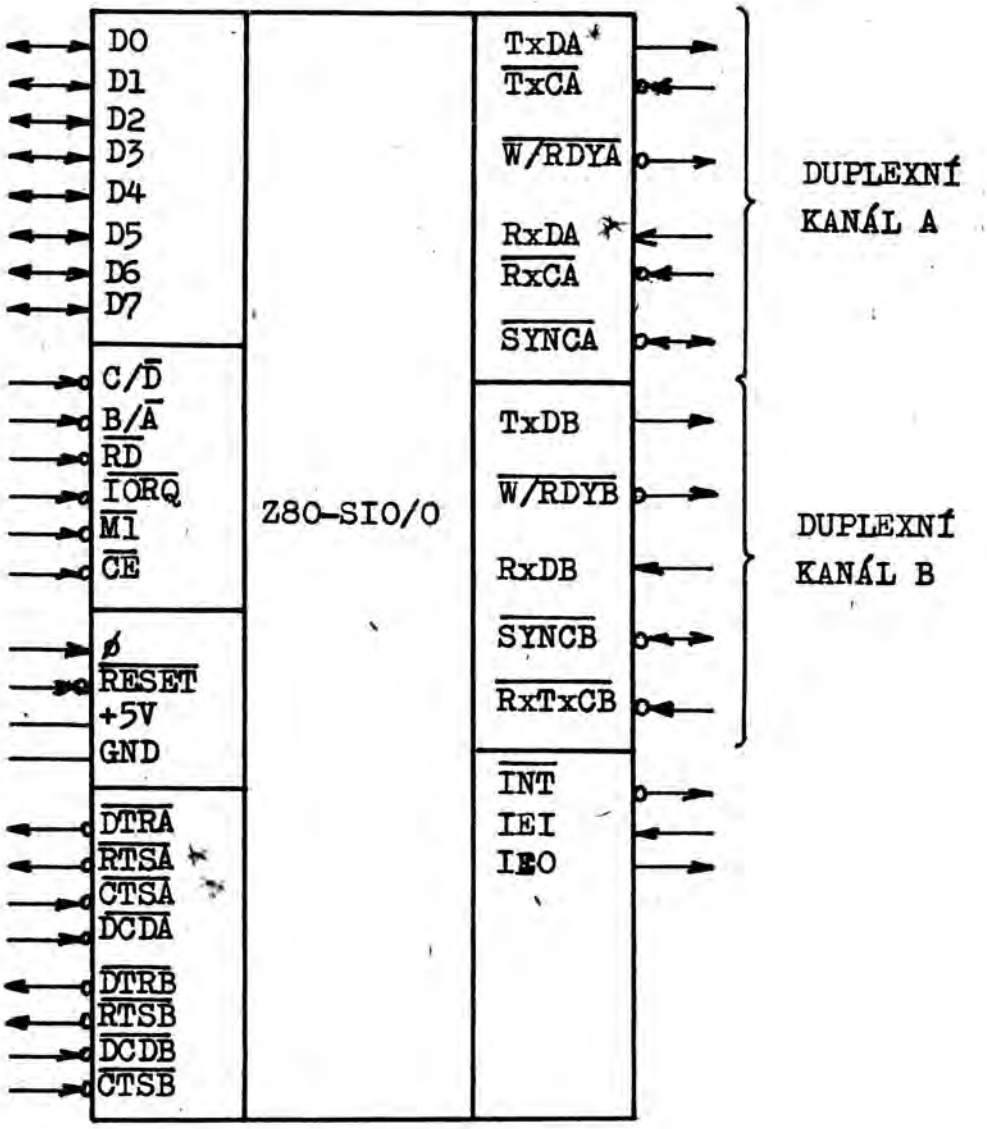
Obr. 83 Synchronní SDLC, HDLC protokol

4.2.2 Popis vývodů pouzdra

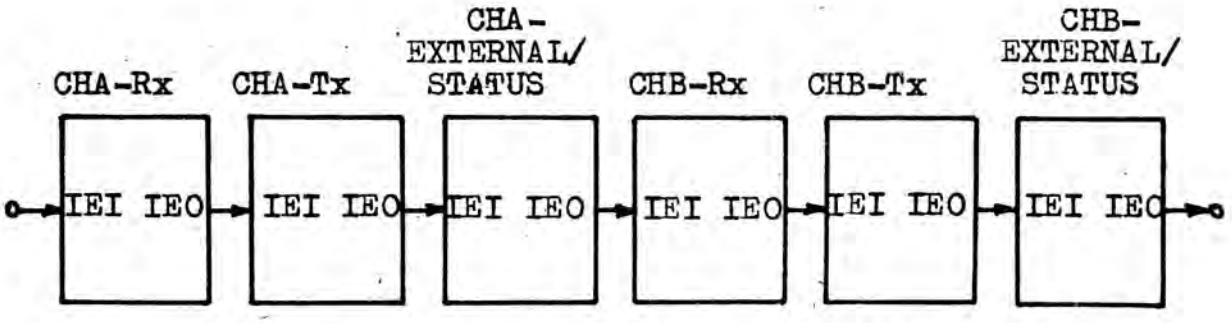
Schematická značka na obr. 84 poskytuje zapojení základní varianty sériového adaptoru Z80-SIO/0 se společnými hodinami kanálu B. Rovněž popis vývodů se týká této varianty, která je ekvivalentní dováženému prvku z NDR U856D.

D0 ÷ D7 datová sběrnice	třístavová obousměrná vstup/výstupní sběrnice zajišťuje přenos řídicích a datových slov mezi procesorem a Z80-SIO.
C/$\overline{D}$ výběr typu přenosu	stav signálu informuje o charakteru přenášených dat. Je-li vstup na úrovni H, jsou informace přenášené po datové sběrnici považovány za řídicí slova. Je-li vstup na úrovni L, jsou považovány za data.

B/$\overline{A}$ výběr kanálu	vstupní signál aktivující bránu A nebo B. Úroveň H aktivuje bránu B, úroveň L aktivuje bránu A.
$\overline{M1}$ prvý strojový cyklus	význam tohoto signálu se váže na stav signálů $\overline{RD}$ a $\overline{IORQ}$.
$\overline{RD}$ čtení	vstupní signál obvodu SIO. Sériový adaptor signálu $\overline{RD}$ využívá společně s $\overline{M1}$ k návratu z podprogramu přerušení pomocí instrukce RETI a negace $\overline{RD}$ k indikaci cyklu zápisu.
$\overline{IORQ}$ žádost na V/V zařízení	řídicí vstupní signál. Signálu $\overline{IORQ}$ se dohromady se signály $\overline{CE}$, B/$\overline{A}$, C/$\overline{D}$ a $\overline{RD}$ používá k přenosu řídicích slov dat a stavů do a z obvodu SIO. Obvod nemá řídicí vodič $\overline{WR}$, takže cyklus zápisu je definován neaktivním signálem $\overline{RD}$. Je-li $\overline{IORQ}$ = 0 a $\overline{M1}$ = 0,jde o přerušovací cyklus.
$\overline{CE}$ výběr obvodu	signál $\overline{CE}$ řídí aktivování obvodu jako celku při úrovni L.
ø systémové hodiny	vstup pro připojení systémových hodin. Provádí interní synchronizaci signálů.
$\overline{RESET}$ nulování	aktivní nulová úroveň zabezpečí inicializaci obvodu SIO. Zablokuje vysílač i přijímač. Výstupy vysílačů TxDA, TxDB jsou na úrovni H, rovněž veškeré modemové signály jsou neaktivní tj. na úrovni H. Po resetovací funkci je nutné obnovit stav řídicích registrů před zahájením datového přenosu. Po dobu nulování jsou zablokována všechna přerušení.
IEI povolení generování přerušení	slouží pro vytváření sériových řetězců.
IEO povolení generování přerušení	opět signál vytvářející řetězce. Prioritní vazbu mezi kanálem A a B a jejich přijímací, vysílací a stavovou částí vyjadřuje obr. 85.



Obr. 84 Schematická značka Z80-SIO/0



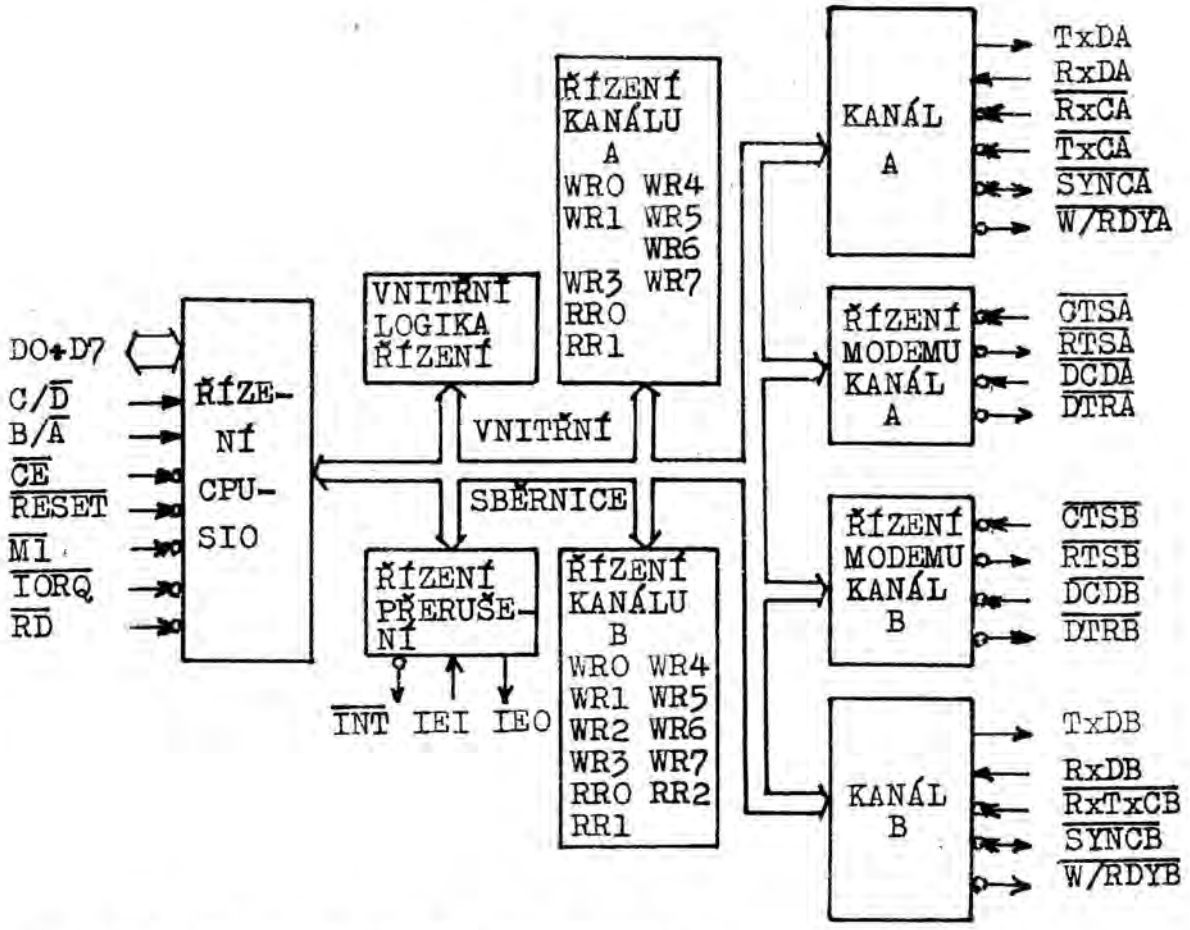
Obr. 85 Vnitřní prioritní vazba obvodu Z80-SIO

$\overline{INT}$ přerušení	výstup na otevřeném kolektoru. Žádost o přerušení při stavu $\overline{INT}$ = 0
$\overline{W/RDYA}$ a $\overline{W/RDYB}$ zpomalení práce Z80-CPU/Z80-SIO připraven pro vedení DMA přenosu	výstupní signály na otevřeném kolektoru ve funkci $\overline{WAIT}$ a na úrovni TTL ve funkci $\overline{READY}$. Výstupu se používá pro synchronizaci práce SIO a mikroprocesoru ($\overline{WAIT}$) nebo pro synchronizaci práce SIO a řadiče DMA ($\overline{READY}$). Při $\overline{RESET}$u je signál v odpojeném stavu .
$\overline{CTSA}$ a $\overline{CTSB}$ uvolnění vysílací cesty	dva vstupy pro uvolnění přenosu vysílače v režimu automatického povolení vysílání, který povoluje aktivní úroveň $\overline{CTSA}$=$\overline{CTSB}$=0. Při nenaprogramovaném automatickém povolení vysílání se tyto vstupy používají jako obecné vstupy. Z80-SIO pak tyto signály testuje a žádá o přerušení při změně jejich logické úrovně.
$\overline{DCDA}$ a $\overline{DCDB}$ uvolnění přijímací cesty	významově shodné vstupy jako signály $\overline{CTSA}$, $\overline{CTSB}$ s tím rozdílem, že řídí uvolnění přijímací cesty při své nulové úrovni.
$\overline{RTSA}$ a $\overline{RTSB}$ požadavek na vysílání	výstupní signály s aktivní nulovou úrovní. Těmito signály se upozorňuje protější strana (přijímač), že se budou vysílat zprávy ($\overline{RTS}$ je přivedeno do $\overline{DCD}$). V asynchronním režimu, je-li bit řídicího slova RTS = 1, je vývod $\overline{RTS}$ uveden na nulu a zůstane na této úrovni až do okamžiku, kdy registry vysílače jsou prázdné (zpráva je odvysílána). V synchronním režimu výstup $\overline{RTS}$ sleduje přesně stav bitu RTS řídícího slova.
$\overline{DTRA}$ a $\overline{DTRB}$ přijímač připraven	výstupní signály s aktivní úrovní L. Výstup $\overline{DTR}$ se propojuje do vstupu $\overline{CTS}$ protější přijímací strany. Signál $\overline{DTR}$ uvolňuje vysílač protější strany. Výstup $\overline{DTR}$ kopíruje nastavení bitu DTR řídícího slova.

RxDA a RxDB vstupy přijímačů	signály RxD jsou vstupy přijímacího sériového datového kanálu s aktivní úrovní H.
TxDA a TxDB výstupy vysílačů	signály TxD jsou výstupy vysílacího sériového datového kanálu s aktivní úrovní H.
$\overline{\text{TxCA}}$ hodinový kmitočet	hodinový kmitočet vysílače kanálu A. Dělící poměr mezi hodinovým vstupem a přenosovou frekvencí je programově volitelný v asynchronním režimu 1, 16, 32, 64 : 1. Výstup dat vysílače TxDA aktivuje své úrovně se závěrnou hranou hodin $\overline{\text{TxCA}}$.
$\overline{\text{RxCA}}$ hodinový kmitočet	hodinový kmitočet přijímače kanálu A. Dělící poměr pro asynchronní režim podléhá stejným pravidlům jako v předcházejícím případě. Přijímací vstup RxDA je vzorkován náběžnou hranou hodin $\overline{\text{RxCA}}$.
$\overline{\text{RxTxCB}}$ společný hodinový kmitočet	společné hodiny kanálu B. Společný hodinový kmitočet kanálu B je určen pro vysílací a přijímací cestu. Vzhledem k 40 vývodovému pouzdru má Z80-SIO/O společné hodiny pro kanál B. V asynchronním režimu může být poměr hodinového kmitočtu k přenosové rychlosti v poměru 1, 16, 32, 64 : 1. Pro zajištění hodinových vstupů kanálu A a B se nejčastěji využívají výstupy podpůrného obvodu Z80-CTC.
$\overline{\text{SYNCA}}$ a $\overline{\text{SYNCB}}$ synchronizační signály	v asynchronním režimu jsou tyto signály vstupy a chovají se obdobně jako vstupy $\overline{\text{CTS}}$ a $\overline{\text{DCD}}$. Úroveň synchronizačních signálů lze číst ve stavovém slově RR0. Při synchronním režimu s vnější synchronizací jsou signály $\overline{\text{SYNC}}$ také vstupy. Při synchronním režimu s vnitřní synchronizací pracují signály $\overline{\text{SYNC}}$ jako výstupy. Svou nulovou úrovní signalizují příchod synchronizačního znaku nebo synchronizačních znaků v režimu BISYNC.

4.2.3 Architektura obvodu

Spojení se systémovou adresní, datovou a řídicí sběrnicí řeší blok řízení CPU-SIO, který pomocí vnitřní sběrnice komunikuje se všemi bloky obvodu. Blok řízení přerušení umožňuje vytváření jednak vnitřního a jednak vnějšího přerušovacího sériového řetězce. Vnější priorita je dána začleněním obvodu do kaskádního řetězce, kdežto vnitřní je pevně dána interním nadefinováním. Nejvyšší prioritu přerušení má přijímací cesta kanálu A a priorita postupně klesá přes vysílací cestu kanálu A, změnu stavu kanálu A, přijímací cestu kanálu B, vysílací cestu kanálu B a změnu stavu tohoto kanálu. Protože existuje v obvodu Z80-PIO pouze jeden vektor přerušení, zabezpečí výše uvedené vnitřní požadavky přerušení modifikování tohoto vektoru na bitech D1, D2, D3. Blokové schéma obvodu Z80-SIO je na obr. 86.



Obr. 86 Blokové schéma Z80-SIO/0

Programování činnosti kanálu A a B je samostatné a zabezpečuje se 8 zápisovými řídícími slovy WR0 + WR7. Stav jednotlivých kanálů lze zjistit vyčtením 3 stavových registrů RR0 + RR2. Platí zásada, že společný vektor přerušení je zapisován do řídicího registru WR2 kanálu B a jeho obsah může být čten do Z80-CPU jako obsah stavového registru RR2 zase jen z kanálu B. Datová komunikace v příslušném protokolu je vedena přes blok kanálu A a B. Vysílací část kanálu obsahuje jeden paralelní a jeden výstupní posuvný sériový registr. Přijímací část kanálu má sice jeden vstupní posuvný sériový registr, ale tři bytové paralelní registry, které pracují systémem FIFO. Kanály mají generátory CRC polynomů a obvody zajišťující kontrolu tohoto zabezpečení přenosu v přijímací cestě. Modemové signály generuje a zpracovává blok řízení modemu kanálu A a B. Vnitřní logika řízení provádí formátování, synchronizaci a kontroly v přijímací cestě.

4.2.4 Spolupráce obvodu s Z80-CPU a s Z80-DMA

Existují tři způsoby začlenění obvodu Z80-SIO do mikroprocesorového systému:

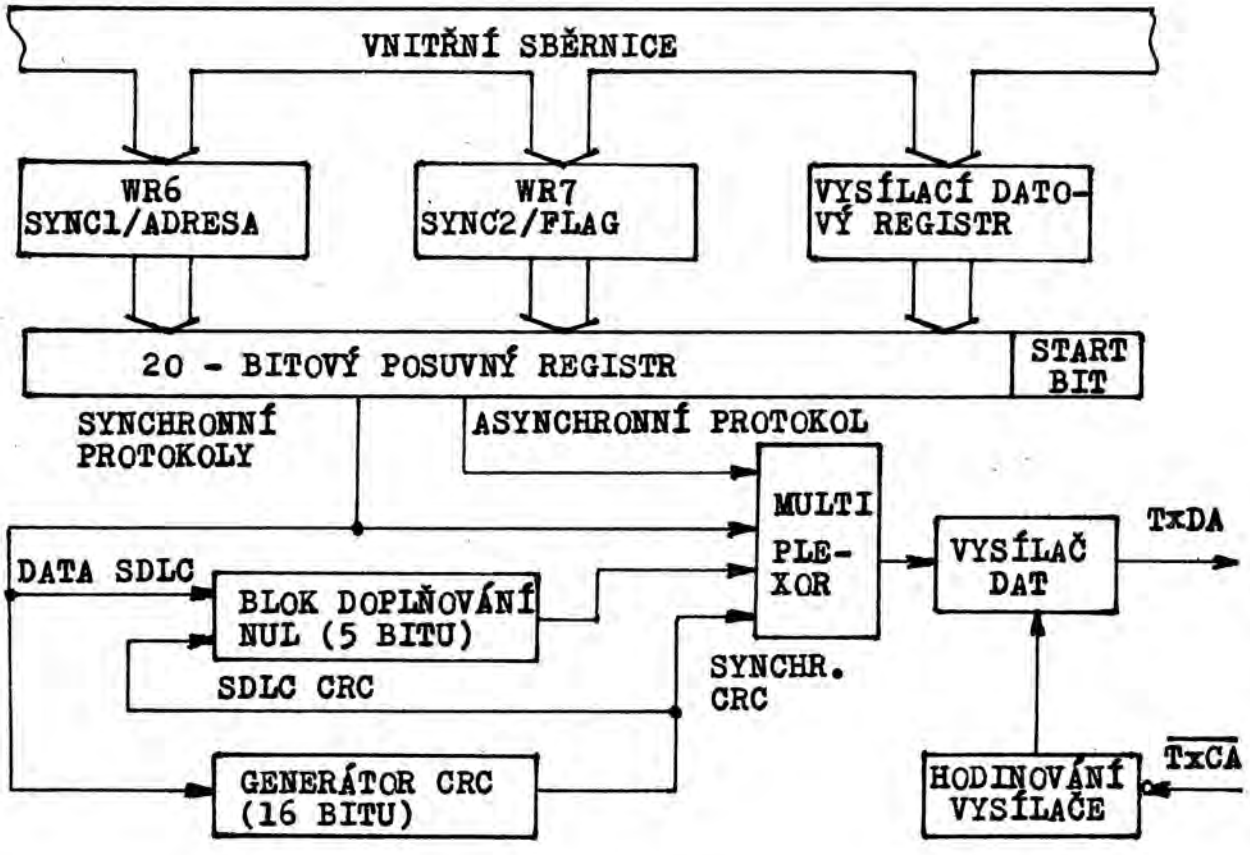
a) Synchronizace spolupráce sériového řadiče s mikroprocesorem je zajištěna testováním stavových registrů RR0 a RR1. Tato dotazovací forma je časově nejnáročnější.

b) Optimálnější synchronizace práce Z80-SIO s mikroprocesorem je řešena přes přerušovací signály. Přerušení je vyvoláno příslušným odblokovaným uzlem v rámci vnitřního prioritního řetězce přerušení.

c) Nejvýhodnější řešení získáme spojením řadiče SIO s DMA obvodem. Přenos je pak veden přímo mezi periferií spojenou se sériově paralelním převodníkem a oblastí paměti bez účasti mikroprocesoru za řízení řadiče Z80-DMA. Sériový řadič musí mít naprogramovan výstup $\overline{W/RDY}$ ve významu $\overline{RDY}$. Tímto výstupem žádá řadič DMA o zahájení fyzického přenosu.

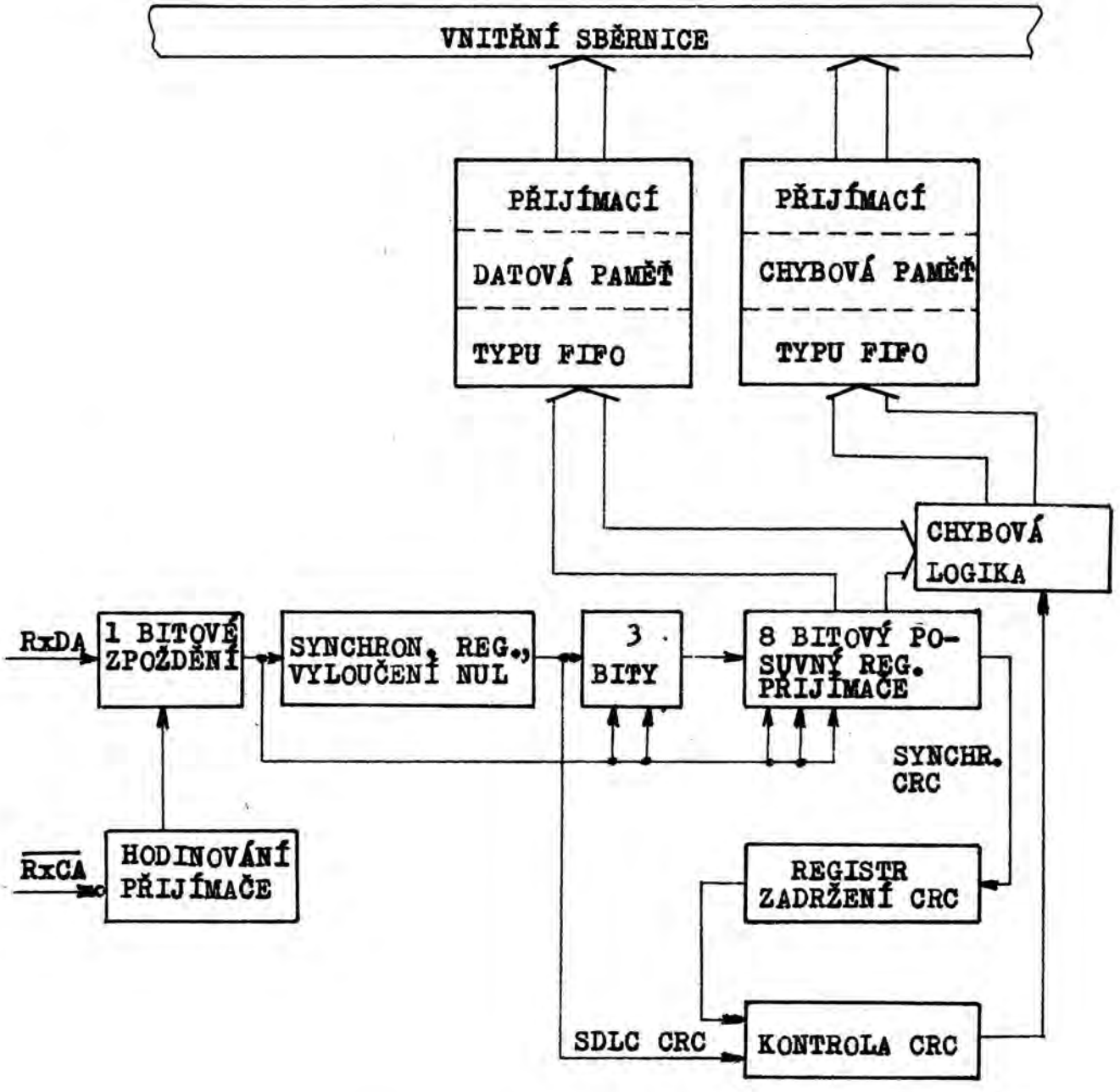
Vysílací část kanálu A

Vysílací část Z80-SIO má tři paralelní osmibitové registry a to vysílací datový, WR6, WR7 a 20bitový posuvný registr. Registr WR6 a WR7 obsahuje synchronizační znaky v režimu s jednoduchou popřípadě zdvojenou synchronizací nebo adresu a FLAG příznak v režimu SDLC/HDLC. Během synchronních režimů je obsah WR6 a WR7 zapisován do posuvného registru na začátku každého přenášeného souboru, stejně jako časová výplň uprostřed souboru, kdy CPU nedodá včas data. V SDLC režimu je příznak začátku a konce souboru přepisován do posuvného registru před a po skončení přenosu dat. Asynchronní přenos dat se formuje v posuvném registru včetně START a STOP bitů a je sériově vysílán rychlostí, která byla programována vnitřním děličem vnějšího hodinového kmitočtu. Pro synchronní přenos musí být dělič vstupního přenosového kmitočtu vyřazen.



Obr. 87 Struktura vysílače Z80-SIO

V protokolu SDLC/HDLC jsou data vysílána přes blok doplňování nul. Tyto nuly jsou doplňovány mimo okamžiku vysílání FLAG příznaku ve všech ostatních polích RÁMCE. Nula je doplněna vždy, když následuje pět jedničkových bitů za sebou ve vysílací cestě. Struktura vysílače je zachycena na obr. 87.



Obr. 88 Struktura přijímače Z80-SIO

Přijímací část kanálu A

Přijímací cesta má trojnásobnou vyrovnávací paměť dat a chyby přenosu typu FIFO a jeden osmibitový posuvný registr přijímače. Takovéto uspořádání získává čas pro obsloužení přerušení v CPU. Trojnásobná vyrovnávací paměť chyby přenosu pamatuje chybu parity, formátu a přeběhu atd. pro každé ze tří datových slov uložených ve vyrovnávací paměti přijímače typu FIFO. Datová sériová informace prochází různými cestami dle programovaného typu přenosového protokolu.

V asynchronním režimu sériová data vstupují do 3bitové vyrovnávací paměti, pokud mají délku 7 nebo 8 bitů. Jinak vstupují přímo do osmibitového posuvného registru při délce 5 nebo 6 bitů.

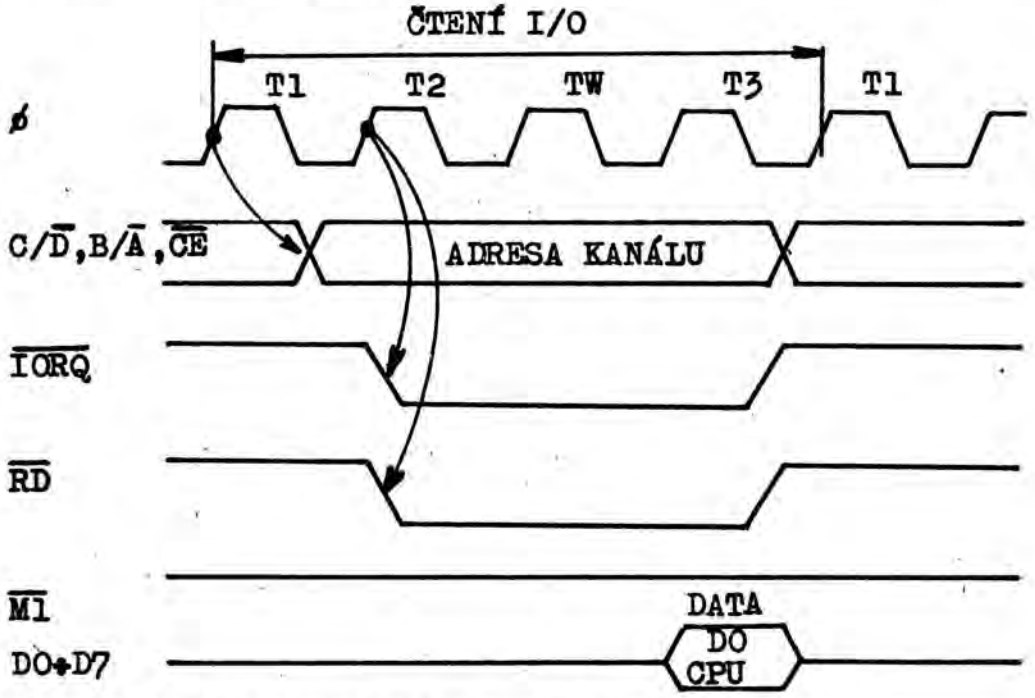
V synchronních protokolech je forma zpracování vstupní informace určena fází přenosu. Na samém začátku ve vyhledávacím režimu synchronizačního slova vstupní pulsní posloupnost vstupuje do registru pro porovnávání přijatého slova s programovaným synchronizačním znakem či znaky nebo FLAG příznakem v SDLC/ /HDLC režimu. Pokud je zařízení naprogramováno s jednoduchým synchronizačním znakem, hledá se shoda se znakem v WR6 registru. Při zdvojené synchronizaci se žádá shoda se znakem v registru WR6 a pak teprve s registrem WR7. V obou případech prochází vstup informací synchronizačním registrem přijímače.

Při příjmu v SDLC/HDLC režimu nejdříve prochází vstup dat synchronizačním registrem přijímače, který stále sleduje přijímanou posloupnost bitů, ze které je připraven vypustit 0,pokud následuje za pěti bity jedniček. Pokud šestý bit je rovněž jednička, testuje sedmý bit. Pokud je po šesti jedničkových bitech bit 0 indikuje se příznak FLAG. Překódovaná vstupní data procházejí 3 bitovou vyrovnávací pamětí do posuvného registru přijímače. Přenos SDLC začíná porovnáváním přicházejících znaků s obsahem registru WR7, kde je uložen FLAG příznak. Jakmile je příznak vyhledán, všechna další data prochází stejnou cestou bez ohledu na jejich bitovou délku znaků. Struktura přijímače Z80-SIO je na obr. 88.

4.2.5 Časové rozbory základních signálů

Komunikace obvodu Z80-SIO probíhá na sběrnici ve čtyřech typech strojových cyklů:

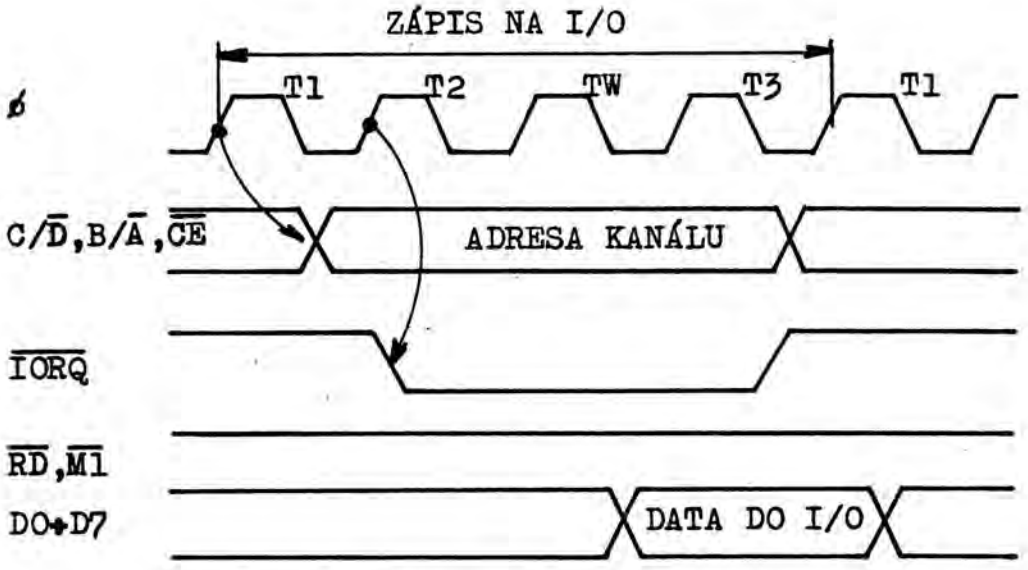
a) cyklus čtení - v tomto strojovém cyklu je ze stykového obvodu přenášena do procesoru datová nebo stavová informace na základě aktivity signálu C/$\overline{D}$. Z kterého kanálu přenos nastává, určuje výběrový vodič B/$\overline{A}$. Do cyklu čtení je automaticky včleněn takt TW za takt T2 bez ohledu na stav signálu $\overline{WAIT}$. Časový rozbor čtecího cyklu je na obr. 89.



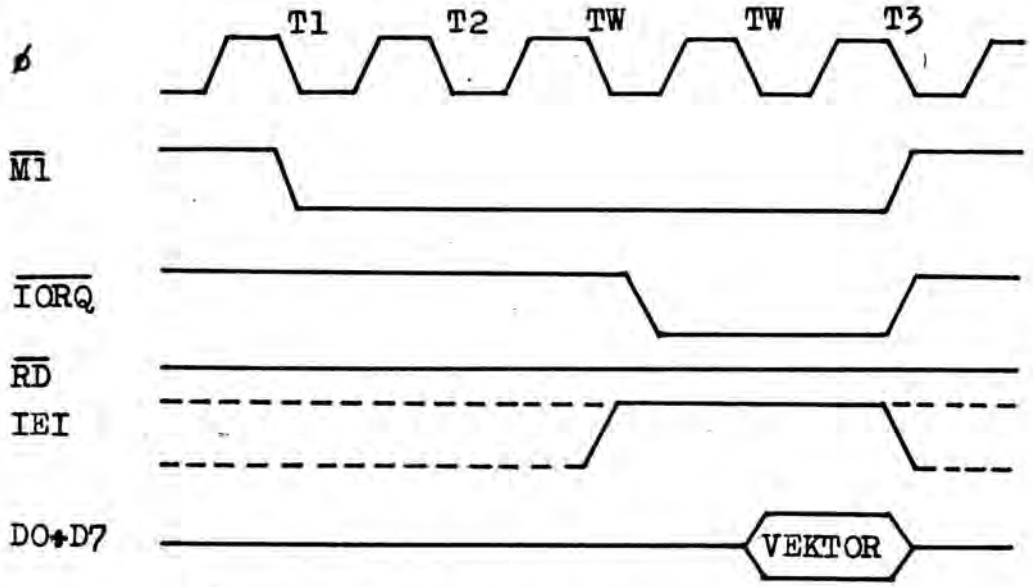
Obr. 89 Strojový cyklus čtení

b) cyklus zápisu - zajišťuje dodání dat nebo řídicích slov do periferního obvodu na základě stavu signálu C/$\overline{D}$. Vybraný kanál akceptuje údaje v taktu T3. Před taktem T3 předchází takt TW bez ohledu na úroveň vstupu $\overline{WAIT}$ viz obr.90.

c) cyklus vyvolání přerušení - časový rozbor na obr.91 ukazuje vstup do přerušovacího podprogramu při nastaveném přerušovacím režimu 2 v Z80-CPU. Vlivem naprogramovaného režimu 2 je v T3 čten Z80-SIO vektor přerušení jako nižší řády adresy celkového šestnáctibitového vektoru. Cyklus přerušení automaticky doplňuje mikroprocesor dvěma čekacími takty TW.



Obr. 90 Strojový cyklus zápisu



Obr. 91 Časový rozbor cyklu přerušení

d) cyklus návratu z přerušení - návrat z podprogramu přerušení je řešen standardně instrukcí RETI. Po příchodu druhého bytu instrukce RETI periferní obvod znovu aktivuje výstupní signál IEO, a tím umožňuje generování dalších požadavků přerušení nižších priorit od sériového řetězce. Časový rozbor návratu z podprogramu přerušení byl uveden při rozboru práce obvodu Z80-PIO.

4.2.6 Programování Z80-SIO

Programování obvodu probíhá samostatně pro kanál A a B pomocí adresního vodiče B/$\overline{A}$. Rozlišení řídicí informace od datové lokalizuje výběrový vodič C/$\overline{D}$. Přesný význam byl uveden v kap. 4.2.2. V dalším popisu budou řídicí slova označena symboly WR0 ÷ WR7, stavová slova symboly RR0 ÷ RR2. Řídicí registr WR0 obsahuje v pozici D0 ÷ D2 číslo registru neboli ukazatel řídicího registru, který bude plněn následujícím bytem přišlým po bytu WR0. Pouze po nulování (povelem nebo vstupem $\overline{RESET}$) probíhá zápis do registru WR0 nezávisle na ukazateli D0 ÷ D2.

Řídicí registr WR0

Mimo ukazatele na registr definuje několik variant nulování kanálu, povolení přerušení atd.

D7	D6	D5	D4	D3	D2	D1	D0	
					0	0	0	WR0
					0	0	1	WR1
					0	1	0	WR2
					0	1	1	WR3
					1	0	0	WR4
					1	0	1	WR5
					1	1	0	WR6
					1	1	1	WR7

ukazatel na řídicí registry

		0	0	0	bez vlivu (NULL CODE)
		0	0	1	zrušení rámce (v protokolu SDLC)
		0	1	0	nulování EXT/STATUS přerušení
		0	1	1	nulování lokalizovaného kanálu
		1	0	0	povolení přerušení od dalšího znaku
		1	0	1	nulování přerušení od vysílače
		1	1	0	nulování chyb v přenosu
		1	1	1	návrat z přer. (jako RETI) - kanál A
0	0	bez vlivu			
0	1	nulování CRC v přijímači			
1	0	nulování CRC ve vysílači			
1	1	nulování paměti vysílače - konce souboru			

Řídicí registr WR1

Nastavuje způsoby přenosů vysílače, přijímače a přerušení.

D7	D6	D5	D4	D3	D2	D1	D0

D0
- 1 povolení EXT/STATUS přerušení

D1
- 1 povolení přerušení od prázdného vysílače

D2
- 1 povolení modifikace vektoru přerušení kanálu B a A

D4 D3
- 0 0 blokování vzniku přerušení od přijímače
- 0 1 povolení přerušení od 1. znaku přijímače
- 1 0 povolení přerušení od každého znaku
- 1 1 povolení přerušení od každého znaku, generování fixního vektoru

D5
- 0 $\overline{W/RDY}$ = 0 (vysílač je plný)
- 1 $\overline{W/RDY}$ = 0 (přijímač je prázdný)

D6
- 0 $\overline{W/RDY}$ je ve funkci $\overline{WAIT}$
- 1 $\overline{W/RDY}$ je ve funkci $\overline{RDY}$

D7
- 0 povolení výstupu $\overline{W/RDY}$
- 1 výstup $\overline{W/RDY}$ setrvá v log. 1

Poznámka

Nastavení bitu D0 = 1 aktivuje přerušení při změně signálu $\overline{DCD}$, $\overline{CTS}$, $\overline{SYNC}$ nebo při vyslání CRC znaku a synchronizačního znaku při zakončení přenosu.

Na základě bitu D2 = 1 řídicího slova WR1 je zabezpečena modifikace vektoru přerušení.

Modifikace vektoru přerušení kanálu A

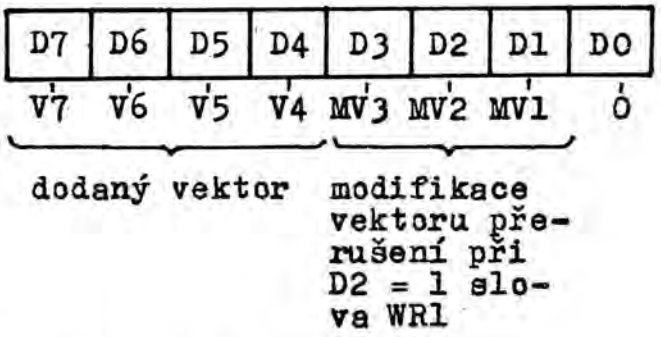
V3	V2	V1	
0	0	0	vysílací registry jsou prázdné
0	0	1	změna EXT/STATUS podmínek
0	1	0	znak byl přijmut
0	1	1	speciální podmínky přijímače

Modifikace vektoru přerušení kanálu B

V3	V2	V1	
1	0	0	vysílací registry jsou prázdné
1	0	1	změna EXT/STATUS podmínek
1	1	0	znak byl přijmut
1	1	1	speciální podmínky přijímače

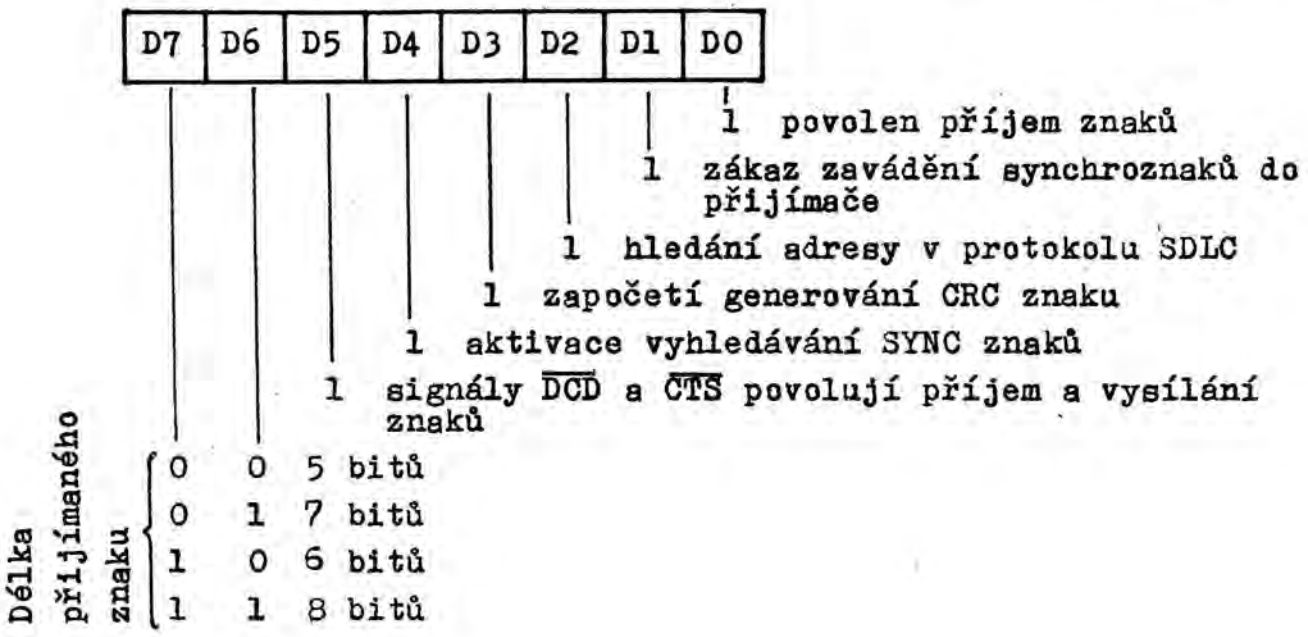
Řídicí registr WR2

Registr obsahuje dodaný vektor přerušení. Umístění registru je pouze v kanálu B. Bity D0, D4 ÷ D7 nelze ovlivnit vnitřní přerušovací podmínkou. Bity D1 ÷ D3 lze modifikovat dle výše uvedené tabulky, je-li bit D2 = 1 v řídicím slově WR1.



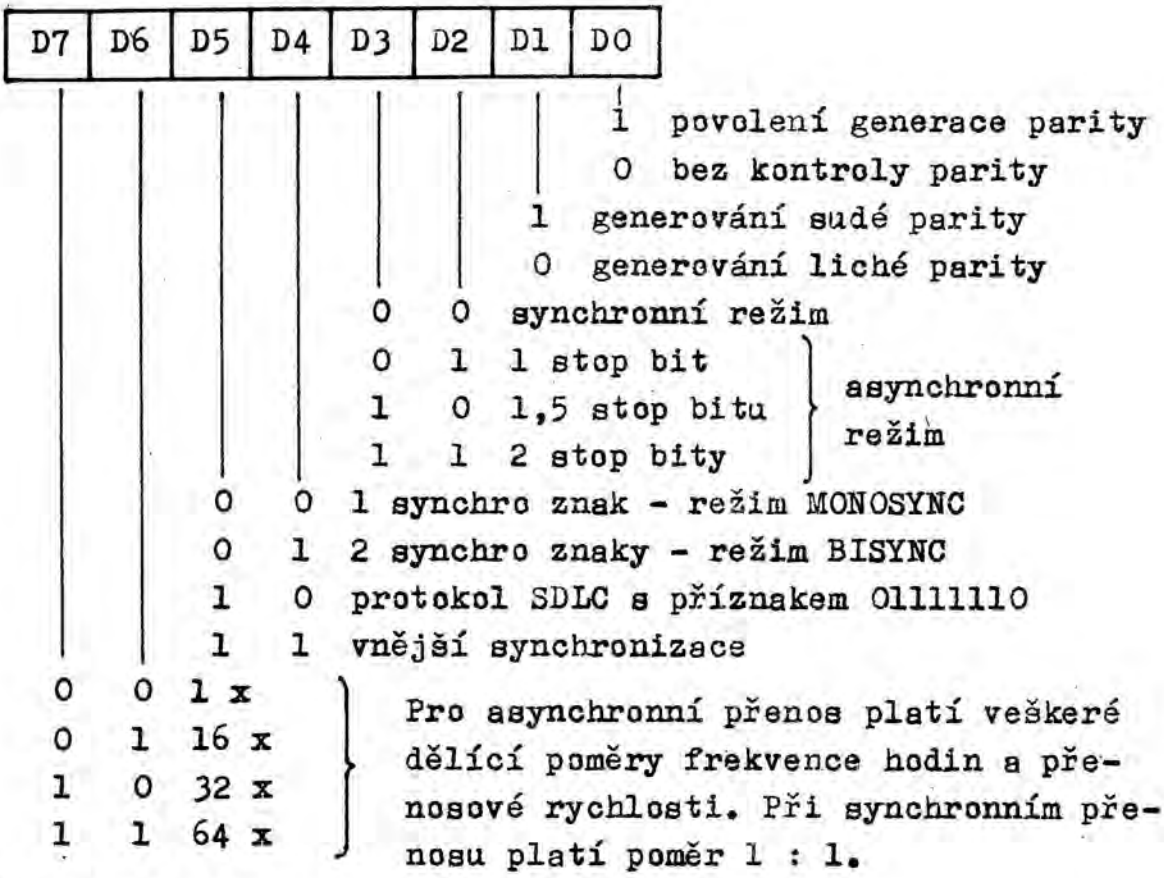
Řídicí registr WR3

Nastavení bitů řídicího registru WR3 řídí činnost přijímače.



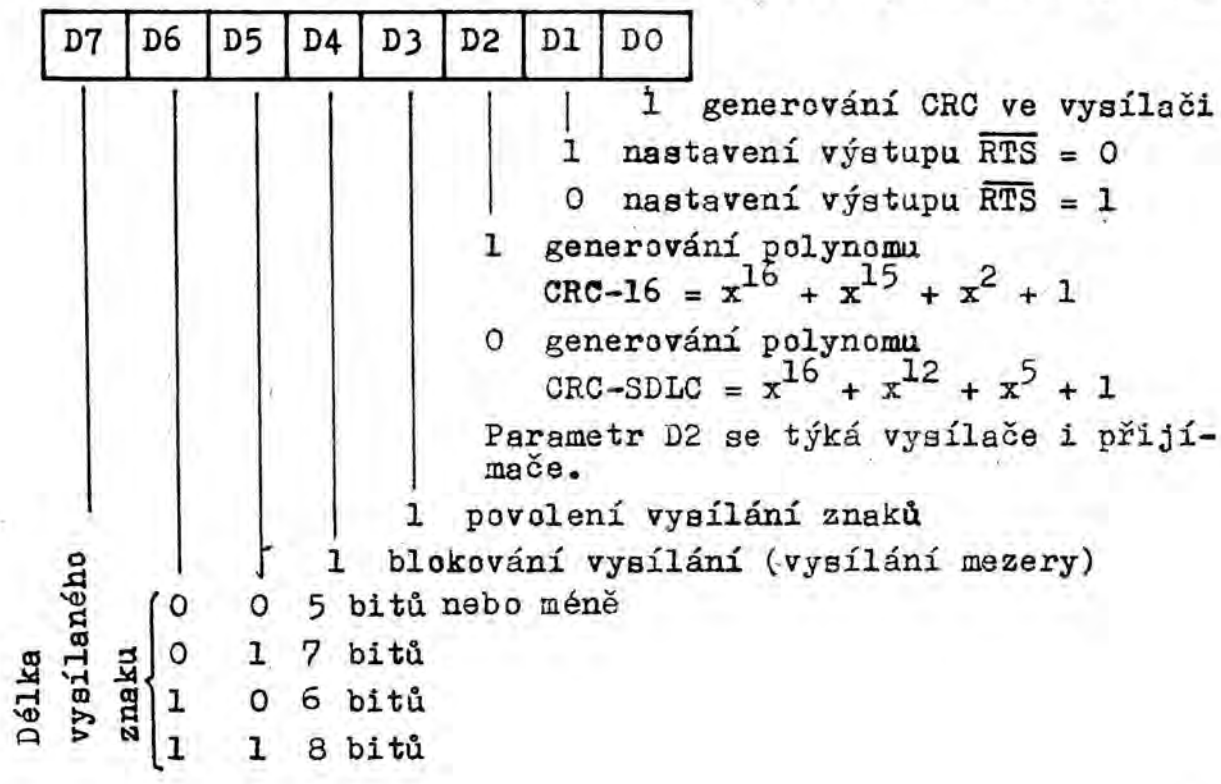
Řídicí registr WR4

Jednotlivé bity řídicího registru WR4 ovlivňují činnost vysílací a přijímací cesty.



Řídicí registr WR5

Obsah tohoto registru řídí činnost vysílače, ale i přijímače



D7	D6	D5	D4	D3	D2	D1	D0

1 nastavení výstupu $\overline{DTR}$ = 0
0 nastavení výstupu $\overline{DTR}$ = 1

Řídicí registr WR6

Řídicí registr WR6 obsahuje synchronizační znak v režimu MONOSYNC nebo první synchronizační znak v režimu BISYNC nebo adresu v režimu SDLC. Synchronní režim s vnější synchronizací nevyužívá obsahu registru WR6.

D7	D6	D5	D4	D3	D2	D1	D0

Řídicí registr WR7

Registr WR7 plní obdobnou funkci jako registr WR6. V režimu BISYNC definuje obsah druhého synchronizačního znaku a v režimu SDLC obsahuje příznakový byte nebo-li profil FLAG 01111110. Ostatní přenosové protokoly registr WR7 nevyužívají.

D7	D6	D5	D4	D3	D2	D1	D0

Následuje přehled 3 stavových registrů RR0 ÷ RR2. Číslo stavového registru, který je přenesen do Z80-CPU, definuje opět ukazatel v pozici D0 ÷ D2 řídicího registru WR0. Registr RR2 je obsažen pouze v kanálu B.

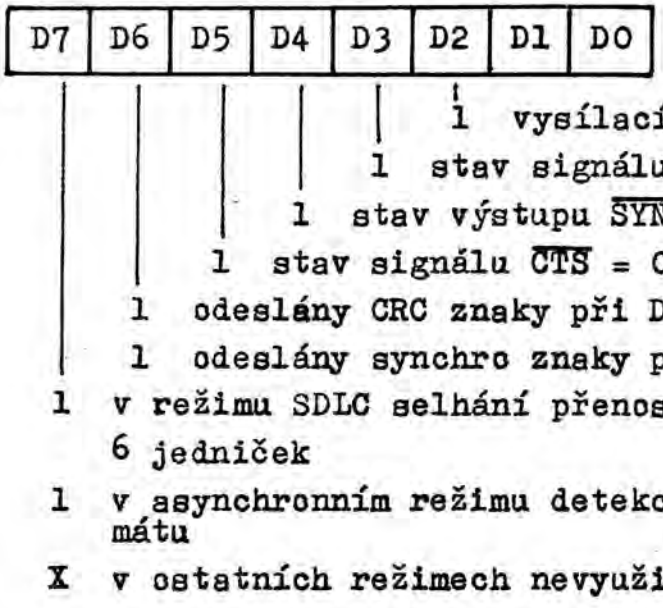
Stavový registr RR0

Stavový registr RR0 informuje mikroprocesor o stavu vysílače, přijímače a přerušení.

D7	D6	D5	D4	D3	D2	D1	D0

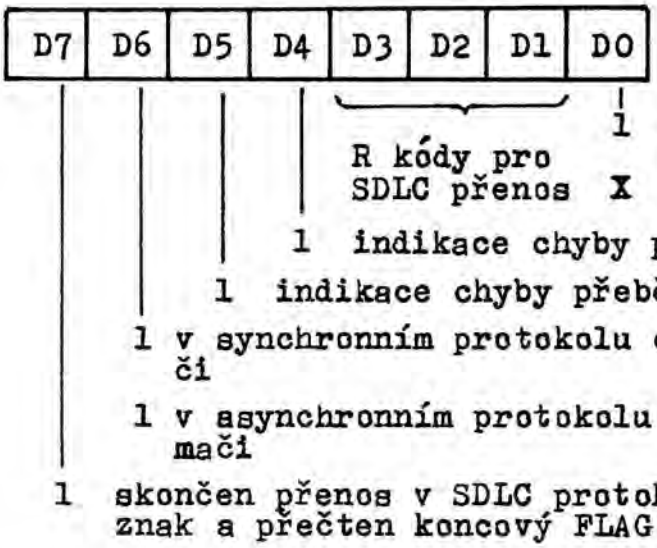
1 přijímač obsahuje nejméně 1 znak dat

1 vzniklo jakékoliv přerušení v Z80-SIO (kanál B tento bit nastavuje vždy na nulu - nevyužívá jej)



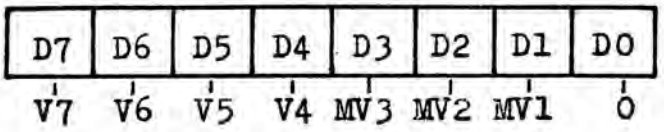
Stavový registr RR1

Registr RR1 dodává do mikroprocesoru stav speciálních podmínek přijímače.



Stavový registr RR2

Obsah registru RR2 je možné získat jen vyčtením z kanálu B. Stavový registr přenáší modifikovaný vektor přerušení pro oba kanály, byla-li modifikace vektoru povolena bitem D2 řídicího slova WR1. Za stavu bitu D2 = 1 slova WR1 se obsadí stavové slovo RR2 v pozici D1, D2 = 1 a D3 = 0, pakliže není požadováno přerušení.



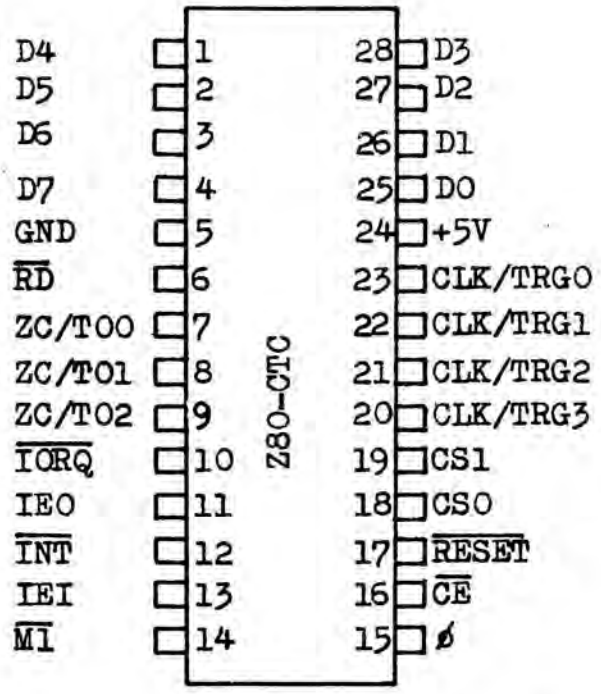
4.3 Z80-CTC

Obvod Z80-CTC (COUNTER/TIMER CONTROLLER) je čtyřkanálová jednotka, mající funkci čítače a časovače. Pro generování časových smyček, hodin reálného času využívá funkci čítače, kdežto časovač slouží k vytváření přenosových frekvencí např. pro obvod Z80-SIO.

Jednotlivé kanály jsou nezávislé, nastavitelné do jednoho ze dvou režimů - čítací režim nebo časovací režim. Přitom kanály lze na sebe vázat také vnějším propojením. Obvod může generovat přerušení při průchodu čítače k nule nulou v čítacím i časovacím režimu. Ošetření přerušení je totožné s ostatními podpůrnými obvody. Obsah vektoru přerušení odpovídá číslu kanálu, který přerušení vyvolal. Maximální prioritu má kanál 0 a tato priorita klesá ke kanálu 3.

Oba režimy umožňují číst stav čítače k nule a tím testovat okamžitý stav čítací a časovací funkce. Stav čítače k nule po dopočtu do nuly je automaticky obnovován z registru předvolby v obou režimech práce. Kanál 0 + 2 má vyveden výstup čítače k nule ZC/TO, kdežto kanál 3 pro nedostatek vývodů pouzdra tento vývod k dispozici nemá, ale používá ke stejnému účelu přerušovacího výstupu $\overline{INT}$. Čítač se spouští a dekrementuje postupně s náběžnou nebo padací hranou signálu CLK/TRG ve významu CLK. Časovač má možnost automatického spuštění od zanesené hodnoty předvolby nebo externího spuštění pomocí definované aktivní úrovně signálu CLK/TRG ve významu TRG.

Obvod je napájen z + 5 V, pouzdro je 28 vývodové DIL, veškeré vstupy a výstupy jsou plně slučitelné s TTL. Obr. 92 ukazuje zapojení vývodů pouzdra Z80-CTC. Adresování jednotlivých kanálů pouzdra provádí dvojice výběrových vodičů CS0 a CS1, které se připojují na adresní sběrnici mikroprocesoru.



Obr. 92 Zapojení vývodů pouzdra Z80-CTC

4.3.1 Popis vývodů pouzdra

Schematická značka obvodu je na obr. 93, dále je popsán význam jednotlivých vývodů.

D0 ÷ D7
datová sběrnice

třístavová vstup/výstupní sběrnice. Sběrnice se používá k přenosu veškerých datových a řídicích informací mezi Z80-CPU a Z80-CTC.

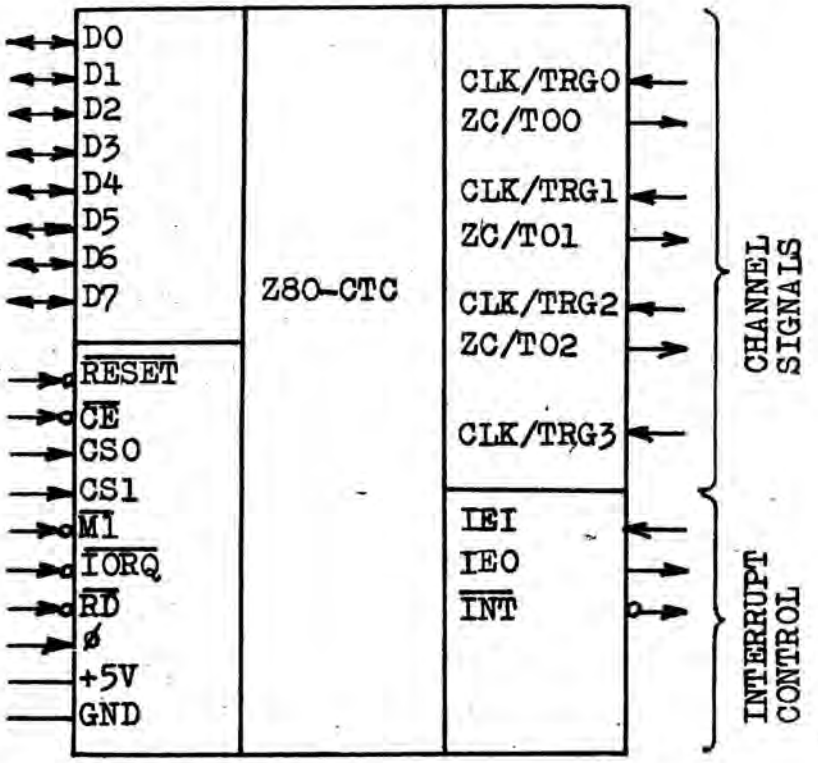
$\overline{CE}$
výběr obvodu

vstupní signál. Obvod je aktivován pouze při $\overline{CE} = 0$.

CS1, CS0
volba kanálu

vstupy adresují jeden ze čtyř nezávislých kanálů Z80-CTC při čtecích a zápisových operacích.

CS1	CS0	KANÁL
0	0	0
0	1	1
1	0	2
1	1	3



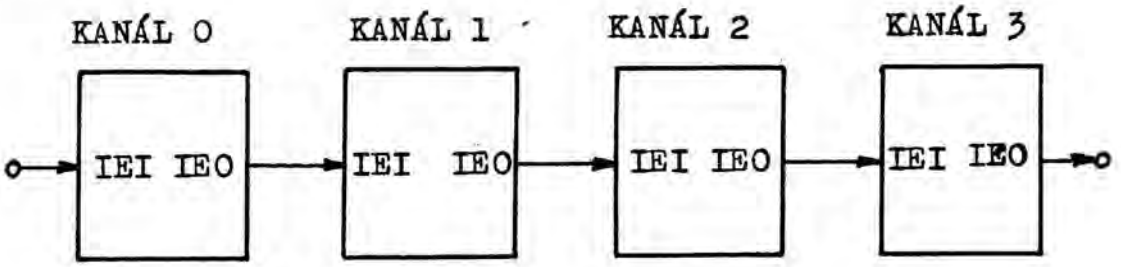
Obr. 93 Schematická značka Z80-CTC

ø systémové hodiny	synchronizační signál hodin se používá pro vnitřní synchronizaci.
$\overline{RD}$ čtení	vstupní řídicí signál, který bude vysvětlen v souvislosti se signály $\overline{IORQ}$ a $\overline{M1}$.
$\overline{M1}$ prvý strojový cyklus	vstupní řídicí signál $\overline{M1}$ informuje společně se signály $\overline{RD}$ a $\overline{IORQ}$ o sběrnicovém cyklu: - při $\overline{M1}$ = 0 a $\overline{RD}$ = 0 je čtena instrukce z paměti - při současné aktivitě $\overline{M1}$ = 0 a $\overline{IORQ}$ = 0 CPU potvrzuje přijetí požadavku na přerušení a obvod Z80-CTC vkládá na datovou sběrnici vektor přerušení.
$\overline{IORQ}$ žádost na V/V zařízení	vstupní řídicí signál $\overline{IORQ}$ se využívá společně se signály $\overline{M1}$, $\overline{CE}$ a $\overline{RD}$. Obvod Z80-CTC nemá vstup $\overline{WR}$, a proto zápisový cyklus je

lokalizován stavem signálů $\overline{CE}$ = 0, $\overline{IORQ}$ = 0 a $\overline{RD}$ = 1. Je-li $\overline{M1}$ = 0 a $\overline{IORQ}$ = 0, jde o přerušovací cyklus.

IEI
povolení generování přerušení
IEO
povolení generování přerušení

oba vodiče slouží pro vytváření sériových řetězců priorit. Vnitřní pevnou prioritní vazbu mezi kanálem 0 ÷ 3 uvádí obr. 94. Nejvyšší prioritu má kanál 0, nejnižší prioritu kanál 3.



Obr. 94 Vnitřní prioritní řetězec kanálů Z80-CTC

$\overline{INT}$
přerušení

výstupní signál na otevřeném kolektoru. $\overline{INT}$ je vyvolán při průchodu čítače k nule nulou.

$\overline{RESET}$
inicializace obvodu

signál $\overline{RESET}$ zastavuje čítání v kanálu 0 ÷ 3, nastavuje bit D7 na nulu tj. přerušení je blokováno a výstupní signály typu ZC/TO a $\overline{INT}$ přechází do neaktivního stavu. Datová sběrnice se nachází ve třetím stavu a výstup IEO nese úroveň přítomnou na vstupu IEI.

CLK/TRG0 ÷ CLK/TRG3
vnější hodiny/spouštění časovače

vstupní signál s programově nastavitelnou aktivní úrovní pomocí bitu D4 režimového řídícího slova. Čtyři vstupy CLK/TRG0 ÷ CLK/TRG3 odpovídají jednotlivým kanálům. V režimu čítače se každou aktivní hranou vstupního signálu dekrementuje stav čítače k nule.

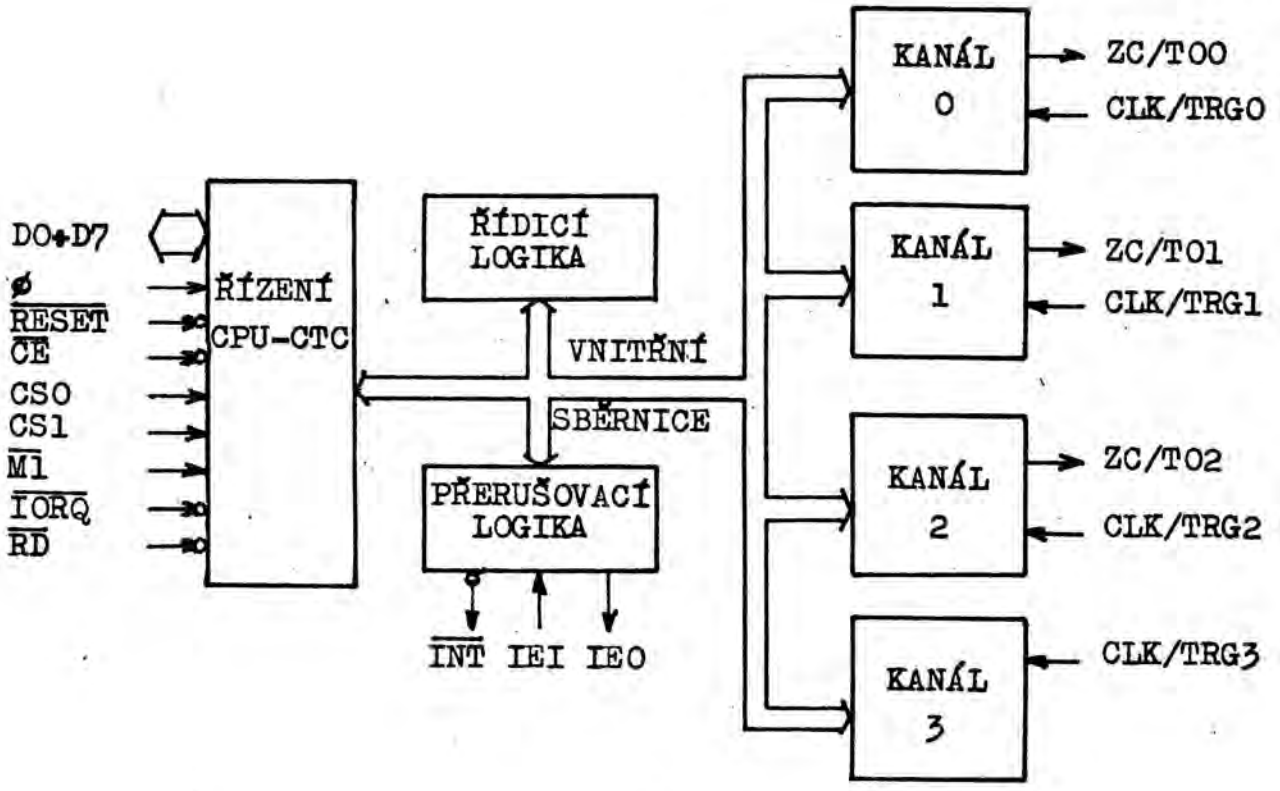
Aktivní hrana je specifikována bitem D4 režimového řídicího slova. V režimu časovače se aktivní hranou na vstupech CLK/TRG0 ÷ CLK/TRG3 spouští časování.

ZC/T00 ÷ ZC/T02 načteno do nuly/ konec časového intervalu

výstupní signály indikují impuls při načítání do nuly. Čtvrtý kanál nemá výstup typu ZC/TO vyveden vzhledem k omezenému počtu vývodů pouzdra.

4.3.2 Architektura obvodu

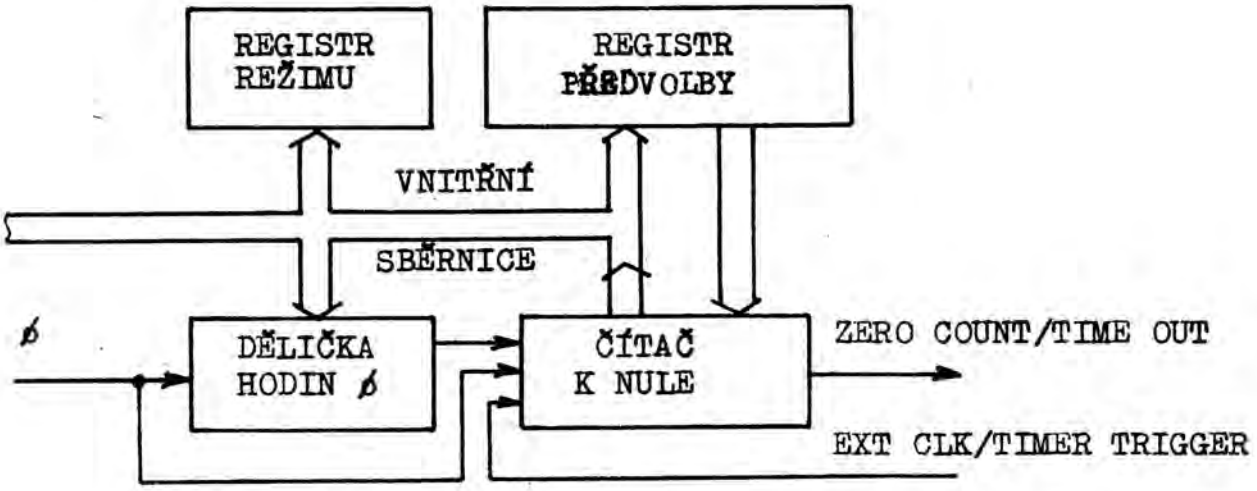
Blokové schéma Z80-CTC je na obr. 95 .



Obr. 95 Blokové schéma Z80-CTC

Bloky obvodu jsou propojeny vnitřní sběrnicí, která zajišťuje zanesení řídicích slov do jednotlivých kanálů. Styk s mikroprocesorem pomocí datové a řídicí sběrnice provádí blok řízení CPU-CTC. Interní řízení je zabezpečeno řídicí logikou, odbavení přerušení od jednotlivých kanálů řídí přerušovací logika. Bloky kanálů 0 + 3 pracují na stejných principech, priorita klesá od kanálu 0 ke kanálu 3. Kanál 3 nemá výstupní indikaci nulového stavu čítače k nule.

Struktura každého kanálu se rozpadá na bloky, které jsou znázorněny na obr. 96 .



Obr. 96 Vnitřní struktura kanálu Z80-CTC

Režim práce obvodu čítač/časovač a další podmínky práce kanálu je třeba po provedeném RESETu nadefinovat osmibitovým řídícím slovem, které plní registr režimu. Zároveň se programově dodává společný vektor přerušení, který je ale při fyzickém odbavení přerušení doplněn automaticky na pozici D1, D2 číslem kanálu, který žádá o přerušení. Objasnění funkce bloků kanálu je rozebráno v kapitole 4.3.3 ve vazbě na režim práce čítače a časovače.

4.3.3 Režimy práce

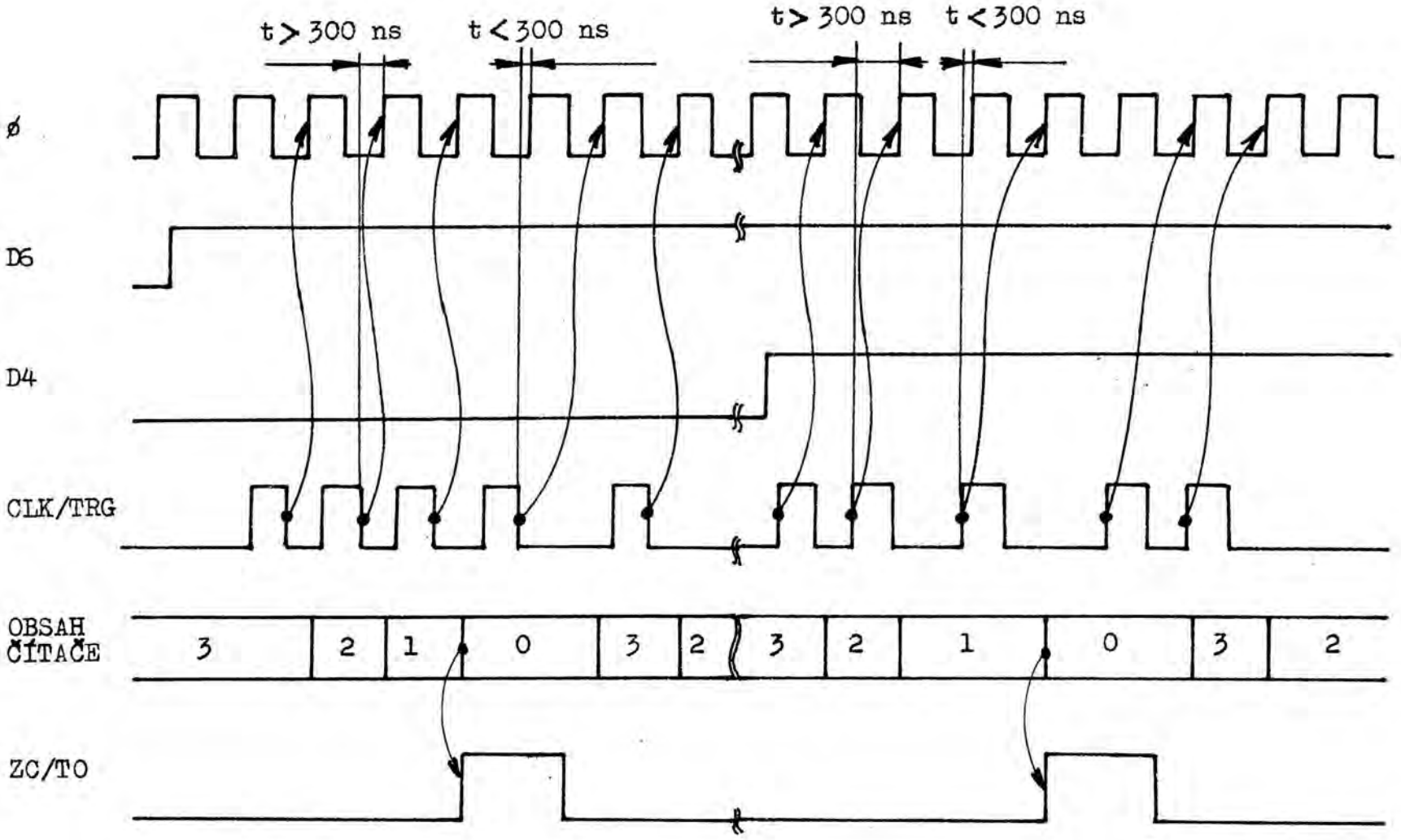
Režim čítače

Režim čítače, nastavený řídicím slovem režimu bit D6 = 1, pracuje s registrem předvolby a čítačem k nule. Výstupní signál ZC/TO je využit jako ZERO COUNT a vstupní signál CLK/TRG jako hodinový vstup CLOCK. Počáteční hodnotu čítače k nule definuje programovaný obsah registru předvolby. Tato hodnota se zanese do čítače k nule, který je dekrementován vždy s náběžnou hranou hodin ϕ následující po splnění spouštěcí podmínky dané náběžnou či padací hranou signálu CLK/TRG. Náběžnou či padací hranu specifikuje bit D4 řídicího slova režimu. Minimální odstup spouštěcí hrany a náběžné hrany hodin ϕ je 300 ns, je-li nižší, je dekrementace až s další náběžnou hranou hodin ϕ. U kanálu 0 + 2 při průchodu čítače nulou se generuje jedničkový impuls na výstupu ZC/TO. Zároveň se obnovuje obsah čítače k nule z registru předvolby, a proto tuto cyklickou činnost lze porovnat např. s MODE2 obvodu I8253. Čítání periodicky pokračuje podle již výše uvedených zásad s hodinami CLK/TRG a ϕ. Časování režimu čítače je na obr. 97.

Režim časovače

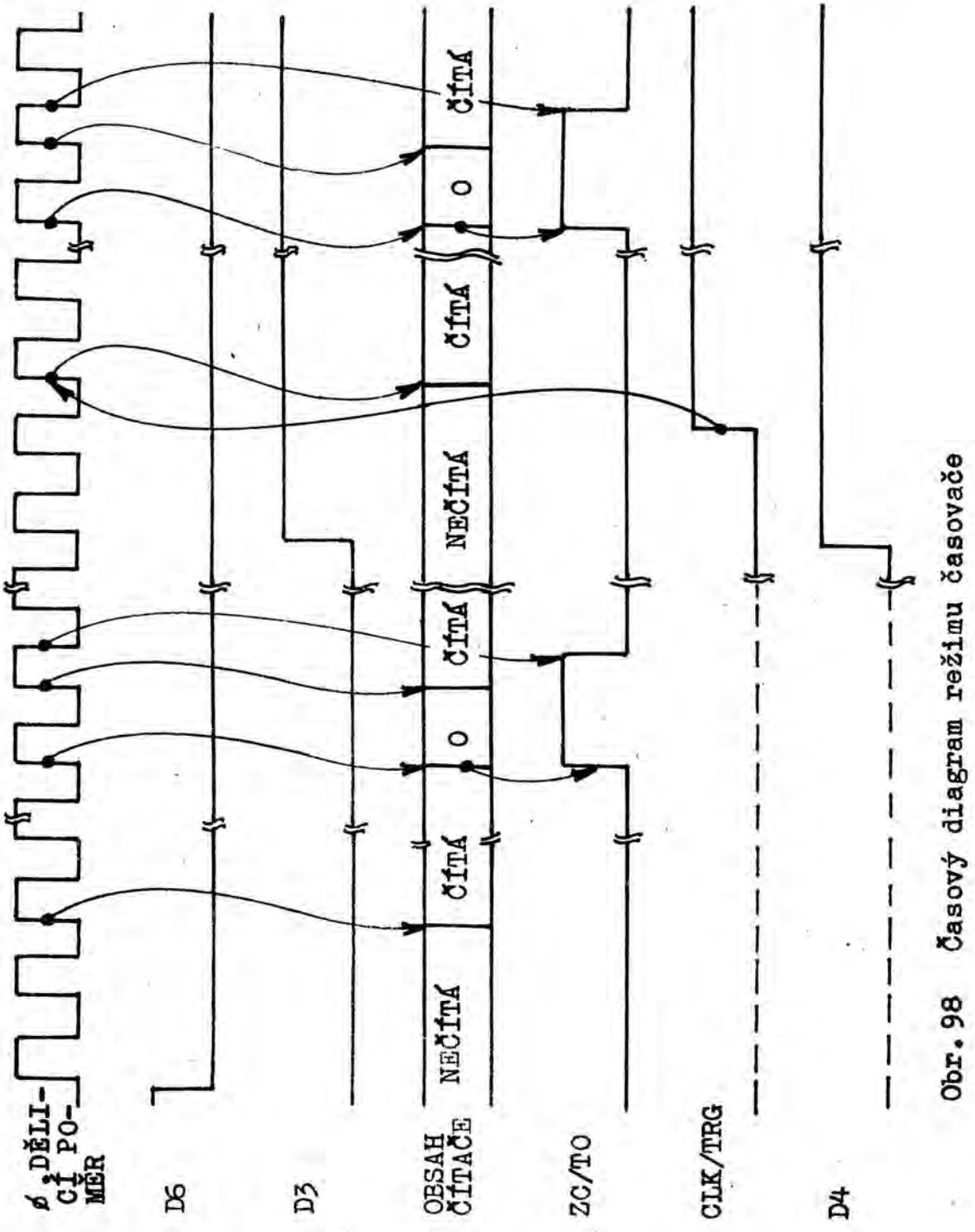
Režim časovače se nastavuje programovým slovem režimu s obsahem bitu D6 = 0. V režimu časovače je kromě již rozbíraných bloků uplatněna dělička hodin ϕ, na úrovni které dochází k dělení vstupní frekvence hodin ϕ v poměru 1 : 16 nebo 1 : 256 dle naprogramované hodnoty bitu D5 v režimovém řídicím slově. Takto podělená frekvence působí na čítač k nule nastavený z registru předvolby. Stejně jako v režimu čítače je obsah registru předvolby použit k znovunastavení čítače k nule po dosažení nulové hodnoty čítače k nule. Aktivita výstupu ZC/TO ve významu TIME OUT je jedničková po dobu nulového stavu čítače k nule. Opakovací perioda aktivity signálu ZC/TO je dána součinem periody hodin ϕ, dělícího poměru 1: 16 nebo 1 : 256 a hodnotou předvolby časovače.

Pomocí bitu D3 řídicího slova lze volit automatické nebo externí spouštění časovače. Spouštění automatické nastává ve



Obr. 97 Časový diagram režimu čítače

strojovém cyklu následujícím za strojovým cyklem zápisu předvolby. Extermí spouštění se provádí signálem CLK/TRG ve funkci TRIGGER ve vazbě na aktivitu signálu CLK/TRG nadefinovanou bitem D4. Fyzické spouštění nastává až náběžnou hranou hodin ø následující po aktivní podmínce CLK/TRG. Obr. 98 znázorňuje časový diagram režimu časovače.



Obr. 98 Časový diagram režimu časovače

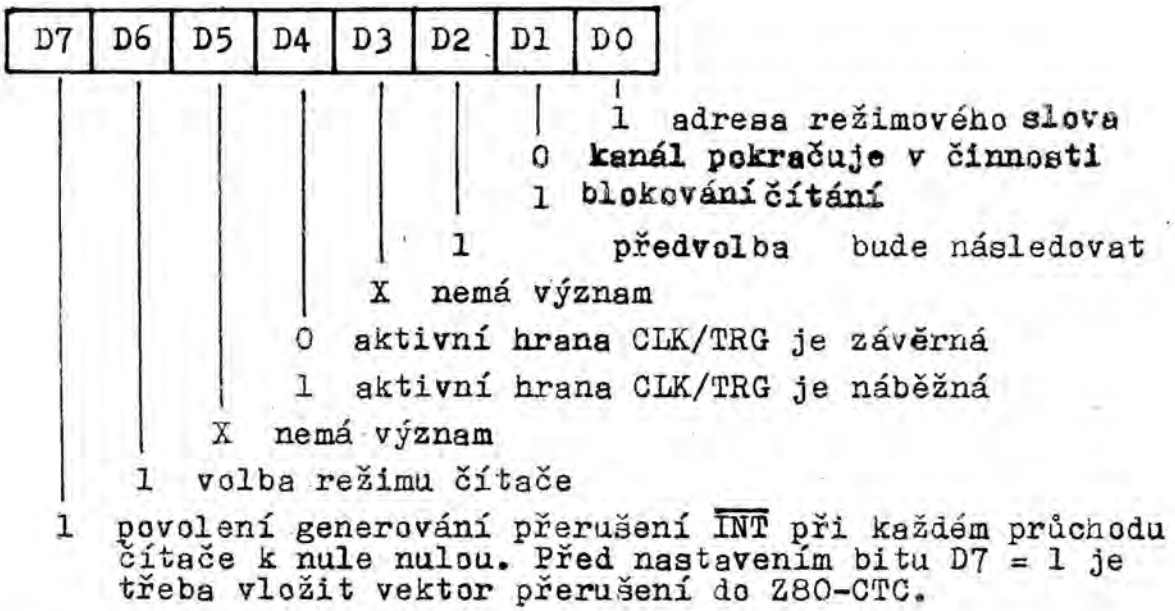
4.3.4 Programování Z80-CTC

Vkládání programových slov do jednotlivých kanálů je specifikováno adresou kanálu na vstupních vodičích CS0, CS1. Z toho vyplývá, že obvod má např. 4 režimová řídicí slova, 4 slova pro nastavení registrů předvolby, ale výjimku tvoří společný registr vektoru přerušení, který se zapisuje pod adresou kanálu nula tj. CS0= 0 a CS1= 0.

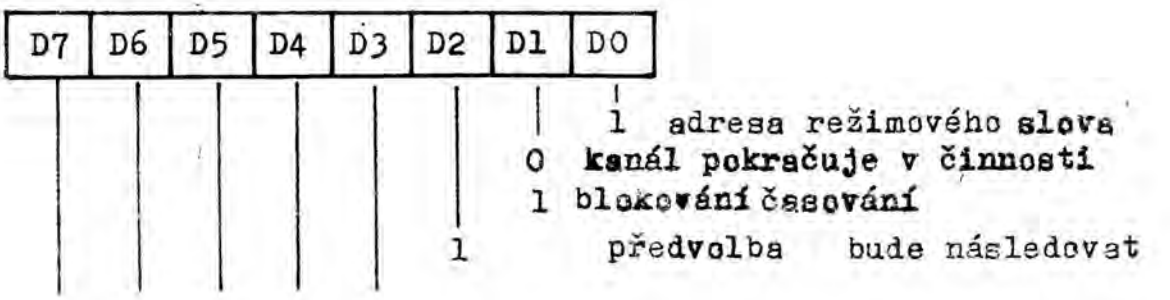
Režimové řídicí slovo

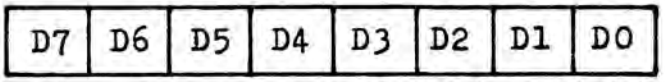
Toto slovo nastavuje funkci obvodu čítač/časovač a řadu dalších podmínek. Význam některých bitů režimového slova je odlišný pro režim čítače a časovače.

Režimové řídicí slovo čítače



Režimové řídicí slovo časovače





0 automatické spouštění náb. hranou po T2 po zavedení předvolby
1 externí spouštění signálem CLK/TRG

0 spouštění závěrnou hranou CLK/TRG
1 spouštění náběžnou hranou CLK/TRG

0 dělící poměr 1 : 16
1 dělící poměr 1 : 256

0 volba režimu časovače

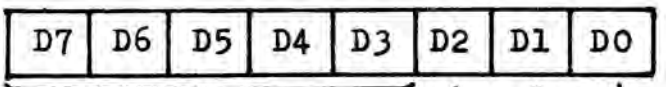
1 povolení generování přerušení od každého průchodu čítače k nule nulou. Opět je třeba zapsat vektor přerušení do Z80-CTC před nastavením bitu D7 = 1.

Předvolba čítače/časovače

Osmibitová hodnota **předvolby** čítače/časovače se nahrává do registru **předvolby** po vložení řídicího režimového slova s bitem D2 = 1, který definuje, že za režimovým slovem následuje předvolba. Adresa kanálu CS0, CS1 musí být totožná pro obě programová slova.

Vektor přerušení

Zápis společného přerušovacího vektoru se provádí do kanálu 0, tj. při adresaci CS0= 0, CS1= 0. Slovo specifikující vektor využívá pro adresaci pozici D0 = 0. Vektor je zapsán do pozic D3 + D7 a zbylé pozice D1 a D2, na kterých při programování nezáleží, jsou doplněny číslem kanálu žádajícího obsluhu přerušení.



0 sudá hodnota vektoru

X	X	modifikace vektoru přerušení obvodem Z80-CTC
0	0	kanál 0
0	1	kanál 1
1	0	kanál 2
1	1	kanál 3

4.4 Z80-DMA

Z80-DMA (DIRECT MEMORY ACCESS CONTROLLER) je programově řízený jednokanálový dvouadresový řadič, který generuje adresy, časové sekvence a řídicí signály pro DMA přenosy dat. Rychlý přenos dat se uskutečňuje obecně mezi dvěma oblastmi mikropočítačového systému, kterými mohou být paměti nebo vstup/výstupní zařízení. Mimo fyzického přenosu dat řadič DMA umožňuje vyhledat informaci v bloku dat anebo po vyhledání informace zahájit přenos dat. Protože obvod Z80-DMA generuje veškeré signály potřebné pro zabezpečení přenosu, je schopen převzít řízení na sběrnici po odpojení mikroprocesoru od sběrnice.

Přenos může probíhat po bytech,tj. vrácení řízení mikroprocesoru po každém bytovém přenosu,nebo spojitě,tj. blokově, kdy se mikroprocesor ujímá řízení až po dokončení přenosu bloku dat nebo po vzniku neaktivního signálu RDY.

Dva adresní šestnáctibitové registry generují adresu čteného a zapisovaného místa , přenosová délka je lokalizovaná v šestnáctibitovém registru. Datový DMA přenos je veden přes jednobytový vyrovnávací registr obvodu Z80-DMA.

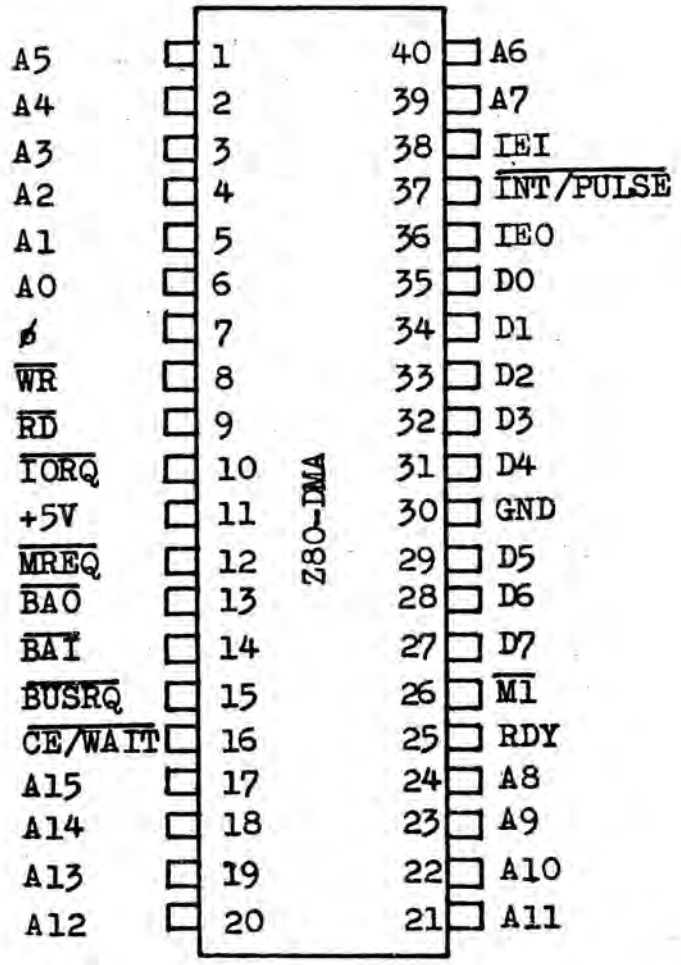
Obdobný prvek firmy Intel řadič DMA přenosu I 8257 má sice 4 kanály, ale jen jeden šestnáctibitový adresní registr a délka přenosu je omezena čtrnáctibitovým registrem na 16 kB.

Zapojení vývodů pouzdra je na obr. 99. Obvod je zapouzdřen v 40 vývodovém pouzdře DIL s napájením + 5 V, přenosová rychlost do 1,25 MB/s v SEARCH nebo TRANSFER MODE.

4.4.1 Popis vývodů pouzdra

Význam vývodů pouzdra je dán tím, zda řadič DMA pracuje v systému jako řízený obvod (SLAVE) nebo jako řídicí obvod (MASTER). Ve stavu MASTER řadič zajišťuje jednotlivé režimy DMA přenosů. Schematická značka Z80-DMA je na obr. 100.

D0 ÷ D7 datová sběrnice	třístavová obousměrná sběrnice. Za stavu SLAVE slouží pro přenos sady řídicích slov nebo pro čtení vnitřních stavů řadiče DMA.

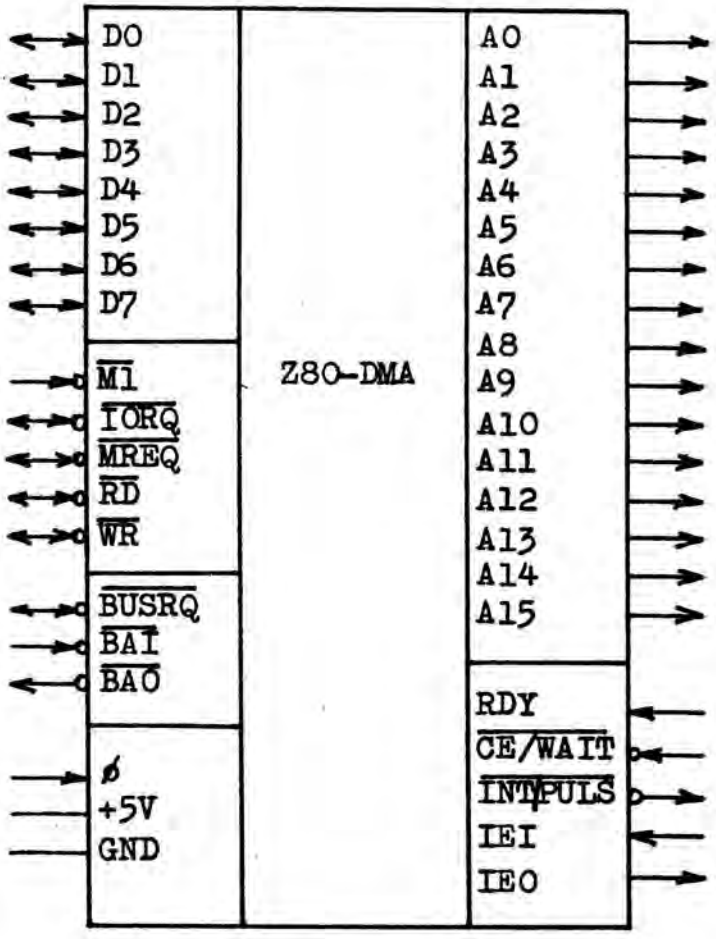


Obr. 99 Zapojení vývodů pouzdra Z80-DMA

Za stavu MASTER je přes ní veden přenos do vyrovnávací datové paměti a zpět k příjemci datové informace.

A0 ÷ A15
adresová sběrnice

adresní třístavová sběrnice je pouze výstupní a slouží pro multiplexní adresaci dvou oblastí při DMA přenosu. Adresa při DMA přenosu může být po každém přeneseném bytu inkrementována, dekrementována nebo setrvává fixní dle naprogramování. Pokud řízení sběrnice přísluší Z80-CPU, je adresní sběrnice řadiče DMA ve třetím stavu.



Obr. 100 Schematická značka Z80-DMA

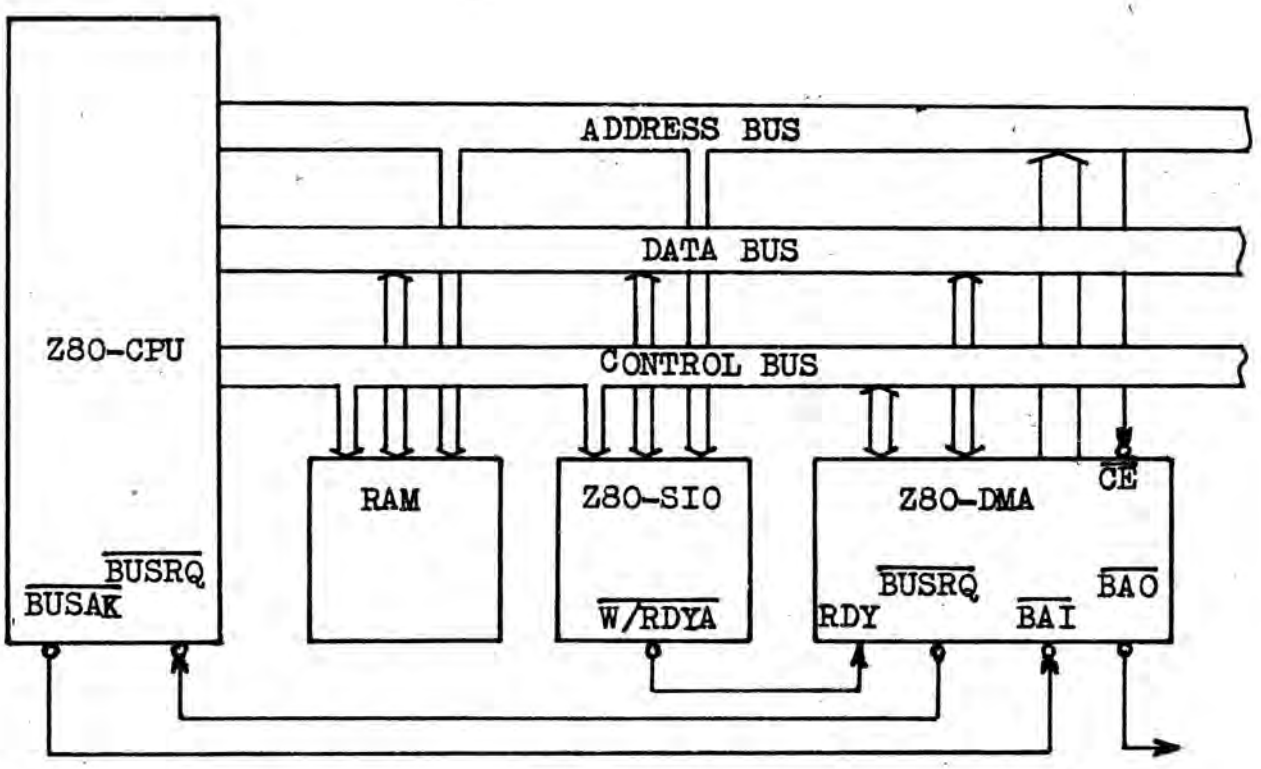
$\overline{M1}$ první strojový cyklus	výhradně vstupní řídicí signál, který při své aktivitě za stavu $\overline{IORQ}$ = 0 identifikuje cyklus přerušení.
$\overline{IORQ}$ žádost na V/V zařízení	vstupní nebo výstupní řídicí signál. Při zápisu řídicích slov do Z80-DMA a při čtení vnitřních stavů obvodu je $\overline{IORQ}$ vstupem. Režim DMA využívá signálu jako výstupního pro řízení periferních jednotek.
$\overline{MREQ}$ žádost o paměť	vstup/výstupní signál využívaný pro řízení bloku paměti při DMA přenosu.

$\overline{WR}$ zápis	vstupní nebo výstupní řídicí signál. Jako vstup působí při vkládání řídicích slov do řadiče. Výstupem je v rámci zápisových operací při DMA přenosu.
$\overline{RD}$ čtení	vstupní nebo výstupní řídicí signál. Jako vstup je využit v etapě čtení stavových slov z DMA obvodu, jako výstup v době čtení dat při DMA přenosu.
ϕ systémové hodiny	od tohoto vstupu je odvozena vnitřní synchronizace signálů.
$\overline{CE/WAIT}$ výběr obvodu/zpomalení DMA řadiče	vstupní signál. Je-li Z80-DMA řízeným obvodem, signál $\overline{CE}$ je typickým výběrovým vodičem a působí spolu se signály $\overline{WR}$, $\overline{RD}$, $\overline{IORQ}$. Programově lze přiřadit při DMA přenosu vodiči význam $\overline{WAIT}$, čímž se přizpůsobuje komunikace DMA řadiče s pomalejšími prvky. K vložení jednoho taktu do strojového cyklu čtení nebo zápisu dojde tehdy, je-li při testování úrovně na vstupu $\overline{WAIT}$ nula. Vzorkování se provádí při sestupné hraně taktu T2. Strojový cyklus je možné upravit ještě programově. Standardní délka strojových cyklů odpovídá délce cyklů mikroprocesoru. Na tuto délku se řadič DMA nastaví vždy po připojení napájení nebo po programovém znulování. Za pomoci programového slova je délka strojového cyklu nastavitelná v rozmezí 2 ÷ 4 taktů.
$\overline{BUSRQ}$ požadavek na převzetí sběrnice	vstupní nebo výstupní signál. $\overline{BUSRQ}$ je výstupem při požadavku o systémovou sběrnici. Výstup $\overline{BUSRQ}$ se připojuje na stejnojmenný vstup mikroprocesoru. $\overline{BUSRQ}$ je vstupním signálem při sériovém řazení řadičů DMA. Zde je příznakem, zda ostatní řadiče již ukončily přenos. Sériový prioritní řetězec DMA obvodů se od sériového přerušovacího řetězce liší v tom,

	že DMA přenos řadiče s nižší prioritou nelze přerušit řadičem s vyšší prioritou. Řadič s vyšší prioritou se ujme sběrnice až po ukončení DMA přenosu řadiče s nižší prioritou.
$\overline{BAI}$ potvrzení požadavku převzetí sběrnice	vstupní signál $\overline{BAI}$ se připojuje na mikroprocesorový výstup $\overline{BUSAK}$, a tím je potvrzen stav odpojenosti Z80-CPU od sběrnic. Řadič DMA se ujímá řídící činnosti. $\overline{BAI}$ je možné využít také pro vytváření prioritního řetězce DMA obvodů ve spolupráci s výstupním signálem $\overline{BAO}$.
$\overline{BAO}$ potvrzení požadavku na převzetí sběrnice DMA obvodem	výstupní signál indikující převzetí sběrnice DMA obvodem. $\overline{BAO}$ ve spolupráci se vstupním signálem $\overline{BAI}$ vytváří prioritní řetězce DMA řadičů.
RDY připravenost spolupracující periferie	vstup připravenosti spolupracující jednotky k DMA přenosu. Aktivita signálu RDY může ovlivňovat délku trvání $\overline{BUSRQ}$ na základě naprogramovaného typu přenosu. Aktivní úroveň RDY je programována řídicím slovem R5 a to bitem D3.
IEI povolení generování přerušení	vstup povolující generovat přerušení. Je vytvářen předešlým obvodem kaskády.
IEO povolení generování přerušení	výstup, který povoluje přerušení následujícího obvodu prioritního řetězce.
$\overline{INT/PULSE}$ přerušení/puls	výstup na otevřeném kolektoru. V režimu SLAVE žádá DMA obvod o přerušení tímto signálem ve významu $\overline{INT}$. V režimu MASTER se na výstupu ve významu $\overline{PULSE}$ objeví impuls na základě shody mezi registrem pulsu a počtem přenesených bytů. Tyto pulsy mají informativní význam o počtu přenesených bytů DMA kanálu.

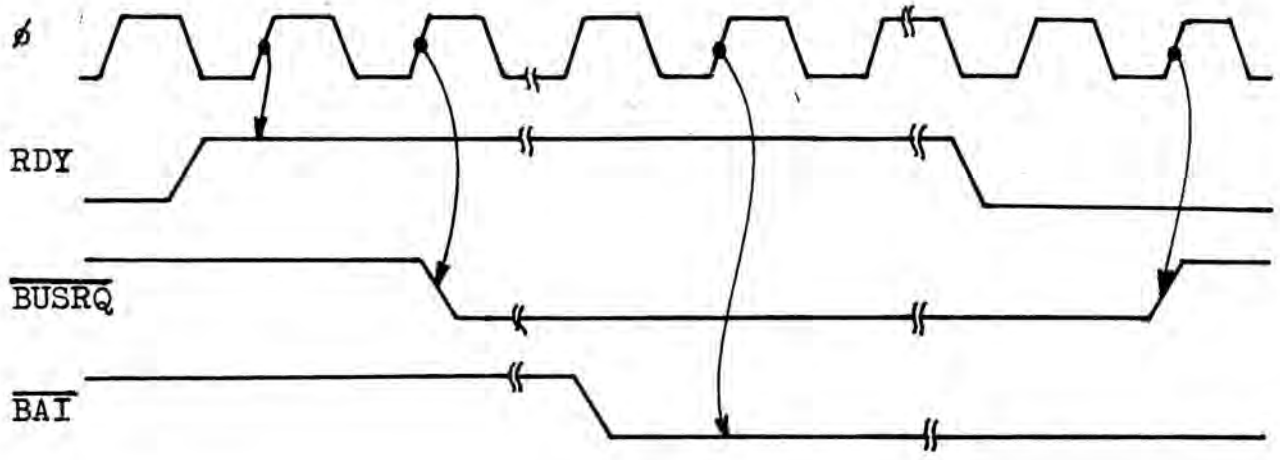
4.4.2 Časové rozbory řízení DMA kanálu

Obr.101 zachycuje základní včlenění řadiče DMA do mikroprocesorového systému Z80. Význam signálů RDY, $\overline{BUSRQ}$, $\overline{BAI}$ a $\overline{BAO}$ byl jednak objasněn v souvislosti s popisem vývodů pouzdra a jednak bude zobrazen na časových diagramech, které následují.



Obr. 101 Blokové schéma řízení DMA přenosu v systému Z-80

Cyklus žádosti o DMA přenos je na časovém diagramu obr. 102. Vstupní signál RDY je řadičem DMA vzorkován vždy s náběžnou hranou hodin ϕ. Jestliže RDY = 1, je od následující náběžné hrany hodin ϕ generován výstupní signál $\overline{BUSRQ}$. $\overline{BUSRQ}$ je přiveden na

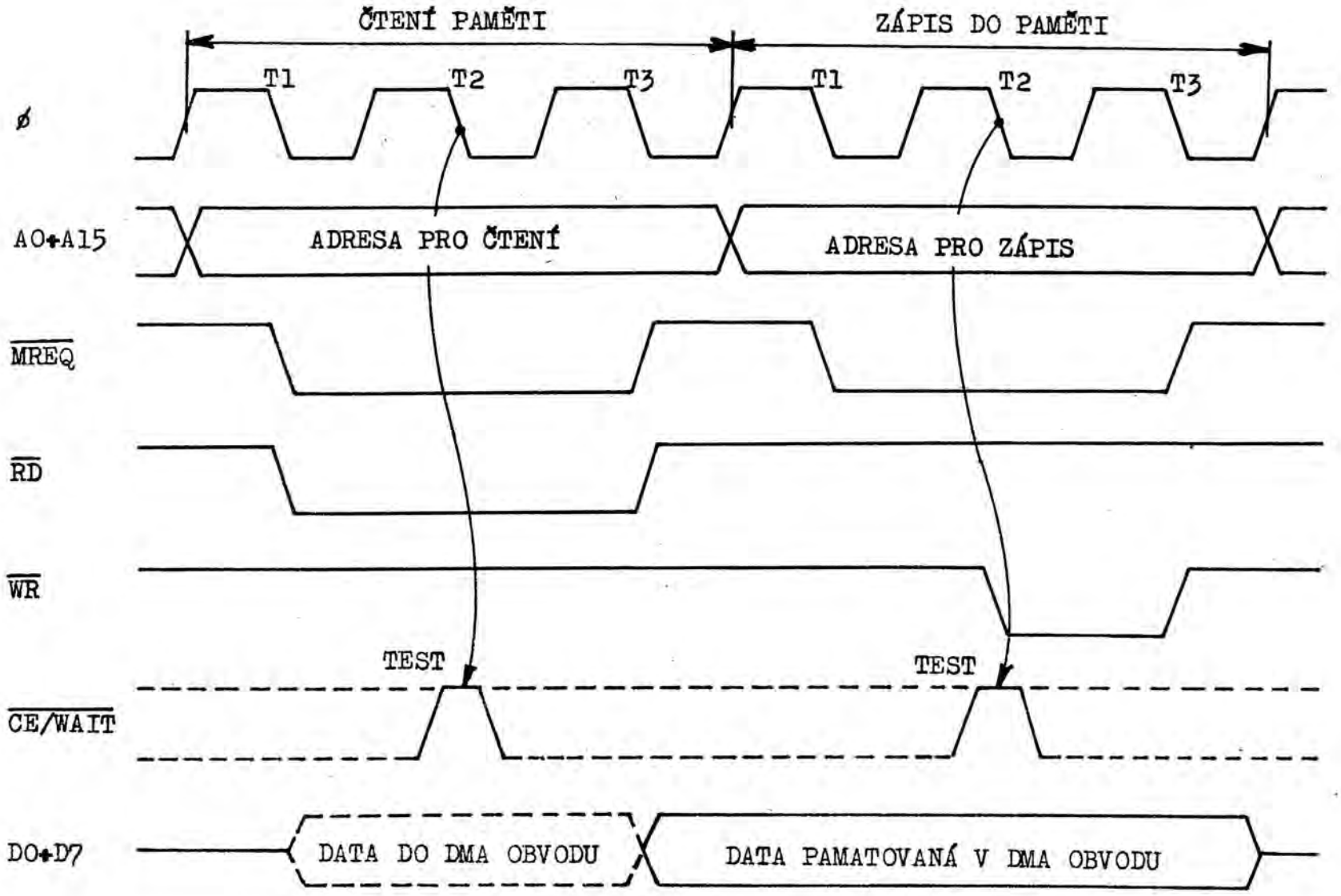


Obr.102 Časový diagram žádosti o DMA cyklus

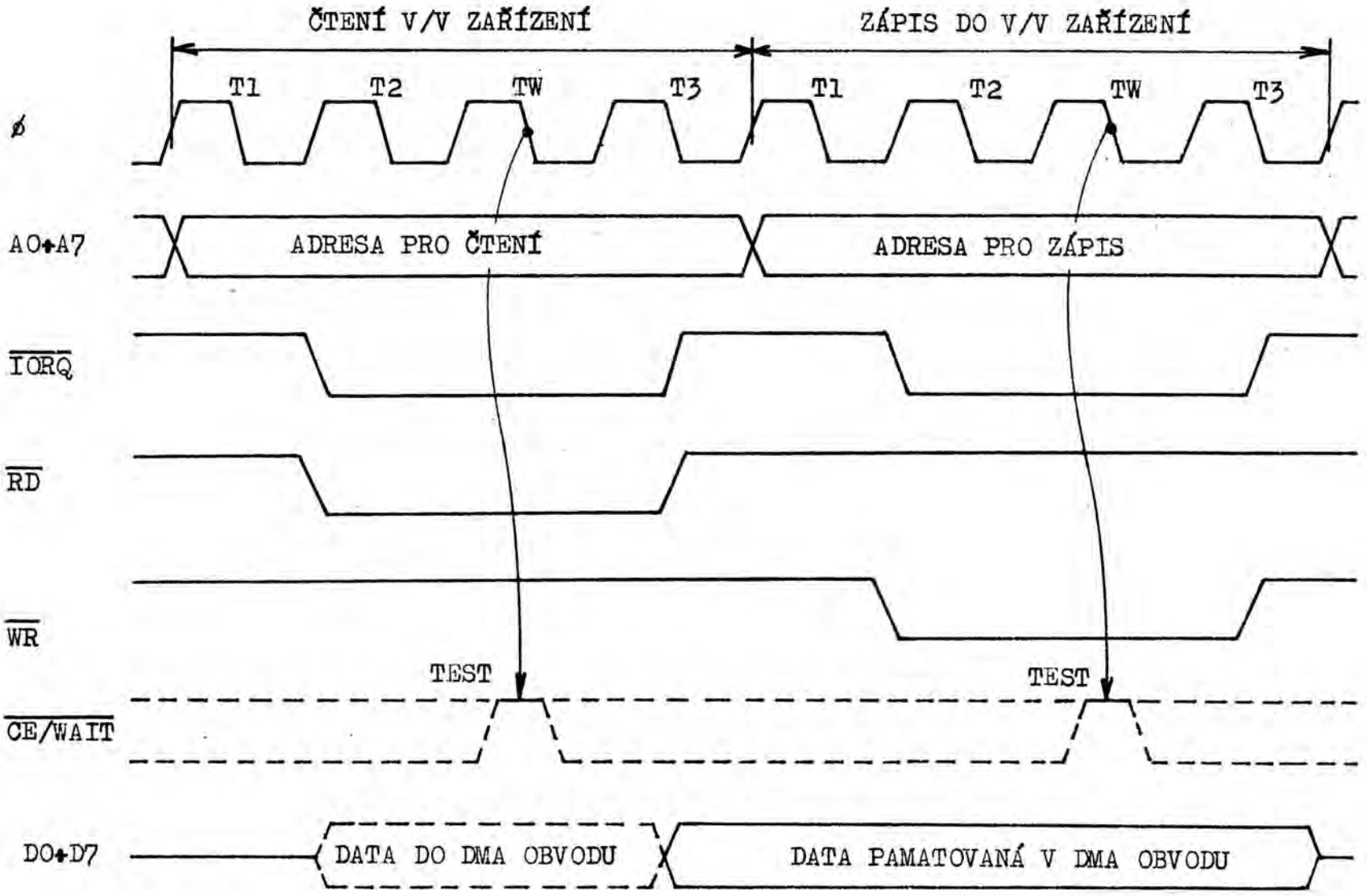
stejnojmenný vstup mikroprocesoru. Z80-CPU přechází do třetího stavu na sběrnici a generuje výstupní signál $\overline{BUSAK}$, který je připojen na $\overline{BAI}$ Z80-DMA. $\overline{BAI}$ je testován náběžnou hranou ø. Po zjištění $\overline{BAI}$ = 0 následují 2 periody hodin, po kterých se zahajuje vlastní DMA přenos. Ukončení DMA přenosu od signálu RDY probíhá se zpožděním 1 periody od zjištěné nulové úrovně RDY.

Následuje rozbor časové posloupnosti při DMA přenosu vedeném např. mezi dvěma zónami paměti (viz časový diagram na obr.103). Adresní sběrnice A0 ÷ A15 nese v cyklu čtení paměti adresu zdrojového bytu. Vyčtená informace je zanesena do jednobytového vyrovnávacího registru DMA řadiče, kde je pamatována a s následujícím zápisovým DMA cyklem je informační profil zapsán na lokalizované cílové místo paměti. Lokalizace je provedena aktivními signály A0 ÷ A15, $\overline{MREQ}$, $\overline{WR}$.

Obdobná je časová specifikace DMA přenosů mezi dvěma periferiemi nebo mezi periferií a pamětí či pamětí a periferií. Příklad DMA kanálu,vytvořeného mezi dvěma periferiemi,je na časovém diagramu obr.104.



Obr.103 Časový diagram DMA přenosu mezi dvěma zónami paměti



Obr.104 Časový diagram DMA přenosu mezi dvěma periferiemi

4.4.3 Architektura obvodu

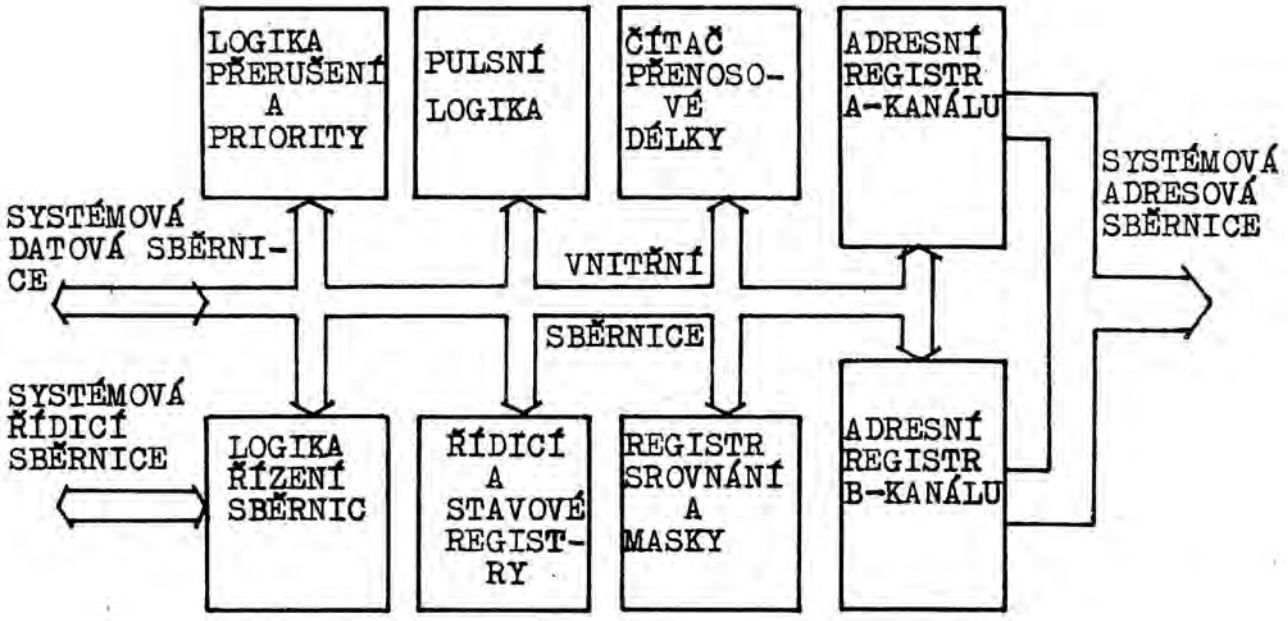
Vnitřní strukturu obvodu lze chápat jako několik bloků vzájemně spojených vnitřní sběrnicí. Bloky jsou plněny programovými slovy v etapě, kdy obvod Z80-DMA je řízeným prvkem systému. Po odpojení mikroprocesoru slouží informace z vnitřních registrů DMA řadiče pro řízení rychlých přenosů mezi lokalizovanými místy. Základ tohoto řízení tvoří adresní registry A - kanálu a B - kanálu, kde jsou 2 šestnáctibitové adresy zdrojového a cílového místa. Zdrojovým místem může být paměť či periferie, přičemž cílové místo má stejnou specifikaci. Proto DMA přenos je veden mezi pamětí a pamětí, periferií a pamětí, pamětí a periferií nebo periferií a periferií. Adresu lze vyvíjet vzestupně nebo sestupně dle naprogramování. Délka přenosu je uložena v čítači přenosové délky a je postupně s každým přeneseným bytem dekrementována. Přenos bloku řadič značkuje pomocí bloku pulsní logiky. Základní značkování je po každých 256 přenesených informacích, ale je možné jej změnit obsahem jednobytového registru generátoru pulsů na interval 1 ÷ 255. Synchronizaci generování řídicích signálů provádí logika řízení sběrnic. Pro obsluhu přerušení např. po vyčerpání přenosové délky nebo při srovnání profilu DMA přenosu s obsahem registru klíčového slova je v řadiči logika přerušení včetně registru vektoru přerušení. V režimu hledání klíčového slova je použito registru srovnání a masky. Při zjištění shody je nastaven do nuly bit D4 stavového registru SO.

Práce řadiče musí být definována sekvencí řídicích slov, které se posílají do obvodu pomocí výstupní repeatové instrukce OTIR. Stav o DMA přenosu lze získat čtecí operací. Suma těchto informací je zapisována nebo čerpána z bloku řídicí a stavové registry. Blokové schéma řadiče Z80-DMA je na obr. 105.

4.4.4 Režimy práce

Řadič DMA může pracovat v 6 základních režimech při vedení DMA přenosu, tj. v době, kdy převzal řízení nad sběrnicemi:

1. Přenos paměť - paměť - bytový DMA přenos mezi lokalizovaným zdrojovým a cílovým místem paměti do maximální adresní kapacity 64 kB.



Obr.105 Blokové schéma řadiče Z80-DMA

2. Přenos V/V periferie - V/V periferie - stejný způsob vedení DMA styku jako při přenosu mezi dvěma paměťovými zónami s tím rozdílem, že adresovaná místa jsou V/V periferie.

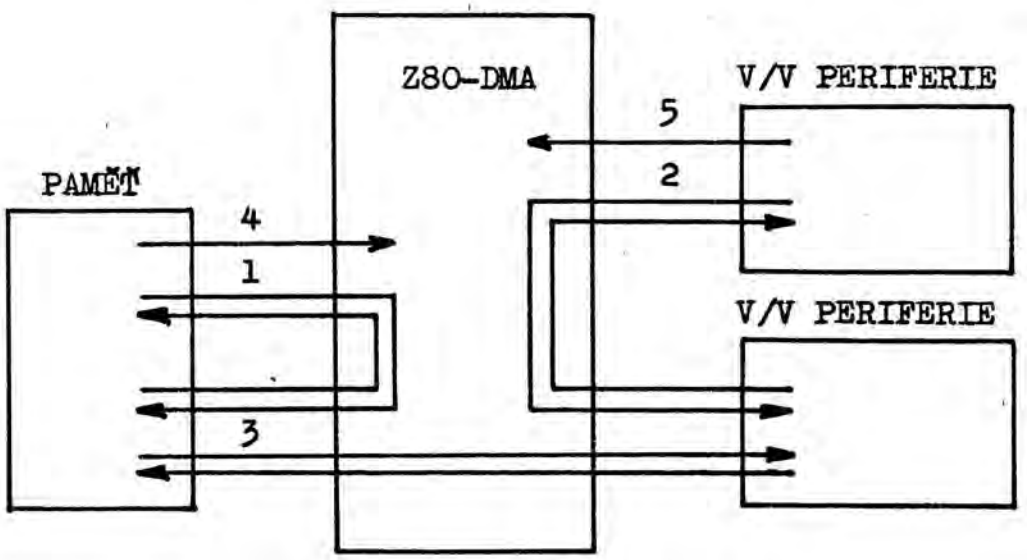
3. Přenos paměť - V/V periferie - kombinace výše uvedených přenosů, tj. přenos mezi paměťovým a periferním místem. Tento způsob přenosu zajišťoval řadič DMA firmy Intel I 8257.

4. Vyhledávání informace v paměti - data čtená z paměti jsou porovnávána s registrem srovnání a s registrem masky v řadiči DMA. Při shodě se nastaví do nuly bit D4 stavového registru SO, případně vznikne přerušení. Registr masky vybírá jen podstatné bity pro srovnání. V režimu vyhledávání je možné docílit rychlosti až 1,25 Mbytů/s.

5. Vyhledávání informace v V/V periferii - rozdíl proti vyhledávání v paměti je pouze v lokalizaci oblasti, kterou je nyní V/V periferie.

6. Vyhledávání informace s přenosem - po vyhledání bytu v registru srovnání, maskovaném registrem masky, se zahájí přenosová operace. Podmínky pro zakončení přenosu jsou programovatelné.

Režimy práce řadiče DMA jsou na obr. 106.



1 Přenos paměť-paměť (TRANSFER MEMORY TO MEMORY)
2 Přenos V/V periferie - V/V periferie (TRANSFER I/O TO I/O)
3 Přenos paměť - V/V periferie (TRANSFER MEMORY TO I/O)
4 Vyhledání informace v paměti (SEARCH MEMORY)
5 Vyhledání informace v V/V periferii (SEARCH I/O)

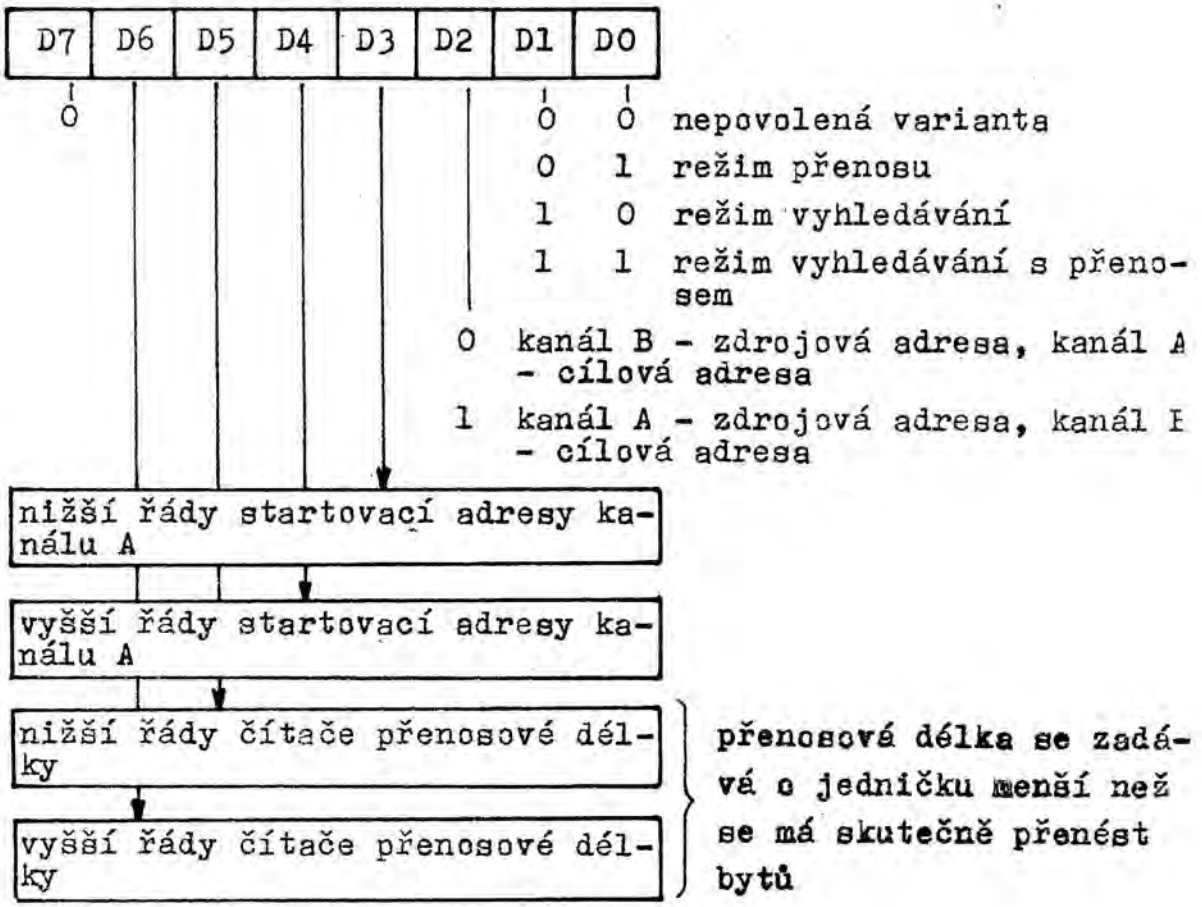
Obr. 106 Základní režimy práce Z80-DMA

4.4.5 Programování Z80-DMA

Činnost řadiče je specifikována 7 základními řídicími slovy R0 + R6, která v sobě obsahují příznakové bity podmiňující příchod dalších řídicích slov, přičemž tato slova zajišťují detailnější definování způsobu práce DMA řadiče. Adresování základních slov je pevným obsazením některých bitů těchto řídicích slov, a to bitů D0, D1, D7. Příznakové bity se ohledávají zprava doleva.

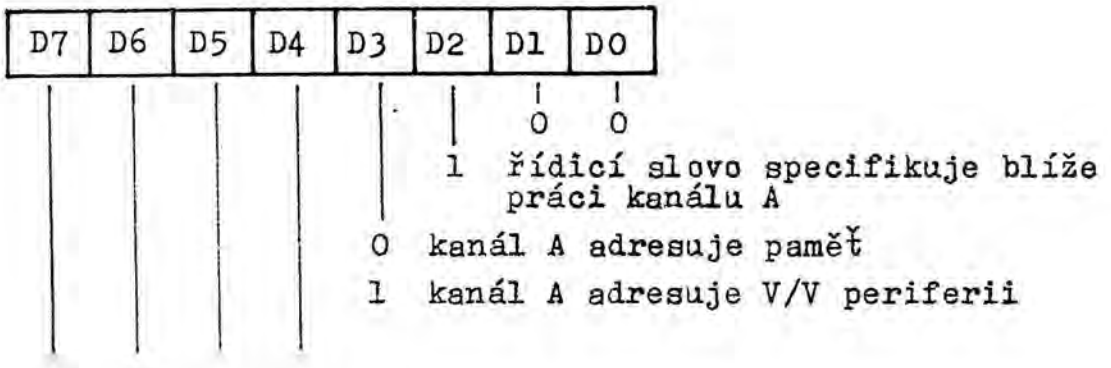
Řídicí registr R0

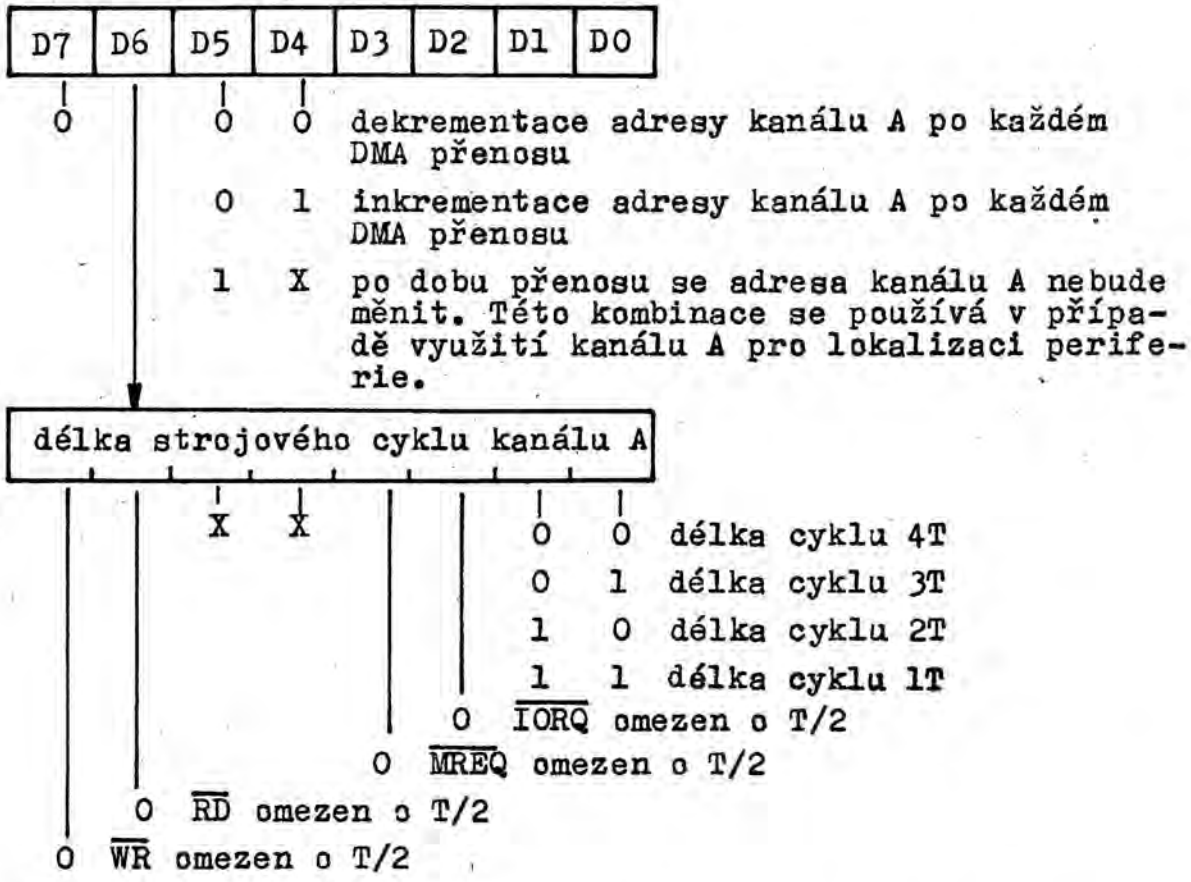
Registr R0 definuje základní režimy práce DMA obvodu. Příznakové bity registru R0 určují posloupnost zadávání startovací adresy kanálu A a přenosové délky.



Řídicí registr R1

Registr R1 blíže určuje činnost kanálu A včetně bytu dodaného při aktivním příznakovém bitu, který lokalizuje délku DMA cyklu.



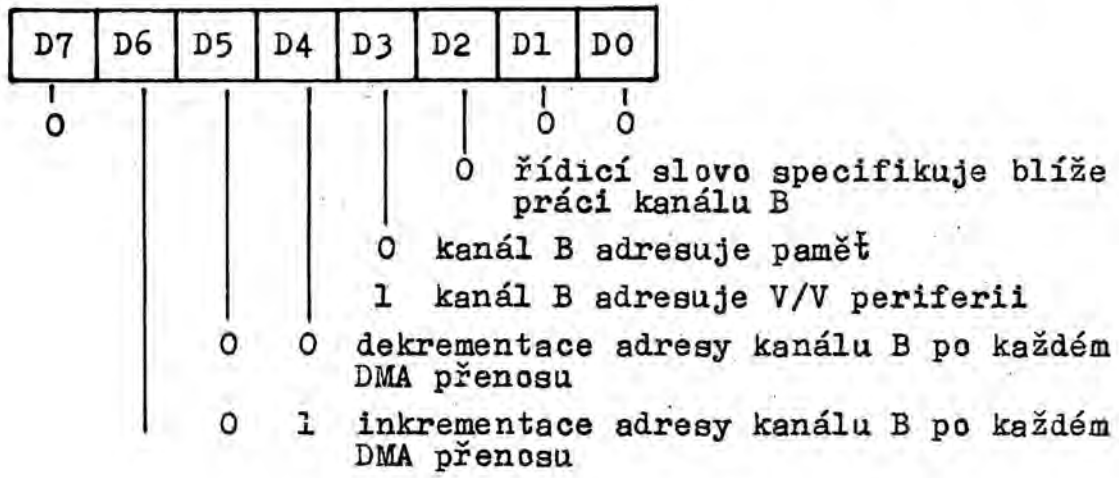


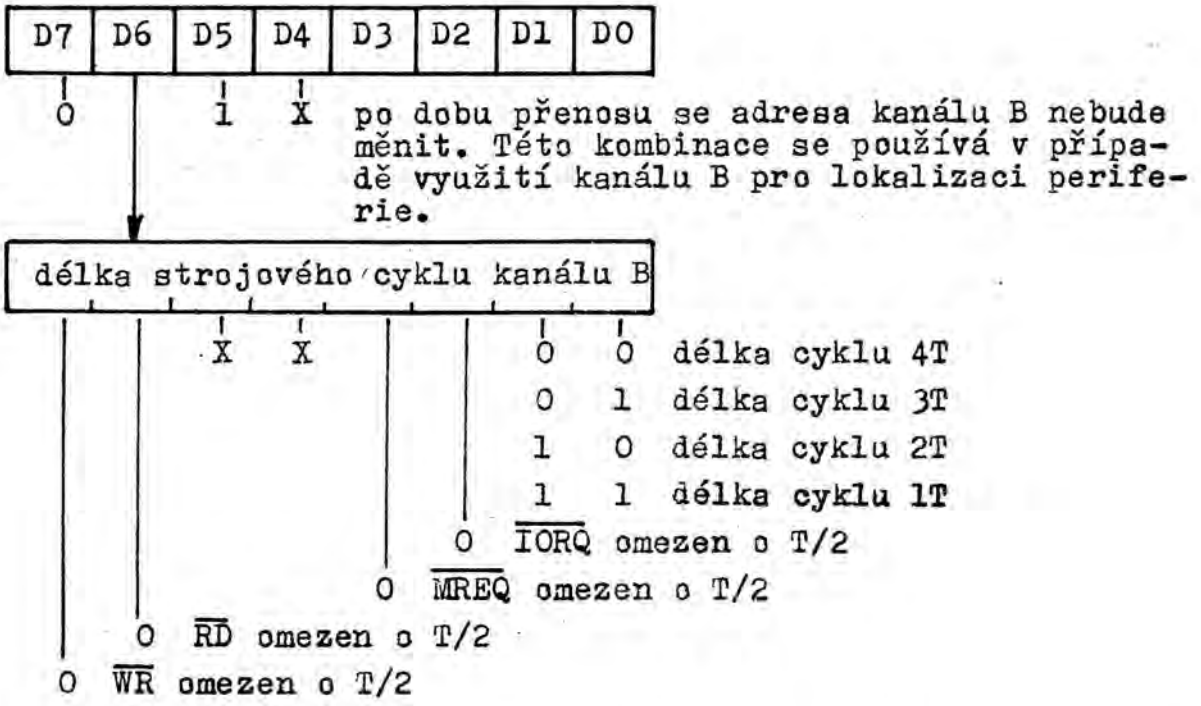
Poznámka

Délka přenosové operace čtení/zápis může mít minimální délku 2T. Stejnou minimální dobu má operace vyhledávání, při které je generován pouze cyklus čtení.

Řídicí registr R2

Registr R2 blíže určuje činnost kanálu B včetně bytu, dodaného při aktivním příznakovém bitu.



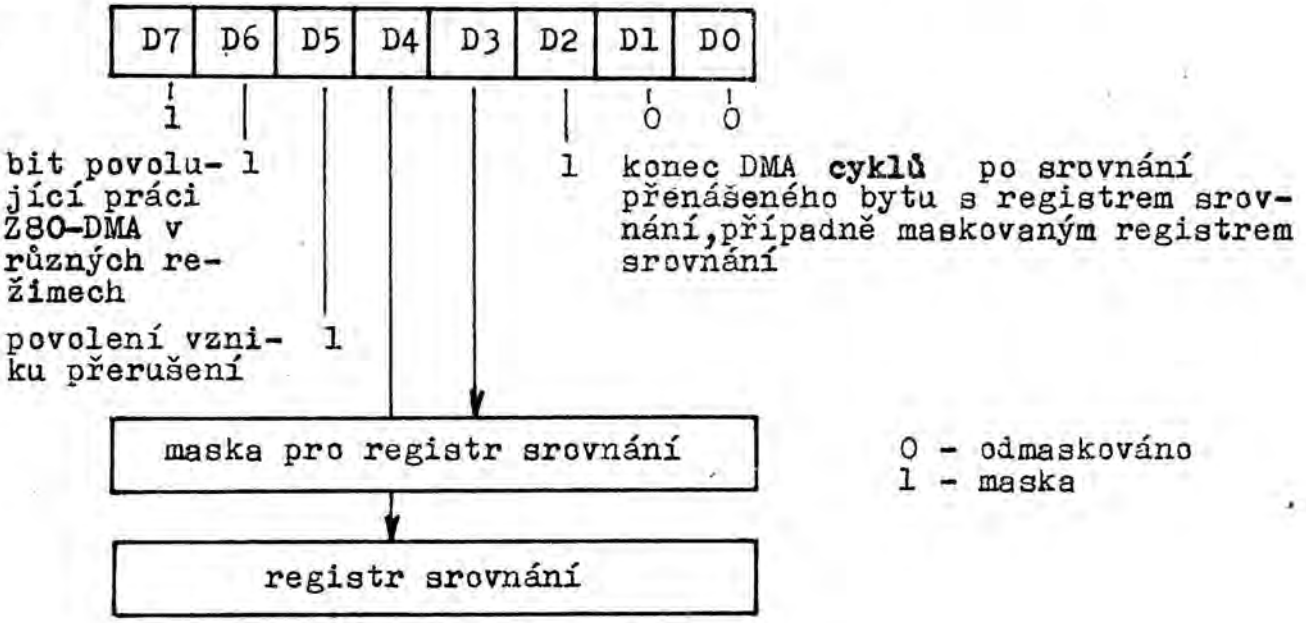


Poznámka

Délka přenosové operace čtení/zápis může mít minimální délku 2T. Stejnou minimální dobu má operace vyhledávání, při které je generován pouze cyklus čtení.

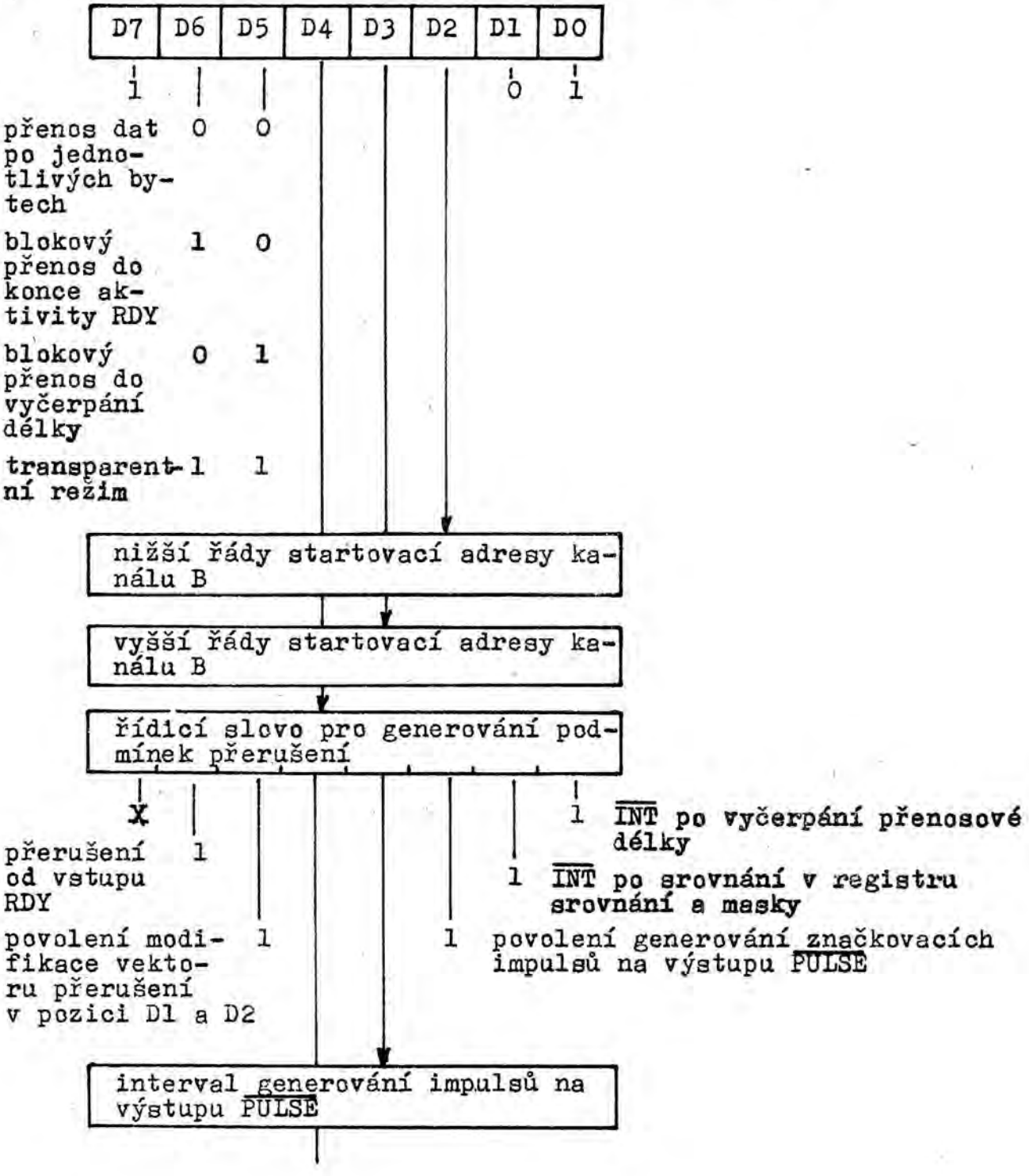
Řídicí registr R3

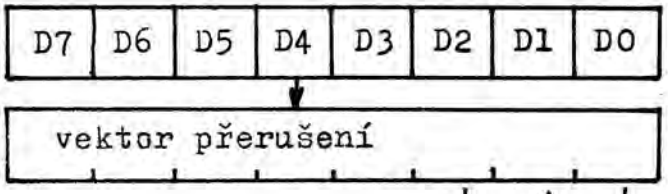
Registr R3 povoluje DMA přenos, přerušení a v režimu vyhledávání posloupnost následujících bytů,tj. bytu,který plní registr masky a registr srovnání.



Řídicí registr R4

Registr R4 určuje způsob vedení přenosu - blokově nebo po bytech. Příznakové bity definují jednak startovací adresu kanálu B, varianty generování přerušení na výstupu $\overline{INT}$ a délku intervalu pro vznik značkovacího impulsu na výstupu $\overline{PULSE}$.



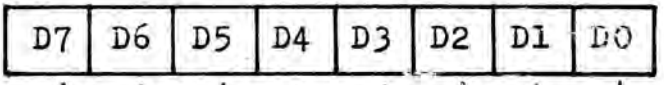


modifikace vektoru přerušení, je-li povolena bitem D5 výše uvedeného slova

0	0	přerušení od RDY
0	1	přerušení od registru srovnání a masky
1	0	přerušení od vyčerpání přenosové délky
1	1	přerušení od registru srovnání a masky nebo od vyčerpané délky

Řídicí registr R5

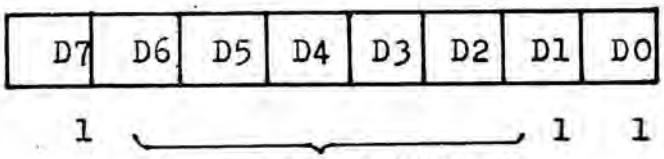
Registr R5 především definuje aktivní úroveň signálu RDY a přiřazení významu vstupu $\overline{CE/WAIT}$.



- 0 aktivita RDY = 0
- 1 aktivita RDY = 1
- 0 vstup $\overline{CE/WAIT}$ má výhradně význam $\overline{CE}$
- 1 vstup $\overline{CE/WAIT}$ má význam buď $\overline{CE}$, nebo $\overline{WAIT}$
- 0 stop činnosti po vyčerpání bloku bez obnovy parametru
- 1 na konci bloku dojde k znovunastavení přenosové délky a startovacích adres obou kanálů

Řídicí registr R6

Naadresování řídicího slova R6 zabezpečí bity D0, D1, D7=1. Pozice D2 ÷ D6 mohou nabývat 16 kombinací, jejichž význam vyjadřuje připojená tabulka. V tabulce jsou uvedeny také profily registru R6. **Obvod je tímto slovem zároveň inicializován.**

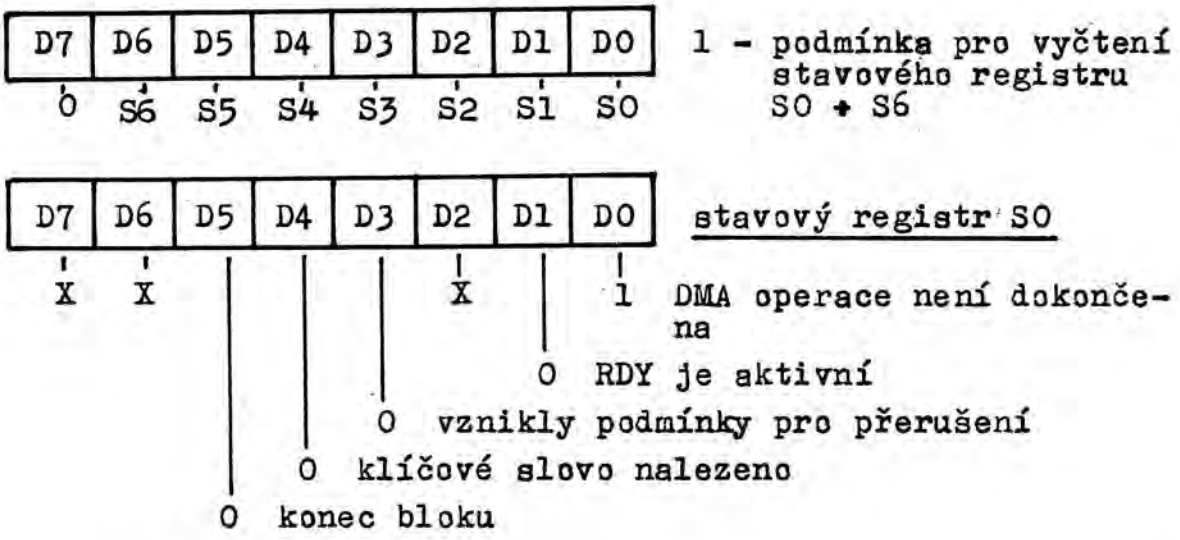


D7	D6	D5	D4	D3	D2	D1	D0	
1						1	1	
HEX								
83	0	0	0	0	0			dočasné blokování funkce DMA řadiče dle předcházejícího naprogramování tj. nastavená funkce obvodu po odblokování je stále platná
87	0	0	0	0	1			povolení funkce DMA řadiče dle předcházejícího naprogramování
8B	0	0	0	1	0			nulování bitu D4 a D5 stavového registru S0
A3	0	1	0	0	0			stejný vliv jako má instrukce RETI
A7	0	1	0	0	1			start vyčítání stavových registrů povolených maskou čtení registrů stavu
AB	0	1	0	1	0			povolení přerušení
AF	0	1	0	1	1			blokování přerušení
B3	0	1	1	0	0			vyvolání vnitřního signálu RDY (nyní nezávislého na vstupním signálu RDY) pro zabezpečení přenosu paměť - paměť
B7	0	1	1	0	1			povolení funkce DMA řadiče až po příchodu instrukce RETI
BB	0	1	1	1	0			následující zápisová operace je vložení masky čtení registrů stavu
BF	0	1	1	1	1			povolení čtení stavového registru S0 v následující čtecí operaci
C3	1	0	0	0	0			inicializace DMA řadiče, blokování aktivity výstupů $\overline{\text{BUSRQ}}$, $\overline{\text{INT}}$, $\overline{\text{WAIT}}$, vnitřních podmínek RDY. Potlačení automatického RESTARTu na konci bloku.
C7	1	0	0	0	1			inicializace brány A se standardním rytmem časování dle Z80-CPU
CB	1	0	0	1	0			inicializace brány B se standardním rytmem časování dle Z80-CPU
CF	1	0	0	1	1			zápis startovací adresy kanálu A a kanálu B do čítačů adresy a znovunastavení čítače délky z registru délky
D3	1	0	1	0	0			pokračování v činnosti od adres nastavených po DMA přenosu v kanálu A a B s nutností obnovy stavu vynulovaného registru přenosové délky

Stav jednokanálového řadiče DMA lze zjistit čtecími operacemi. Celkem lze vyčítat 7 vnitřních registrů označených S0 + S6. Registr S0 je typickým stavovým registrem, kdežto registry S1 + S6 vyjadřují okamžitý stav čítače přenosové délky a adresních registrů kanálu A a B.

V jaké posloupnosti jsou stavové registry přenášeny do Z80-CPU určí registr masky, jehož příchod následuje po řídícím slově R6 s obsahem BB. Pouhé vyčtení registru S0 do mikroprocesoru je podmíněno řídícím slovem R6 s profilem BF.

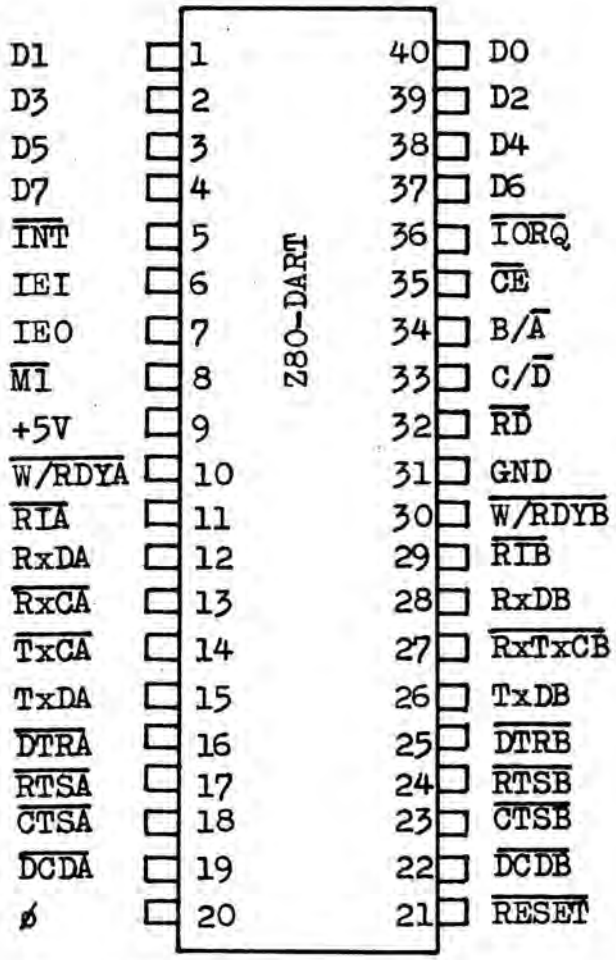
Maska čtení registrů stavu



nižší řády čítače přenosové délky	stavový registr S1
vyšší řády čítače přenosové délky	stavový registr S2
nižší řády adresy kanálu A	stavový registr S3
vyšší řády adresy kanálu A	stavový registr S4
nižší řády adresy kanálu B	stavový registr S5
vyšší řády adresy kanálu B	stavový registr S6

4.5 Z80-DART

Obvod Z80-DART (DUAL ASYNCHRONOUS RECEIVE/TRANSMITTER) vznikl zjednodušením prvku Z80-SIO/O o bloky, které zajišťují synchronní přenosy. Obvod je dvoukanálový, asynchronní vysílač/přijímač. Zapojení vývodů pouzdra odpovídá obvodu SIO/O až na vodiče $\overline{\text{SYNC A}}$ a $\overline{\text{SYNC B}}$, které jsou u Z80-DART nahrazeny signály $\overline{\text{RIA}}$ a $\overline{\text{RIB}}$. Vstupní signály $\overline{\text{RIA}}$ a $\overline{\text{RIB}}$ indikují stav přenosové smyčky a slouží jako zdroje přerušení. Pouzdro je 40vývodové, napájené z + 5 V a je na obr. 107.



Obr. 107 Zapojení vývodů pouzdra Z80-DART

Stručná charakteristika pouzdra:

- dva duplexní asynchronní kanály A a B
- 5 řídicích registrů pro kanál A
- 2 stavové registry pro kanál A
- 6 řídicích registrů pro kanál B
- 3 stavové registry pro kanál B
- každý kanál disponuje 4 řídicími signály pro zajištění funkce MODEMu
- generování přerušení na základě obsazení vysílací a přijímací cesty
- generování přerušení externími podmínkami $\overline{\text{RIA}}$ a $\overline{\text{RIB}}$
- možnost generování požadavku na zahájení DMA přenosu
- kontrola asynchronního přenosu paritou lichou, sudou nebo přenos probíhá bez kontroly
- zabezpečení přenosu znaků 1, 1,5, 2 stop bity
- frekvence hodin vůči přenosové frekvenci 1, 16, 32, 64 : 1
- na příjmu indikace chyb parity, formátu přenosu a přeběhu, tj. PE, FE, OE

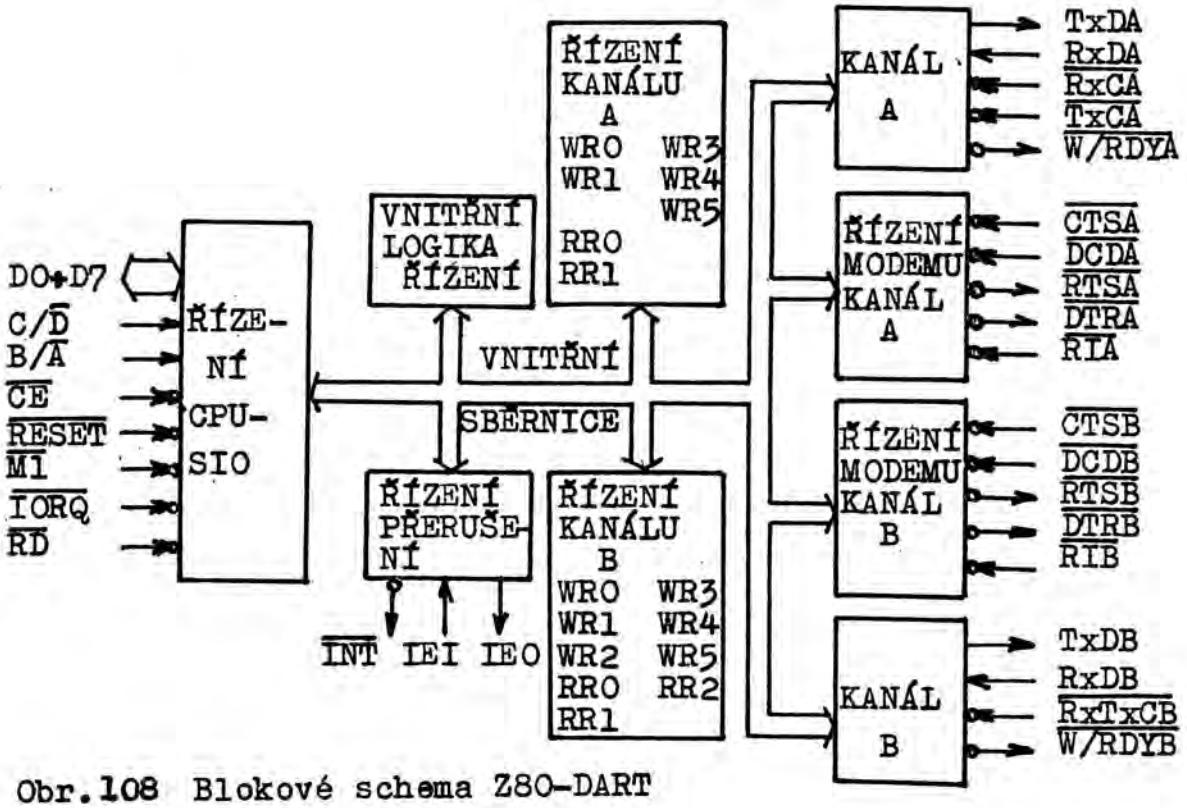
4.5.1 Architektura obvodu

Vnitřní struktura Z80-DART je obdobná obvodu Z80-SIO, jsou v ní ale vynechány bloky pro zajištění synchronních protokolů tj. např. blok pro generování a testování CRC znaků, registry pro vkládání jednoho nebo dvou synchronizačních znaků v režimu MONOSYNC nebo BISYNC, případně sloužících pro zanesení FLAG znaku a osmibitové adresy při SDLC a HDLC protokolu. Blok řízení kanálu A neobsahuje řídicí registr WR2 ani stavový registr RR2. Tyto registry jsou obsaženy v bloku řízení kanálu B, kde slouží pro zanesení vektoru přerušení (WR2) a pro vyčtení modifikovaného vektoru přerušení (RR2).

Blokové schéma obvodu je na obr. 108.

4.5.2 Programování Z80-DART

Kanál A a B se programují samostatně pomocí stavu adresního vodiče B/$\overline{\text{A}}$. Řídicí slova jsou označena symboly WR0 ÷ WR5 a stavová slova symboly RR0 ÷ RR2. Oproti prvku Z80-SIO se



Obr.108 Blokové schema Z80-DART

neprogramují slova WR6 a WR7, která sloužila výhradně pro synchronní režimy.

Řídicí registr WR0 obsahuje v pozici D0 ÷ D2 ukazatel řídicího registru. Registr odpovídající ukazateli bude plněn následujícím bytem zaslaným po řídicím slově WR0. Po RESETu probíhá zápis do registru WR0 nezávisle na ukazateli D0 ÷ D2.

Řídicí registr WR0

Mimo ukazatele na registr definuje několik variant nulování kanálu, povolení přerušení atd.

D7	D6	D5	D4	D3	D2	D1	D0	
					0	0	0	WR0
					0	0	1	WR1
					0	1	0	WR2
					0	1	1	WR3
					1	0	0	WR4
					1	0	1	WR5

ukazatel na řídicí registry

D7	D6	D5	D4	D3	D2	D1	D0

		0	0	0	**bez vlivu (NULL CODE)**
		0	0	1	nepoužito
		0	1	0	nulování EXT/STATUS přerušení
		0	1	1	nulování lokalizovaného kanálu
		1	0	0	povolení přerušení od dalšího znaku
		1	0	1	nulování přerušení od vysílače
		1	1	0	nulování chyb v přenosu
		1	1	1	návrat z přer. (jako RETI) - kanál A
X	X	nepoužity			

Řídicí registr WR1

Nastavuje způsoby přenosů vysílače, přijímače a přerušení.

D7	D6	D5	D4	D3	D2	D1	D0

							1 povolení EXT/STATUS přerušení
						1	povolení přerušení od prázdného vysílače
					1		povolení modifikace vektoru přerušení kanálu B **a A**
			0	0			blokování vzniku přerušení od přijímače
			0	1			povolení přerušení od 1. znaku přijímače
			1	0			povolení přerušení od každého znaku
			1	1			povolení přerušení od každého znaku, generování fixního vektoru
		0	$\overline{W/RDY}$ = 0 (vysílač je plný)				
		1	$\overline{W/RDY}$ = 0 (přijímač je prázdný)				
	0	$\overline{W/RDY}$ je ve funkci $\overline{WAIT}$					
	1	$\overline{W/RDY}$ je ve funkci $\overline{RDY}$					
0	povolení výstupu $\overline{W/RDY}$						
1	výstup $\overline{W/RDY}$ setrvá v log. 1						

Poznámka

Na základě bitu D2 = 1 řídicího slova WR1 je zabezpečena modifikace vektoru přerušení.

Modifikace vektoru přerušení kanálu B

Modifikace vektoru přerušení kanálu A

V3	V2	V1	
0	0	0	vysílací registry jsou prázdné
0	0	1	změna EXT/STATUS podmínek
0	1	0	znak byl přijmut
0	1	1	speciální podmínky přijímače
1	0	0	vysílací registry jsou prázdné
1	0	1	změna EXT/STATUS podmínek
1	1	0	znak byl přijmut
1	1	1	speciální podmínky přijímače

Řídicí registr WR2

Registr obsahuje dodaný vektor přerušení. Umístění registru je pouze v kanálu B. Bity D0, D4 ÷ D7 nelze ovlivnit vnitřní přerušovací podmínkou. Bity D1 ÷ D3 lze modifikovat dle výše uvedené tabulky, je-li bit D2 = 1 v řídicím slově WR1.

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	MV3	MV2	MV1	0

dodaný vektor

modifikace vektoru přerušení při D2 = 1 slova WR1

Řídicí registr WR3

Nastavení bitů řídicího registru WR3 řídí činnost přijímače.

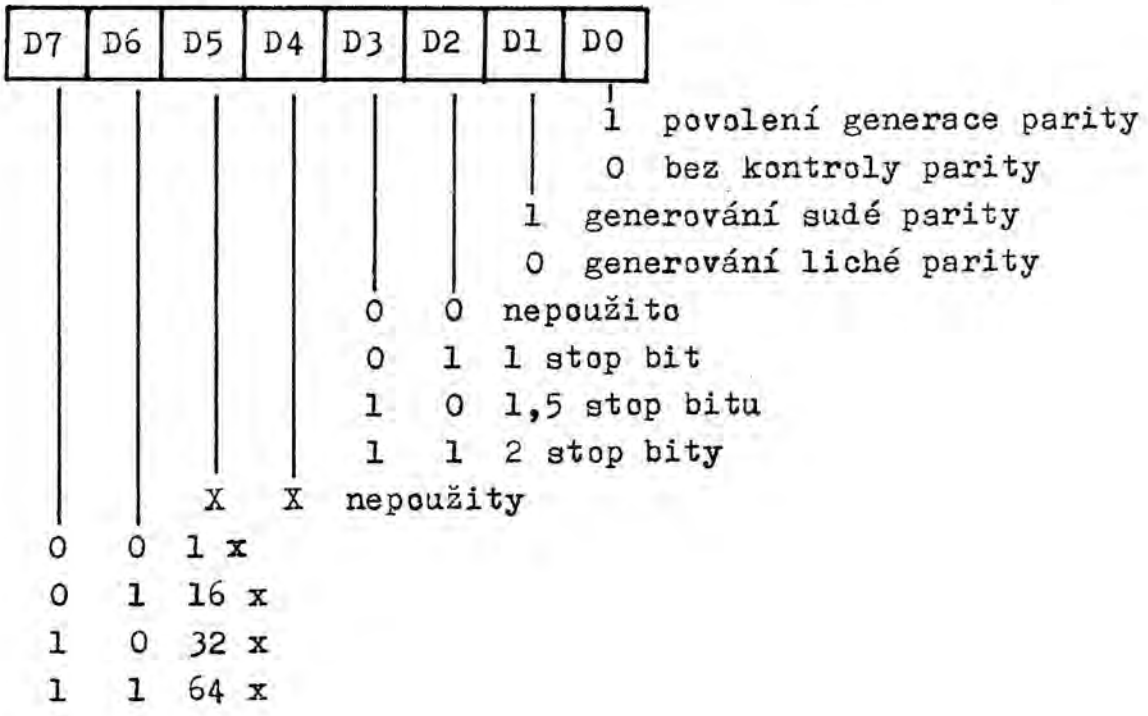
D7	D6	D5	D4	D3	D2	D1	D0	
							1	povolen příjem znaků
			X	X	X	X		nepoužity
		1						signály $\overline{DCD}$ a $\overline{CTS}$ povolují příjem a vysílání znaků

Délka přijímaného znaku

D7	D6	
0	0	5 bitů
0	1	7 bitů
1	0	6 bitů
1	1	8 bitů

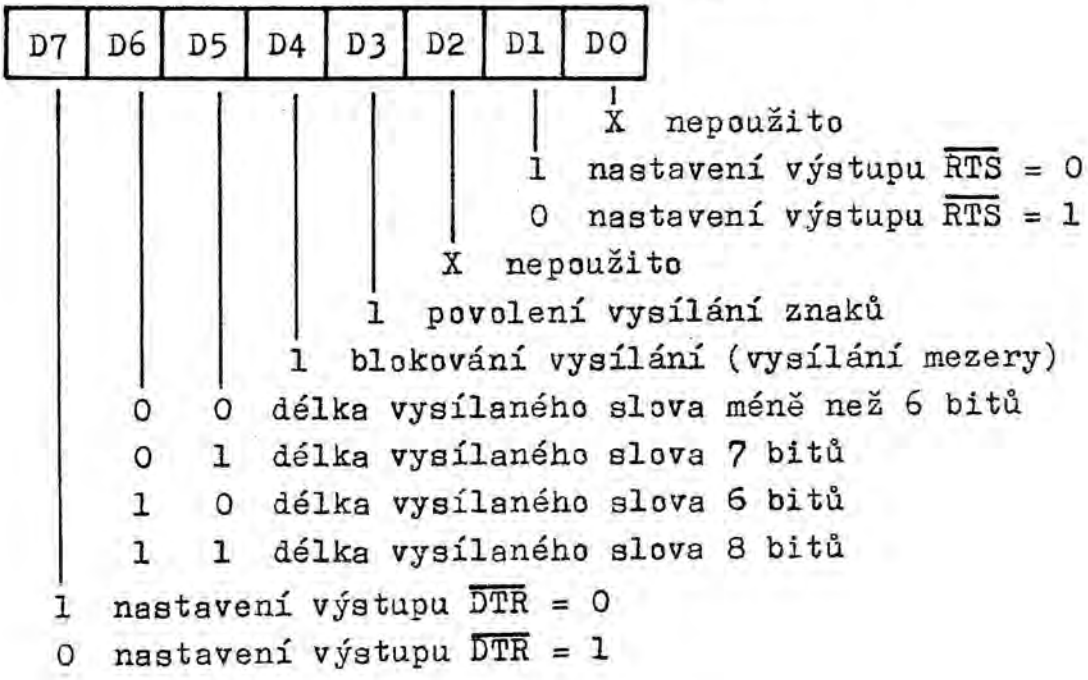
Řídicí registr WR4

Jednotlivé bity řídicího registru WR4 ovlivňují činnost vysílací a přijímací cesty.



Řídicí registr WR5

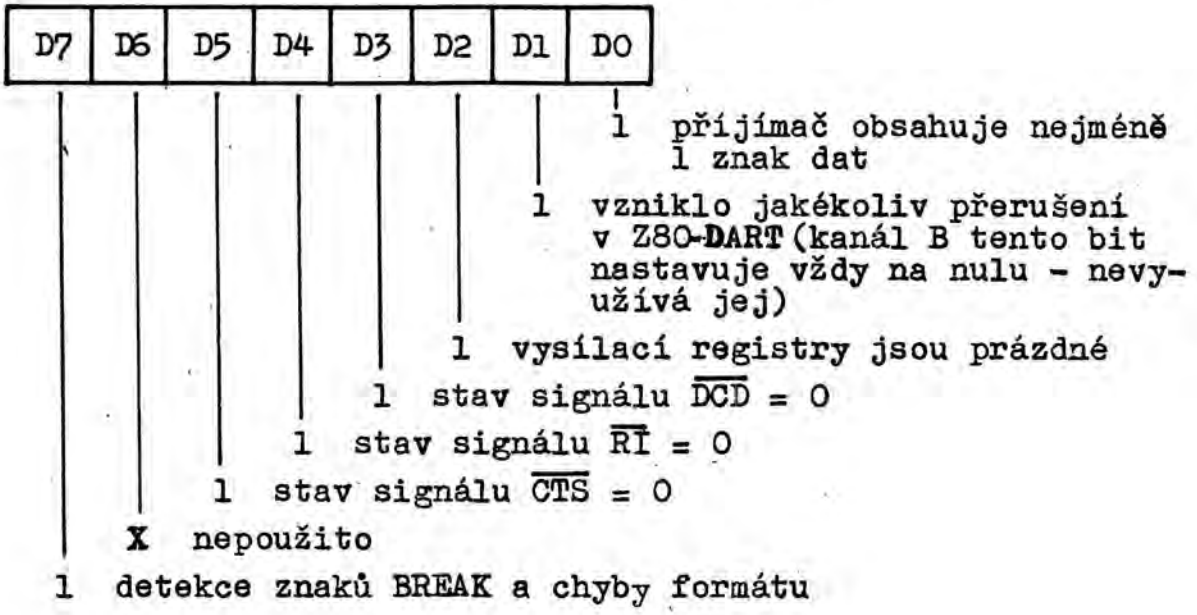
Obsah tohoto registru řídí činnost vysílače.



Následuje přehled 3 stavových registrů RR0 + RR2. Číslo stavového registru, který je přenesen do Z80-CPU, definuje opět ukazatel v pozici D0 + D2 řídicího registru WR0. Registr RR2 je obsažen pouze v kanálu B.

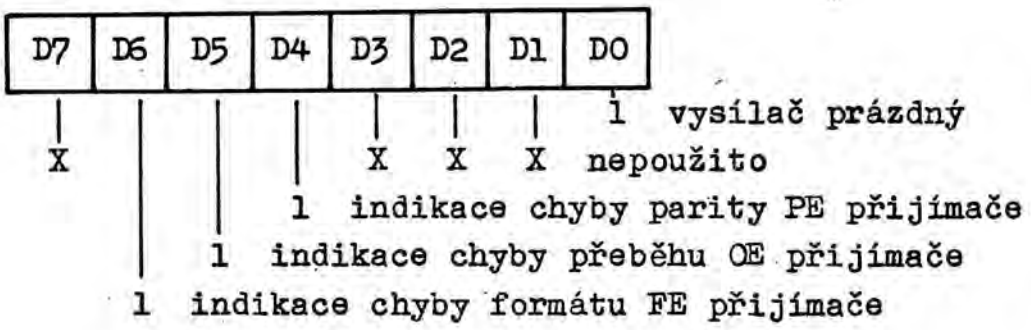
Stavový registr RR0

Registr RR0 informuje mikroprocesor o stavu vysílače, přijímače, modemových vstupních signálech a přerušení.



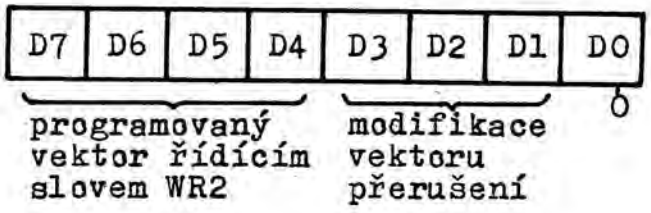
Stavový registr RR1

Registr RR1 dodává do mikroprocesoru stav speciálních podmínek přijímače.



Stavový registr RR2

Obsah registru RR2 je možné získat jen vyčtením z kanálu B. Stavový registr přenáší modifikovaný vektor přerušení pro oba kanály, byla-li modifikace vektoru povolena bitem D2 řídicího slova WR1.



Na závěr rozboru Z80-DART jsou uvedeny formáty datových bytů v případě, že jde o přenos 5 až 1 bitové informace, definované v řídícím slově WR5 bity D5 = 0 a D6 = 0. Formát přijmutého bytu od Z80-CPU je tvořen na nejnižších pozicích významovými bity, následují vždy 3 bity nul a zbylé pozice bytu jsou doplněny jedničkami. Např. z přijmutého bytu profilu 11110001 je odvysílán pouze nejnižší bit,tj. pozice D0, protože se jedná o jednobitový znak.

D7	D6	D5	D4	D3	D2	D1	D0	délka datového znaku
1	1	1	1	0	0	0	d	1
1	1	1	0	0	0	d	d	2
1	1	0	0	0	d	d	d	3
1	0	0	0	d	d	d	d	4
0	0	0	d	d	d	d	d	5

Poznámka

Symbol d je hodnota vysílaného bitu v rozsahu 1 ÷ 5 bitového znaku.

Profil přišlý od Z80-CPU musí mít normalizovaný tvar, protože Z80-DART programově neví zda bude přijímat od mikroprocesoru znak 5, 4, 3, 2, 1 bitový. Tento rozsah znaku musí obvod poznat na základě 3 nul, které pošle mikroprocesor v normalizovaném tvaru.

LITERATURA

1. William Barden - THE Z-80 MICROCOMPUTER HANDBOOK
2. TECHNICAL MANUAL ZILOG Z-80 - 1980
3. COMPONENT DATA CATALOG INTEL - 1980
4. Bayer, Bílek - MIKROPROCESOR Z-80, Dům techniky ČSVTS Ústí nad Labem, 1984
5. Müller, Muselík - MIKROPOČÍTAČOVÉ SYSTÉMY S OBVODY ŘADY Z-80, Dům techniky ČSVTS Brno, 1981
6. TECHNICAL MANUAL MOSTEK - Z80 MICROCOMPUTER DEVICES, MK 3880 CENTRAL PROCESSING UNIT
7. Z-80 MICROPROCESSOR, PROGRAMMING & INTERFACING BOOK1 - 1979
8. Z-80 MICROPROCESSOR, PROGRAMMING & INTERFACING BOOK2 - 1980
9. Dlabola, Starý - SYSTÉMY S MIKROPROCESORY A PŘENOS DAT, NADAS, 1986

OBSAH

		Str.
ÚVOD		3
1. ZÁKLADNÍ POJMY Z MIKROPOČÍTAČOVÉ TECHNIKY		5
1.1	Mikropočítačový systém	5
1.2	Struktura mikroprocesoru	7
1.3	Načtení a zpracování instrukce	11
1.4	Zásobníková paměť a podprogramy	14
1.5	Časování mikroprocesoru	17
1.6	Synchronizace mikroprocesoru s vnějšími bloky	18
1.7	Komunikace se vstupy/výstupy	20
1.7.1	Adresování vstupů/výstupů	21
1.8	Přerušení programu	23
1.9	Konstrukce výstupů LSI obvodů	26
1.10	Přímý přístup do paměti	31
1.11	Polovodičové paměti	33
1.11.1	Členění polovodičových pamětí	33
1.11.2	Typické parametry polovodičových pamětí	39
2. MIKROPROCESOR Z80-CPU		42
2.1	Mikropočítačový systém	42
2.2	Součástkový přehled řady Z-80 (U880)	45
2.3	Srovnání řady Z-80 s I 8080	46
2.4	Architektura Z-80 CPU	49
2.4.1	Vnitřní bloky CPU	50
2.4.2	Přehled uspořádání vývodů pouzdra	54
2.5	Časový průběh vykonání instrukce	59
2.5.1	Čtení operačního kódu instrukce	61
2.5.2	Čtení/zápis dat do paměti	61
2.5.3	Čtení/zápis dat do V/V zařízení	63
2.5.4	Cyklus požadavku/potvrzení požadavku na převzetí řízení sběrnic	64

Str.

2.5.5 Cyklus požadavku/potvrzení požadavku na přerušení 65

2.6 Přerušovací systém 69
2.7 Obnovování stavu dynamických pamětí 78
2.8 Spolupráce s paměťovými a periferními obvody 78
2.9 DMA přenos 79

3. INSTRUKČNÍ SOUBOR 81

3.1 Způsoby adresování 82

3.1.1 Adresování s přímým operandem 82
3.1.2 Přímé adresování 82
3.1.3 Indexové adresování 82
3.1.4 Relativní adresování 83
3.1.5 Adresování nulté stránky 83
3.1.6 Adresování registru 83
3.1.7 Nepřímé adresování šestnáctibitovým registrem 84
3.1.8 Adresování bitu v registru 84

3.2 Indikátory stavů 84

3.2.1 Indikátor přenosu - bit C (CARRY) 85
3.2.2 Indikátor sčítání/odčítání - bit N (NEGATIVE) 85
3.2.3 Indikátor parity/přetečení - bit P/V (PARITY/OVERFLOW) 85
3.2.4 Indikátor polovičního přenosu - bit H (HALF CARRY) 85
3.2.5 Indikátor nuly - bit Z (ZERO) 86
3.2.6 Indikátor znaménka - bit S (SIGN) 86

3.3 Tabulky instrukcí 86

3.3.1 Instrukce přenosu osmibitových dat (LOAD) 86
3.3.2 Instrukce výměny (EXCHANGE) 87
3.3.4 Aritmetické instrukce pro osmibitová data 87

Str.

3.3.5 Logické instrukce 87
3.3.6 Instrukce skoků (JUMP) 87
3.3.7 Instrukce rotací a posuvů 88
3.3.8 Volání podprogramu 88
3.3.9 Návrat z podprogramu 88
3.3.10 Instrukce vstupu a výstupu 89
3.3.11 Instrukce blokových přenosů (BLOCK TRANSFER) 89
3.3.12 Instrukce vyhledání v blocích dat (COMPARE) 90
3.3.13 Instrukce bitových operací 90
3.3.14 Řídicí instrukce 91
3.3.15 Instrukční soubor Z80-CPU 91
3.3.16 Seznam instrukcí mikroprocesoru Z80-CPU podle operačních kódů 107
3.3.17 Abecední seznam instrukcí mikroprocesoru Z80-CPU 118
3.4 Rozbor instrukcí souboru Z-80 129
3.4.1 Instrukce výměny registrů 129
3.4.1.1 Výměna 129
3.4.1.2 Výměna skupiny registrů 130
3.4.2 Instrukce blokových přenosů 130
3.4.2.1 Blokový přenos s inkrementací bez opakování 131
3.4.2.2 Blokový přenos s inkrementací a opakováním 131
3.4.2.3 Blokový přenos s dekrementací bez opakování 132
3.4.2.4 Blokový přenos s dekrementací a opakováním 133
3.4.3 Instrukce vyhledávání v blocích dat 134
3.4.3.1 Blokové vyhledávání s inkrementací bez opakování 134

Str.

3.4.3.2 Blokové vyhledávání s inkrementací a opakováním 134
3.4.3.3 Blokové vyhledávání s dekrementací bez opakování 135
3.4.3.4 Blokové vyhledávání s dekrementací a opakováním 136

3.4.4 Instrukce pro přesun dat 137

3.4.4.1 Přesun dat 137
3.4.4.2 Čtení zásobníkové paměti 137
3.4.4.3 Zápis do zásobníkové paměti 139

3.4.5 Aritmetické instrukce 139

3.4.5.1 Aritmetické sčítání 139
3.4.5.2 Aritmetické sčítání s přenosem 140
3.4.5.3 Aritmetické odčítání 142
3.4.5.4 Aritmetické odčítání s přenosem 142
3.4.5.5 Dekadická úprava akumulátoru 143

3.4.6 Logické instrukce 145

3.4.6.1 Logický součin 145
3.4.6.2 Logický součet 146
3.4.6.3 Exkluzivní součet 146
3.4.6.4 Srovnání 147
3.4.6.5 Inkrementace 147
3.4.6.6 Dekrementace 148
3.4.6.7 Komplement 149
3.4.6.8 Negace 149

3.4.7 Řídicí instrukce 150

3.4.7.1 Programové nastavení režimu přerušení 150
3.4.7.2 Zastavení 150
3.4.7.3 Povolení přerušení 150
3.4.7.4 Blokování přerušení 151
3.4.7.5 Inverze indikátorového bitu C 151
3.4.7.6 Nastavení indikátorového bitu C 151
3.4.7.7 Prázdná operace 151

Str.

3.4.8 Instrukce rotací a posuvů 152

3.4.8.1 Rotace akumulátoru vlevo a do CY 152
3.4.8.2 Rotace akumulátoru vlevo přes CY 152
3.4.8.3 Rotace akumulátoru vpravo a do CY 153
3.4.8.4 Rotace akumulátoru vpravo přes CY 153
3.4.8.5 Rotace operandu vlevo a do CY 153
3.4.8.6 Rotace operandu vlevo přes CY 154
3.4.8.7 Rotace operandu vpravo a do CY 155
3.4.8.8 Rotace operandu vpravo přes CY 155
3.4.8.9 Aritmetický posuv vlevo 156
3.4.8.10 Aritmetický posuv vpravo 156
3.4.8.11 Posuv vpravo 157
3.4.8.12 Rotace čtveřic bitů vlevo 157
3.4.8.13 Rotace čtveřic bitů vpravo 158

3.4.9 Instrukce bitových operací 159

3.4.9.1 Nastavení bitu 159
3.4.9.2 Nulování bitu 159
3.4.9.3 Testování bitu 160

3.4.10 Skokové instrukce 160

3.4.10.1 Nepodmíněný skok 160
3.4.10.2 Podmíněný skok 161
3.4.10.3 Relativní nepodmíněný skok 161
3.4.10.4 Relativní podmíněný skok 161
3.4.10.5 Relativní skok podmíněný čítačem 162

3.4.11 Instrukce volání podprogramu nebo podprogramu přerušení 162
3.4.11.1 Nepodmíněné volání podprogramu 162
3.4.11.2 Podmíněné volání podprogramu 163
3.4.11.3 Instrukce RESTART 164

3.4.12 Instrukce návratu z podprogramu nebo podprogramu přerušení 164

3.4.12.1 Nepodmíněný návrat z podprogramu 164
3.4.12.2 Podmíněný návrat z podprogramu 164

Str.

3.4.12.3 Návrat z přerušení režimu 2 165
3.4.12.4 Návrat z nemaskovatelného přerušení 165
3.4.13 Instrukce vstupů a výstupů 165
3.4.13.1 Instrukce vstupu 166
3.4.13.2 Blokový vstup s inkrementací bez opakování 166
3.4.13.3 Blokový vstup s inkrementací a opakováním 167
3.4.13.4 Blokový vstup s dekrementací bez opakování 168
3.4.13.5 Blokový vstup s dekrementací a opakováním 168
3.4.13.6 Instrukce výstupu 169
3.4.13.7 Blokový výstup s inkrementací bez opakování 170
3.4.13.8 Blokový výstup s inkrementací a opakováním 170
3.4.13.9 Blokový výstup s dekrementací bez opakování 171
3.4.13.10 Blokový výstup s dekrementací a opakováním 171

4. PROGRAMOVATELNÉ PODPŮRNÉ OBVODY ŘADY Z-80 173

4.1 Z80-PIO 175
4.1.1 Popis vývodů pouzdra 176
4.1.2 Architektura obvodu 179
4.1.3 Režimy práce 181
4.1.4 Programování Z80-PIO 188

4.2 Z80-SIO 191
4.2.1 Přenosové protokoly 194
4.2.2 Popis vývodů pouzdra 196
4.2.3 Architektura obvodu 201
4.2.4 Spolupráce obvodu s Z80-CPU a s Z80-DMA 202

Str.

4.2.5 Časové rozbory základních signálů 206
4.2.6 Programování Z80-SIO 208

4.3 Z80-CTC 214

4.3.1 Popis vývodů pouzdra 215
4.3.2 Architektura obvodu 218
4.3.3 Režimy práce 220
4.3.4 Programování Z80-CTC 223

4.4 Z80-DMA 225

4.4.1 Popis vývodů pouzdra 225
4.4.2 Časové rozbory řízení DMA kanálu 230
4.4.3 Architektura obvodu 234
4.4.4 Režimy práce 234
4.4.6 Programování Z80-DMA 236

4.5 Z80-DART 244

4.5.1 Architektura obvodu 245
4.5.2 Programování Z80-DART 245

LITERATURA 252

OBSAH 253

Informace pro odběratele

Tato publikace je součástí ediční řady vydávané ČV elektrotechnické společnosti ČSVTS "Mikroprocesorová technika". Všechny tituly knižnice jsou zařazeny v oboru 735-2, a jsou určeny především pro kursy pořádané ČV elektrotechnické společnosti a ostatními odbornými orgány ČSVTS.

Minimální objednávané množství: 10 kusů od zvoleného titulu, nebo kompletní kolekce všech titulů "knižnice", které jsou na skladě /á 1 kus/ (celá kolekce obsahuje cca 25 titulů, cena kolekce je cca 350,-Kčs). **Vyřizujeme pouze objednávky socialistických organizací!!**

Objednávky, které splňují výše uvedenou podmínku **"minimálního objednaného množství"** vyřizuje:

TESLA ELTOS o.p.
závod Uherský Brod
Umanského 141
668 19 Uherský Brod

Aktuální nabídka titulů bývá příležitostně zveřejňována v Technickém Týdeníku v rubrice Zpravodaj ČSVTS.

Žádosti o zaslání informací o způsobu objednávání a aktuální nabídku titulů vyřizujeme **pouze písemně** na adrese:

Český výbor elektrotechnické společnosti ČSVTS
Novotného lávka 5
116 68 Praha 1

K žádosti o informace připojte obálku A5, nadepsanou Vaší zpáteční adresou!! Bez splnění této podmínky nejsme z kapacitních důvodů schopni Vaší žádosti vyhovět!!!

Český výbor elektrotechnické společnosti ČSVTS

MIKROPROCESOR Z 80 A PROGRAMOVATELNÉ OBVODY

Knižnice :	ČSVTS "Mikroprocesorová technika"
Autor :	Vladimír Dvořák
Recenzent :	Ing.Pavel Bukovský, Ing.Pavel Valášek,CSc.
Vydal :	Český výbor elektrotechnické společnosti ČSVTS
Náklad :	5 000, stran 259
Rok :	1988 - číslo publikace 60/840A/88
Distribuce :	TESLA ELTOS - zásilková služba,Uherský Brod
Tisk :	Tiskařské závody,n.p.,závod 3,Slezská 13,Praha 2

Vydala ČSVTS
ve Vydavatelství a nakladatelství Novinář Praha v roce 1988.