



UKŁAD 8-BITOWEGO MIKROPROCESORA

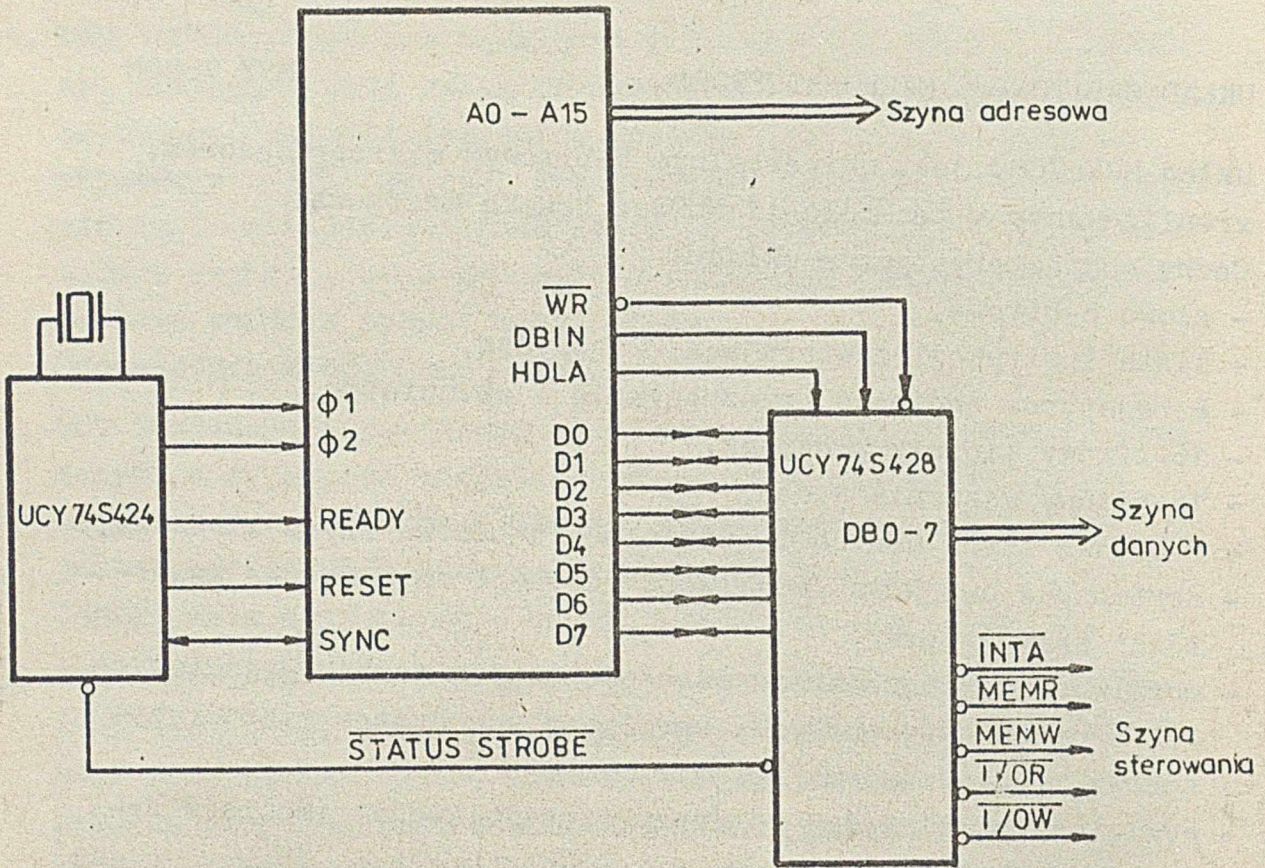
MCY 7880N

Układ MCY 7880 jest uniwersalnym 8-bitowym mikroprocesorem, zrealizowanym w technologii NMOS z bramką krzemową.

Cechy charakterystyczne układu:

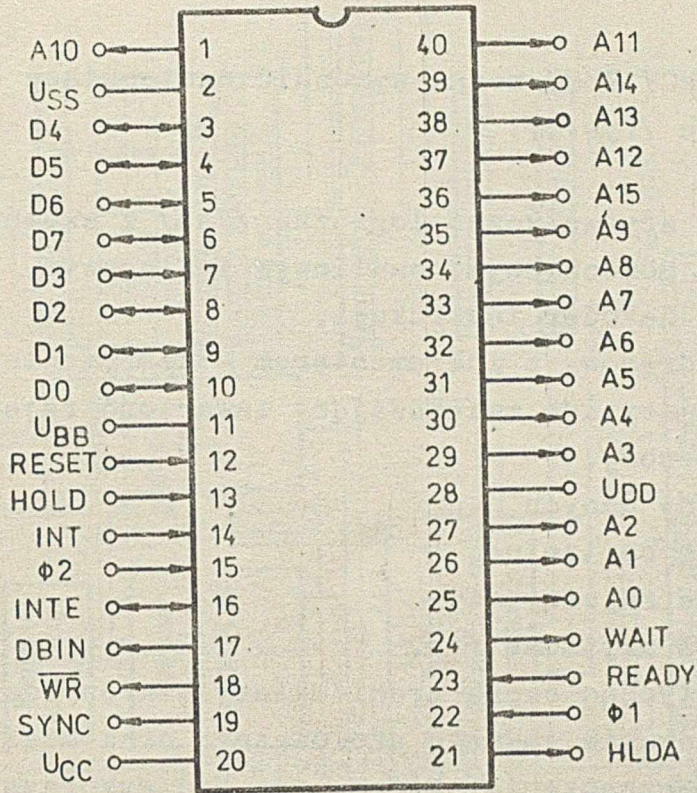
- słowo 8-bitowe,
- lista instrukcji zawierająca 72 pozycje,
- 6 rejestrów ogólnego przeznaczenia i akumulator,
- 16-bitowy licznik programu,
- 16-bitowy wskaźnik stosu,
- 8-bitowa szyna danych i oddzielna 16-bitowa szyna adresowa,
- arytmetyka dwójkowa /możliwość korekcji dziesiętnej zawartości akumulatora/,
- możliwość bezpośredniego adresowania 256 portów wejściowych i 256 portów wyjściowych, sprzęgających system mikroprocesorowy z urządzeniami zewnętrznymi,
- cykl instrukcji $2 \div 9 \mu\text{s}$ /przy częstotliwości impulsów zegarowych 2 MHz/,
- możliwość pracy w urządzeniach z wektorowym systemem przerwań,
- zgodność sygnałów "0" i "1" ze standardem TTL.

Układ MCY 7880 wraz z układami UCY 74S424 /zegar/ i UCY 74S428 /kontroler/ stanowi kompletną jednostkę centralną systemu mikroprocesorowego /rys. 1/. Oprócz mikroprocesora system musi zawierać jeszcze pamięć z wpisanym programem, np. ROM MCY 7304 lub ROM MCY 7316, jak również może być wyposażony w pamięć o dostępie swobodnym, przeznaczoną dla danych i stosu, np.



Rys. 1. Jednostka centralna systemu mikroprocesorowego opartego na MCY 7880

RAM MCY 7102 lub MCY 7114. W zależności od konfiguracji może także zawierać następujące układy pomocnicze: MCY 7851, MCY 7855, UCY 74S405, UCY 74S412, UCY 74S414 i UCY 74S416. W typowym systemie mikroprocesorowym, zawierającym układ UCY 74S428 wszystkie pamięci sterowane są dwoma sygnałami $\overline{I/O}$ i $\overline{I/O}$. Pola adresowe pamięci programu, pamięci danych i stosu muszą być w tym przypadku rozłączne. Konstrukcja układu MCY 7880 dopuszcza możliwość pracy systemu mikroprocesorowego w trybie DMA. Do tego celu służy sygnał HOLD, wprowadzający szynę danych i szynę adresową mikroprocesora w stan wysokiej impedancji.



Rys.2. Rozkład i opis wyprowadzeń układu MCY 7880

DO-D7 - dwukierunkowa trójstanowa szyna danych; DO-bit najmłodszy,

A0-A15- trójstanowa szyna adresowa; A0-bit najmłodszy,

SYNC - sygnał ważności słowa statusu na szynie danych,

DBIN - sygnał strobujący odczyt z pamięci lub układów we/wy,

WR - sygnał ważności danych na szynie DO-D7 podczas zapisu do pamięci lub układów we/wy,

READY - sygnał gotowości pamięci lub układów we/wy do przyjęcia lub wysłania bajtu danych,

WAIT - sygnał oczekiwania mikroprocesora na gotowość pamięci lub układów we/wy do współpracy,

HOLD - sygnał wprowadzający MCY 7880 w stan wstrzymania,

HLDA - sygnał potwierdzający stan wstrzymania,

INT - sygnał zgłoszenia przerwania,

INTE - sygnał zezwolenia na przerwanie,

RESET - sygnał wprowadzający MCY 7880 w stan początkowy,

phi1, phi2- sygnały zegarowe.

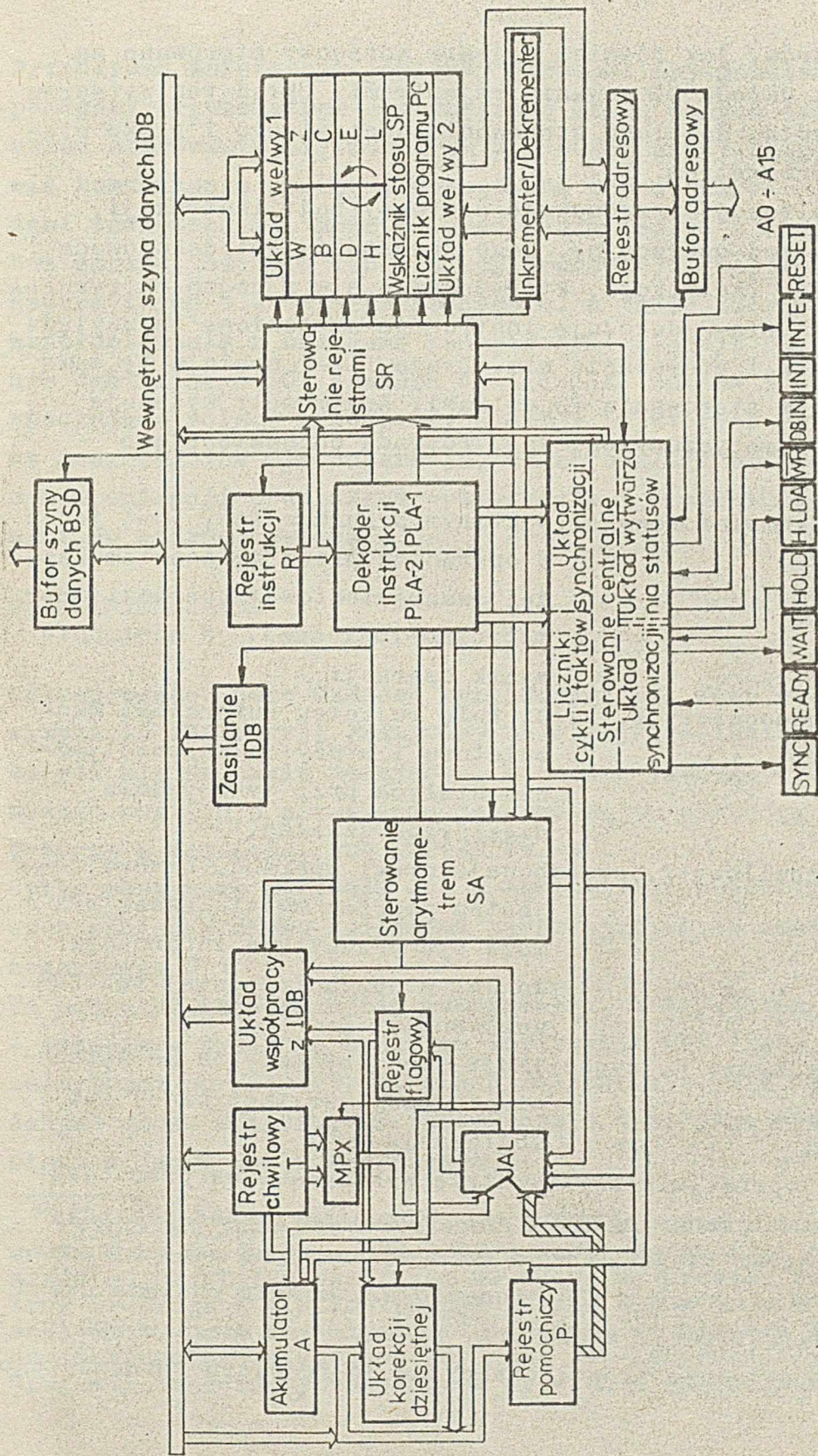
USS, UDD, UCC, UBB - zasilanie.

W układzie MCY 7880 można wyróżnić następujące bloki funkcjonalne /rys.3/:

- rejestry,
- jednostkę arytmetyczno-logiczną /JAL/ z akumulatorem, rejestrem pomocniczym, chwilowym i flagowym,
- rejestr i dekodér instrukcji,
- rejestr adresowy z inkrementerem - dekrementerem /układ zmniejszający lub zwiększający zawartość rejestru o 1/,
- bufor adresowy,
- bufor szyny danych,
- sterowanie centralne,
- sterowanie rejestrami,
- sterowanie arytmometrem.

Charakterystyczną cechą architektury mikroprocesora MCY 7880 jest zastosowanie jednego akumulatora oraz sześciu rejestrów ogólnego przeznaczenia, które mogą być wykorzystywane pojedynczo lub parami jako rejestry BC, DE, HL. W bloku rejestrów znajdują się ponadto wskaźniki stosu SP, licznik programu PC oraz programowo niedostępna para rejestrów WZ. Blok rejestrów jest połączony dwukierunkowo przez układ we/wy 1 z wewnętrzną szyną danych. Układ we/wy 1 działa jak demultiplekser i wzmacniacz zapisu podczas transmisji informacji w kierunku do bloku rejestrów, natomiast w drugim kierunku pełni rolę multipleksera. Układ we/wy 2 działa jako bufor i wzmacniacz zapisu podczas transmisji z układu inkrementacji/dekrementacji do bloku rejestrów oraz pełni rolę układu buforowego przy transmisji z bloku rejestrów do rejestru adresowego.

Podczas operacji zapisu lub odczytu rejestr adresowy przechowuje adres komórki pamięci lub układu we/wy. Rejestr adresowy służy również do przechowywania zawartości wybranej pary rejestrów podczas inkrementacji lub dekrementacji. Bufor adresowy zapewnia dopasowanie prądowe i napięciowe z układami TTL.



Rys. 3. Schemat blokowy układu MCY 7880

Blok rejestrów, jak również rejestr adresowy sterowane są sygnałami z układu sterowania rejestrami. Układ ten wytwarza sygnały zwrotne do bloku sterowania centralnego i bloku sterowania arytmometrem.

Rejestr instrukcji przechowuje pierwszy bajt instrukcji, zawierający kod operacyjny. Jego zawartość jest dekodowana przez dekodér instrukcji, który składa się z dwóch matryc PLA. Jedna z nich /PLA-2/ dekoduje instrukcje wyróżnione ze względu na arytmometr i sterowanie nim. Druga /PLA-1/ spełnia tę samą rolę względem sterowania rejestrami. Arytmometr wykonuje operacje arytmetyczno-logiczne. Posiada on następujące rejestry:

- rejestr akumulatorowy A, w którym przechowywany jest jeden z operandów przed wykonaniem każdej dwuargumentowej operacji arytmetyczno-logicznej, a następnie wynik operacji,
- rejestr pomocniczy P, pełni rolę rejestru buforowego akumulatora i służy do zapamiętywania stałych przy wykonywaniu niektórych działań,
- rejestr chwilowy T, przechowuje drugi operand. Z tego rejestru poprzez multiplekser MPX może być przesyłana informacja do JAL w postaci prostej lub zagnęwanej,
- rejestr flagowy F, sygnalizuje stan arytmometru po wykonaniu operacji arytmetycznej lub logicznej.

Podstawowe operacje w arytmometrze wykonywane są przez jednostkę arytmetyczno-logiczną JAL. Komunikację zaś pomiędzy nią i rejestrem flagowym F, a wewnętrzną szyną danych zapewnia układ współpracy z IDB. Ponadto arytmometr posiada układ korekcji dziesiętnej, który przy wykonywaniu rozkazu ZDD/DAA/ wpisuje odpowiednie słowo korekcyjne do rejestru pomocniczego.

Przepływem informacji pomiędzy układami arytmometru, a także pomiędzy arytmometrem a wewnętrzną szyną danych steruje układ sterowania arytmometrem SA. Bufor szyny danych umożliwia komunikację pomiędzy wewnętrzną szyną danych IDB, a układami zewnętrznymi. Może on pracować w jednym z trzech trybów zapisu, odczytu lub blokady transmisji. Współpracą poszczególnych bloków mikroprocesora, a także koordynacją jego współdziałania z układami zewnętrznymi steruje blok sterowania centralnego SC. Zawiera on liczniki taktów i cykli, wyznaczające ściśle przedziały czasowe, w których są wykonywane poszczególne mikrooperacje, układ obsługi przerwań, układ synchronizacji oraz układ wytwarzania i transmisji słowa statusu mikroprocesora.

DZIAŁANIE UKŁADU

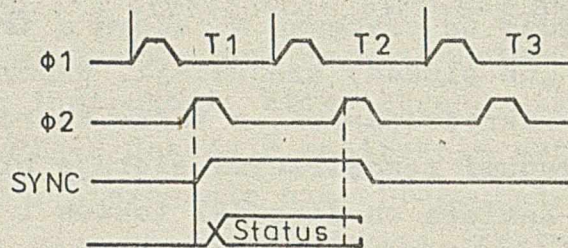
Mikroprocesor jest układem synchronicznym. Czas pobrania i wykonania instrukcji /nazywany dalej cyklem instrukcji/ dzieli się na cykle maszynowe. W czasie każdego cyklu maszynowego następuje przesłanie jednego bajtu pomiędzy mikroprocesorem a pamięcią, lub portem we/wy^{*)}.

Cykle maszynowe składają się z taktów, tj. przedziałów czasowych pomiędzy dwoma kolejnymi narastającymi zboczami sygnału zegarowego $\varnothing 1$.

Na początku każdego cyklu maszynowego, z wyjątkiem drugiego i trzeciego cyklu maszynowego instrukcji DPR /DAD/ dodania zawartości par rejestrów, mikroprocesor wystawia na szynę danych słowo statusu, tzn. informację o tym, jaka operacja nastąpi w danym cyklu maszynowym.

^{*)} Wyjątkiem jest instrukcja STP/HLT/ wejścia w stan zatrzymania oraz instrukcja DPR/DAD/ dodania zawartości pary rejestrów BC, DE, HL lub wskaźnika stosu do zawartości pary rejestrów HL. Druga z w.w. instrukcji składa się z trzech cykli maszynowych, przy czym tylko w pierwszym ma miejsce przesłanie do mikroprocesora /przesyłany jest kod instrukcji/. W cyklach drugim i trzecim wykonywane są wyłącznie operacje wewnętrzne.

Jednocześnie sygnał wyjściowy SYNC przyjmuje wartość 1 /rys.4/. Słowo statusu powinno być odczytane z szyny danych podczas koincydencji sygnałów $\phi 1$ i SYNC.



Rys.4. Przebiegi czasowe dla sygnału SYNC i słowa statusu
Wszystkie możliwe słowa statusu zamieszczone są w tabeli.

Słowa statusu układu MCY 7880

Lp.	Słowo statusu D7 ... DO	Operacja, która nastąpi w danym cyklu maszynowym
1	10100010	Pobranie kodu operacyjnego instrukcji
2	10000010	Odczyt z pamięci
3	00000000	Zapis do pamięci
4	10000110	Odczyt ze stosu
5	00000100	Zapis do stosu
6	01000010	Odczyt z układu wejście/wyjście
7	00010000	Zapis do układu wejście/wyjście
8	00100011	Przerwanie
9	10001010	Wejście w stan zatrzymania
10	00101011	Przerwanie-wyjście ze stanu zatrzymania

W systemie mikroprocesorowym słowa statusu są zapamiętywane w układzie kontrolera UCY 74S428. Układ ten generuje na ich podstawie sygnały zapisu do pamięci, odczytu z pamięci, zapisu do układu we/wy, odczytu z układu we/wy oraz sygnał potwierdzenia przerwania.

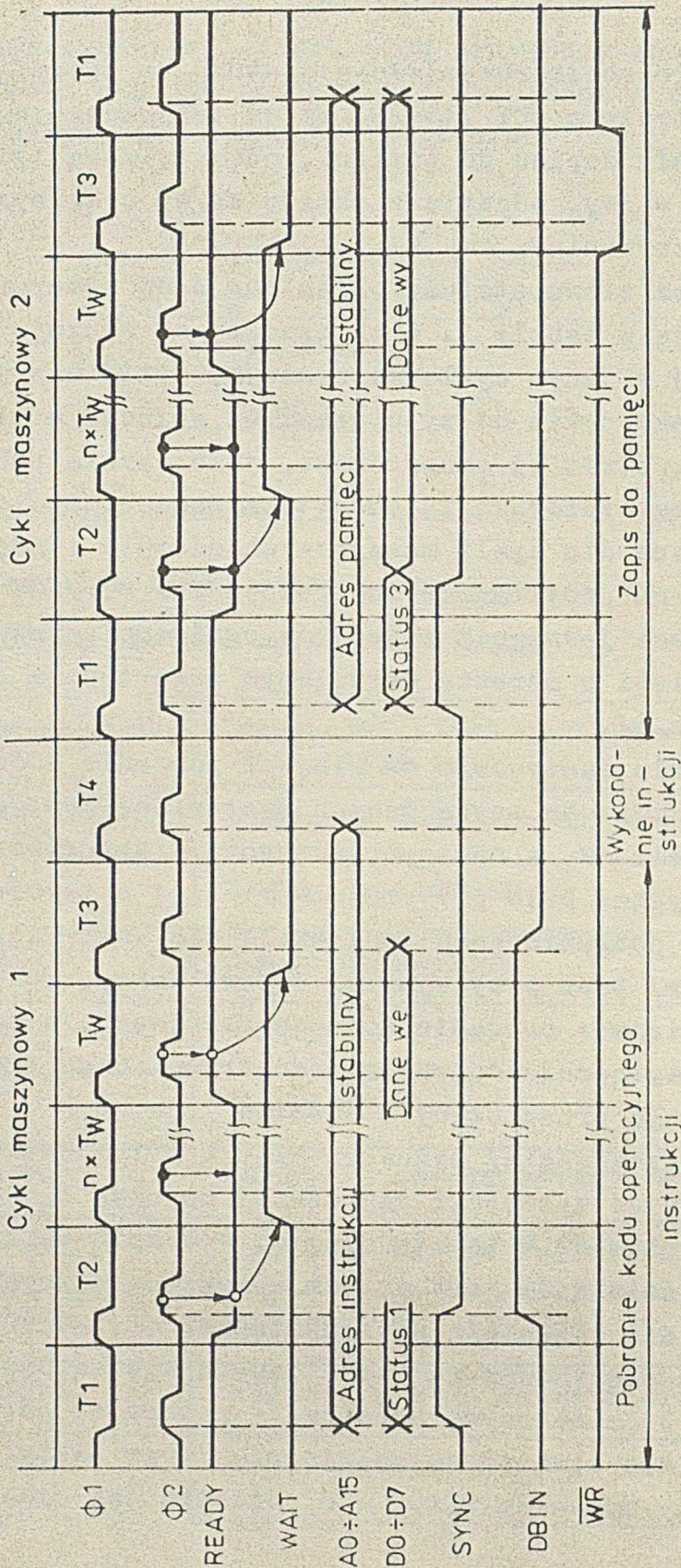
W dalszej części słowa statusu określane będą numerem, pod którym występują w tabeli 1. W zależności od rodzaju operacji przeprowadzanej w danym cyklu maszynowym, cykle maszynowe będą określane jako cykle odczytu, zapisu, wejścia w stan zatrzymania lub bardziej szczegółowo, jako cykle pobrania kodu operacyjnego instrukcji, odczytu z pamięci itd.

Sekwencje zdarzeń dla cykli maszynowych odczytu i zapisu zostaną omówione na przykładzie instrukcji PRZ M,r/MOV M,r/ zapisu zawartości jednego z rejestrów ogólnego przeznaczenia do komórki pamięci o adresie określonym zawartością pary rejestrów HL /rys.5/.

W pierwszym cyklu maszynowym ma miejsce pobranie kodu operacyjnego instrukcji. Na szynę danych wystawione jest słowo statusu 1 /Tabela 1/, a na szynę adresową - zawartość licznika programu. Sygnał DBIN przyjmuje wartość 1 w takcie T2.

Cykl maszynowy pobrania kodu instrukcji dla instrukcji PRZ M,r/MOV M,r/ trwa przynajmniej cztery takty. Podczas taktów T1-T3 ma miejsce pobranie kodu operacyjnego, w takcie T4 wykonywane są wyłącznie operacje wewnętrzne. Czas trwania tego cyklu może być przedłużony o dowolną liczbę taktów T_n /tzw. taktów oczekiwania/ sygnałem READY. W tym celu sygnał READY musi przyjąć wartość 0 najpóźniej na czas t_{RS} przed opadającym zboczem $\emptyset 2$ w takcie drugim. W czasie taktów oczekiwania na wyjściu WAIT jest poziom wysoki, a wyprowadzenia AO-A15 i DBIN nie zmieniają swoich stanów.

W drugim cyklu maszynowym na szynę danych wystawione jest słowo statusu 3, a na szynę adresową - zawartość pary rejestrów HL. Cykl ten trwa co najmniej przez trzy takty i może być przedłużony sygnałem READY = 0, podobnie jak cykl pierwszy.



Rys. 5. Przebieg1 czasowe sygnałów podczas wykonywania instrukcji PRZ M, r/ MOV M, r/

Sygnal \overline{WR} przybiera wartość 0 w takcie T3.

Dla niektórych instrukcji cykle maszynowe pobrania kodu operacyjnego instrukcji są o jeden takt dłuższe. W przypadkach tych takt T5, podobnie jak T4, jest taktem wykonywania instrukcji.

Cykle maszynowe odczytu z pamięci, ze stosu lub układu we/wy składają się z taktów T1, T2, T_w /w przypadku wystąpienia stanu wstrzymania/ i T3. Przebiegają one tak, jak operacja pobrania kodu operacyjnego instrukcji, z tym że w każdym cyklu maszynowym wystawione jest odpowiednie słowo statusu.

Cykle maszynowe zapisu do pamięci stosu i układu we/wy różnią się jedynie wystawianymi słowami statusu. Wyjątkiem jest piąty cykl maszynowy cyklu instrukcji WHL/XTHL/ zamiany zawartości pary rejestrów HL i wierzchołka stosu /cykl zapisu do stosu/, który ma dwa dodatkowe takty T4 i T5 wykorzystywane do operacji wewnętrznych.

Żądania przerwania są zgłaszane przez wystawienie do poziomu logicznej "1" wejścia INT. Przerwania są maskowane wewnętrznym przerzutnikiem, którego stan podawany jest na wyjście INTE /stan wysoki na wyjściu INTE oznacza możliwość przerwania/. Przerzutnik maski jest ustawiany i zerowany programowo. Po wejściu układu w stan początkowy, jak również po przyjęciu zgłoszenia przerwania, przerzutnik maski przyjmuje wartość 0. Warunkiem przyjęcia zgłoszenia przerwania jest $INT \cdot INTE = 1$ najpóźniej na czas t_{IS} przed opadającym zboczem sygnału ϕ_2 w ostatnim takcie cyklu instrukcji. Po przyjęciu zgłoszenia przerwania mikroprocesor wystawia na szynę danych słowo statusu 8 i nie inkrementuje zawartości licznika programu. Po odebraniu słowa statusu 8 układy zewnętrzne muszą podać na szynę danych kod operacyjny instrukcji skoku do podprogramu KZS/CALL/ lub RST. Z punktu widzenia mikroprocesora sekwencje zdarzeń podczas cyklu przerwania i cyklu pobrania kodu operacyjnego różnią się jedynie wystawianym słowem statusu. W przypadku trybajtowej instrukcji KZS/CALL/ w drugim i trzecim cyklu maszynowym wystawiane jest słowo statusu 2.

Po przyjęciu zgłoszenia przerwania przerzutnik maski jest zerowany najbliższym impulsem $\emptyset 2 / \emptyset 2$ w takcie 1/.

Po zrealizowaniu programu mikroprocesor może zostać wprowadzony w stan zatrzymania. Do tego celu służy instrukcja STP /HLT/ /rys.6/.

Po wczytaniu kodu operacyjnego, instrukcji STP/HLT/ wartość sygnału READY nie ma wpływu na przebieg jej wykonywania.

Przed wejściem w stan zatrzymania mikroprocesor wystawia na szynę danych słowo statusu 9. Stan zatrzymania może trwać dowolnie długo, a wyjście z niego może odbyć się poprzez:

- wprowadzenie układu w stan początkowy /patrz niżej/;
- wprowadzenie układu w stan wstrzymania /patrz niżej/;
- po zakończeniu stanu wstrzymania układ powraca do stanu zatrzymania;
- przerwanie /jeżeli przerzutnik maski jest ustawiony/.

Po przyjęciu zgłoszenia przerwania podczas stanu zatrzymania na szynę danych wystawione jest słowo statusu 10.

Przejsie do podprogramu obsługi przerwania przebiega podobnie jak przy przerwaniu poza stanem zatrzymania.

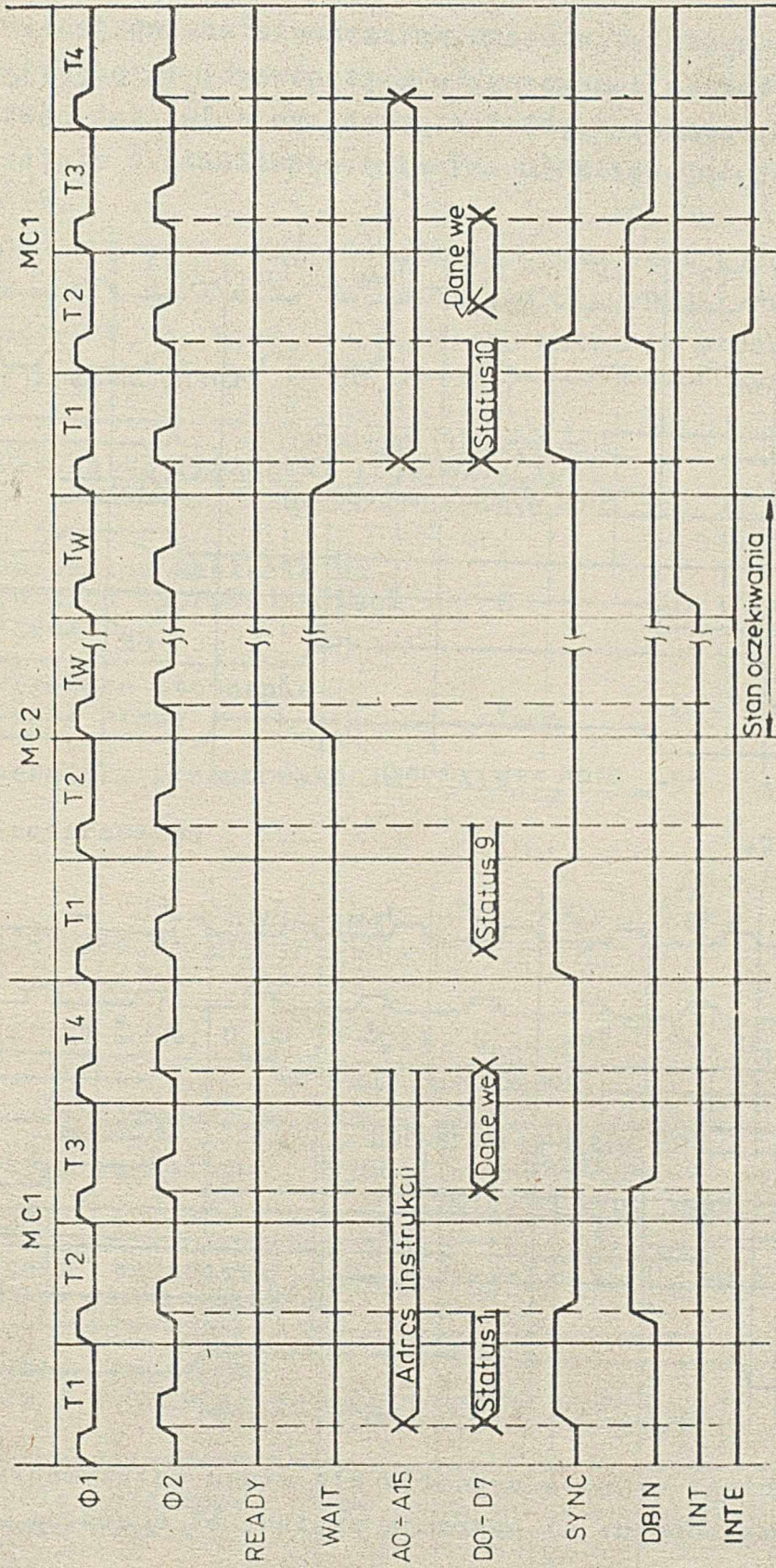
W celu przeprowadzenia transmisji danych w trybie DMA należy wprowadzić układ MCY 7880 w stan wstrzymania sygnałem

HOLD = 1. Jeżeli sygnał HOLD = 1 pojawi się w czasie cyklu maszynowego zapisu lub odczytu, to operacja zapisu /odczytu przebiega normalnie, a w stan wstrzymania mikroprocesor

wchodzi po jej zakończeniu /rys.7^{x)}. W stanie wstrzymania wyprowadzenia AO-A15 są w stanie wysokiej impedancji, a na wyjściu HLDA utrzymywany jest poziom logicznej jedynki.

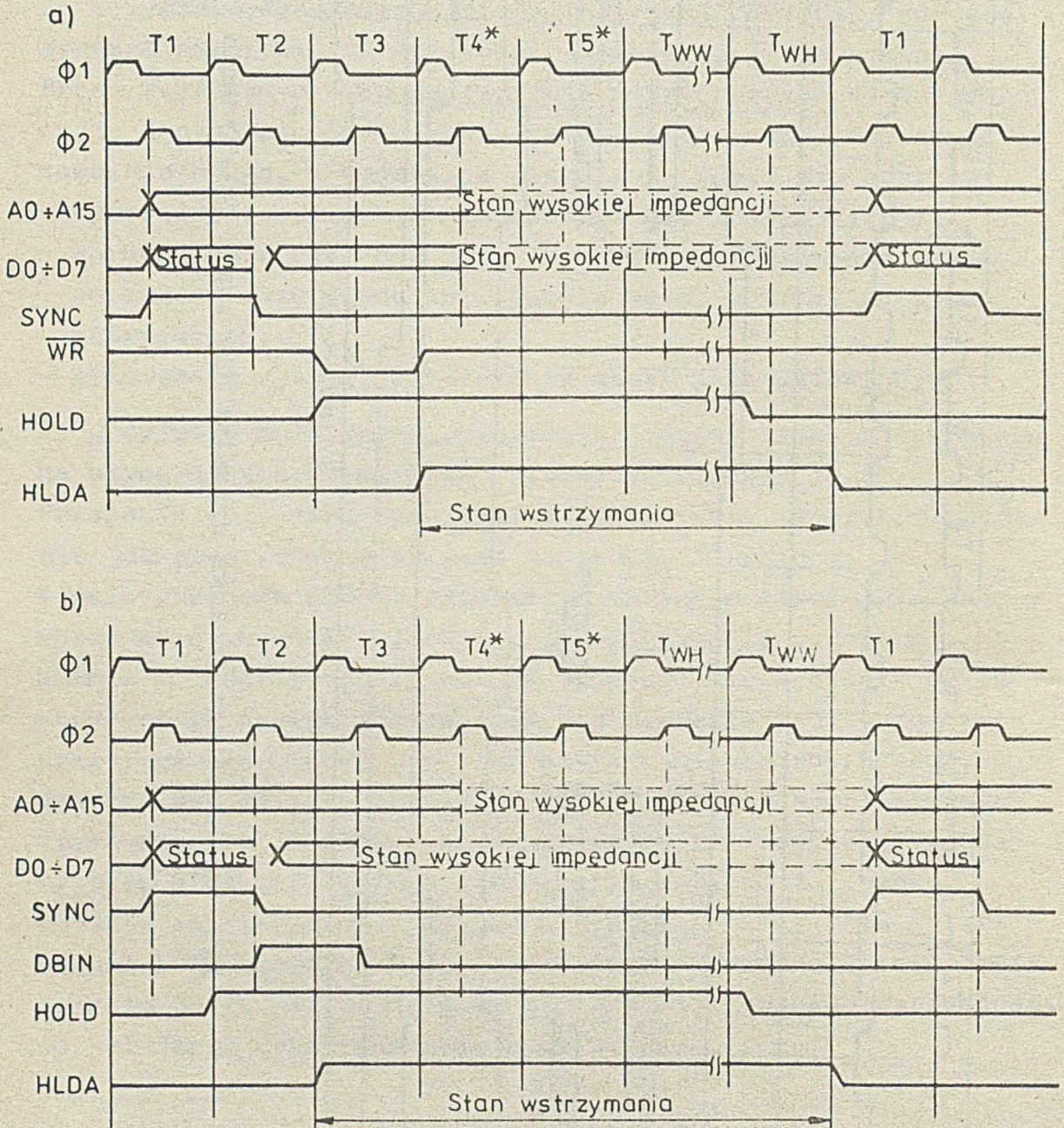
Wyjście ze stanu wstrzymania następuje po przyjęciu przez sygnał HOLD wartości "0". Jeżeli stan wstrzymania trwa dostatecznie długo, to po wyjściu z niego mikroprocesor przechodzi do kolejnego cyklu maszynowego.

*) Zainicjowanie stanu wstrzymania podczas drugiego i trzeciego cyklu maszynowego instrukcji DPR/DAD/ przebiega tak, jak podczas cyklu maszynowego zapisu.



Rys. 6. Przebiegi czasowe sygnałów dla stanu zatrzymania

Jeżeli natomiast wyjście ze stanu wstrzymania następuje przed normalnym końcem danego cyklu maszynowego, to po wyjściu ze stanu wstrzymania cykl ten przebiega dalej tak, jak przebiegałby, gdyby sygnał HOLD = 1 nie pojawił się.



Rys. 7. Przebiegi czasowe sygnałów dla stanu wstrzymania poprzedzonego: a) operacją zapisu, b) operacją odczytu

Po włączeniu zasilania mikroprocesor należy wprowadzić w stan początkowy. W tym celu sygnał RESET powinien przyjąć wartość 1 na czas przynajmniej trzech taktów. Mikroprocesor pozostaje w stanie początkowym dopóki sygnał RESET jest równy "1".

W stanie początkowym licznik programu jest wyzerowany, INTE = 0 i HLDA = 0. Działaniu sygnału RESET nie podlegają rejestry W, Z, B, C, D, E, H, L, wskaźnik stosu, rejestr flagowy i akumulator.

DOPUSZCZALNE PARAMETRY EKSPLOATACYJNE

Dopuszczalne napięcie na wszystkich wyprowadzeniach względem U_{BB}	U	-0,3 ÷ +20	V
Temperatura otoczenia w czasie pracy	t_{amb}	0 ÷ 70	$^{\circ}C$
Temperatura przechowywania	t_{stg}	-55 ÷ +125	$^{\circ}C$
Moc rozpraszana	P_{tot}	1,5	W

ELEKTRYCZNE PARAMETRY CHARAKTERYSTYCZNE

$/U_{DD} = 12V \pm 5\%$, $U_{CC} = 5V \pm 5\%$, $U_{BB} = -5V \pm 5\%$, $U_{SS} = 0V$, $t_{amb} = 0+70^{\circ}C/$

Nazwa parametru	Symbol	Wartość			Jedn.	Warunki pomiaru
		min.	typ.	max.		
1	2	3	4	5	6	7
Napięcie wejściowe zegara w stanie niskim	U_{ILC}	$U_{SS}-1$	-	$U_{SS}+0,8$	V	
Napięcie wejściowe zegara w stanie wysokim	U_{IHC}	9,0	-	$U_{DD}+1$	V	
Napięcie wejściowe w stanie niskim	U_{IL}	$U_{SS}-1$	-	$U_{SS}+0,8$	V	

PARAMETRY CHARAKTERYSTYCZNE c.d.

1	2	3	4	5	6	7
Napięcie wejściowe w stanie wysokim	U_{IH}	3,3	-	$U_{CC}+1$	V	
Napięcie wyjściowe w stanie niskim	U_{OL}	-	-	0,45	V	$I_{OL}=1,9 \text{ mA}$
Napięcie wyjściowe w stanie wysokim	U_{OH}	3,7	-	-	V	$I_{OH}=-150 \text{ } \mu\text{A}$
Średni prąd zasilania / U_{DD} /	I_{DD}		40	70	mA	
Średni prąd zasilania / U_{CC} /	I_{CC}		60	80	mA	$t_{CY}=0,48 \text{ } \mu\text{s}$
Średni prąd zasilania / U_{BB} /	I_{BB}		0,01	1	mA	
Prąd upływności wejść	I_{II}			± 10	$\text{ } \mu\text{A}$	$U_{SS} < U_{IN} < U_{CC}$
Prąd upływności wejść zegarowych	I_{IC}			± 10	$\text{ } \mu\text{A}$	$U_{SS} < U_{CL} < U_{DD}$
Upływność wyprowadzeń szyny danych pracującej w trybie wejściowym	I_{LD}			-100	$\text{ } \mu\text{A}$	$U_{SS} < U_{IN} < U_{SS} + 0,8\text{V}$
				-2,0	mA	$U_{SS} + 0,8\text{V} < U_{IN} < U_{CC}$
Upływność wyprowadzeń szyny danych i szyny adresowej w stanie wysokiej impedancji	I_{LZ}			+10	$\text{ } \mu\text{A}$	$U_{ADR/DANE} = U_{CC}$
				-100	$\text{ } \mu\text{A}$	$U_{ADR/DANE} = U_{SS} + 0,45\text{V}$
Pojemność wejść zegarowych	C_{IC}		17	25	pF	$f_C = 1 \text{ MHz}$
Pojemność wejść	C_I		6	10	pF	Wyprowadzenia nie mierzone podłączone do U_{SS}
Pojemność wyjść	C_O		10	20	pF	
Czas cyklu zegarowego/ ϕ	t_{cp}	0,48		2,0	$\text{ } \mu\text{s}$	
Czas narastania i opadania sygnału zegarowego	t_{rc} t_{fc}	0		50	ns	

PARAMETRY CHARAKTERYSTYCZNE c.d.

1	2	3	4	5	6	7
Czas trwania impulsu $\emptyset 1=1$	$t_{w\emptyset 1}$	60			ns	
Czas trwania impulsu $\emptyset 2=1$	$t_{w\emptyset 2}$	220			ns	
Czas utrzymywania sygnału $\emptyset 2=0$ po zmianie stanu sygnału $\emptyset 1$ na 0	t_{D1}	0			ns	
Czas utrzymywania sygnału $\emptyset 1=0$ po zmianie stanu sygnału $\emptyset 2$ na 0	t_{D2}	70			ns	
Czas utrzymywania sygnału $\emptyset 2=0$ po zmianie stanu sygnału $\emptyset 1$ na 1	t_{D3}	80			ns	
Opóźnienie adresu względem $\emptyset 2=1$ /2/ /4/	t_{DA}			200	ns	$C_L = 100 \text{ pF}$
Opóźnienie danych względem $\emptyset 2=1$ /2/ /4/	t_{DD}			220	ns	
Opóźnienie sygnałów SYNC, WR, WAIT, HLDA, względem $\emptyset 1 = 1$ lub $\emptyset 2 = 1$ /2/ /4/	t_{DC}			120	ns	$C_L = 50 \text{ pF}$
Opóźnienie sygnału DBIN względem $\emptyset 2 = 1$ /2/ /4/	t_{DF}	25		140	ns	
Opóźnienie przejścia szyny danych w tryb wejściowy względem $\emptyset 2 = 1$	t_{DI}			t_{DF}	ns	
Czas podtrzymywania danych podczas DBIN=1 i $\emptyset 1 = 1$	t_{DS1}	30			ns	
Czas podtrzymywania danych podczas DBIN=1 i $\emptyset 2 = 0$	t_{DS2}	150			ns	
Czas podtrzymywania danych podczas DBIN=1 i $\emptyset 2 = 1$ /1/	t_{DH}	/1/			ns	

PARAMETRY CHARAKTERYSTYCZNE c.d.

1	2	3	4	5	6	7
Opóźnienie zmiany stanu sygnału INTE względem $\emptyset 2 = 1$ /2/ /4/	t_{IE}			200	ns	$C_L = 50$ pF
Czas podtrzymywania sygnału READY podczas $\emptyset 2 = 1$	t_{RS}	120			ns	
Czas podtrzymywania sygnału HOLD podczas $\emptyset 2 = 0$	t_{HS}	140			ns	
Czas podtrzymywania sygnału INT podczas $\emptyset 2 = 1$	t_{IS}	120			ns	
Czas podtrzymywania sygnałów READY i INT podczas $\emptyset 2 = 0$ i HOLD podczas $\emptyset 2 = 1$	t_H	0			ns	
Opóźnienie wejścia szyny adresowej i szyny danych w stan wysokiej impedancji względem $\emptyset 2 = 1$ dla stanu wstrzymania	t_{FD}			120	ns	
Czas podtrzymywania adresu przed zmianą \overline{WR} na 0 /2/ /4/	t_{AW}	/5/				
Czas podtrzymywania danych przed zmianą \overline{WR} na 0 /2/ /4/	t_{DW}	/6/				
Czas podtrzymywania danych po zmianie \overline{WR} na 1 /2/ /4/	t_{WD}	/7/				$C_L = 100$ pF dla wyprowadzeń szyny danych i szyny adresowej;
Czas podtrzymywania adresu po zmianie \overline{WR} na 1 /2/ /4/	t_{WA}	/7/				$C_L = 50$ pF dla wyjść \overline{WR} HLDA i DBIN.
Opóźnienie wejścia szyny danych i szyny adresowej w stan wysokiej impedancji względem HLDA = 1 /2/ /4/	t_{HF}	/8/				

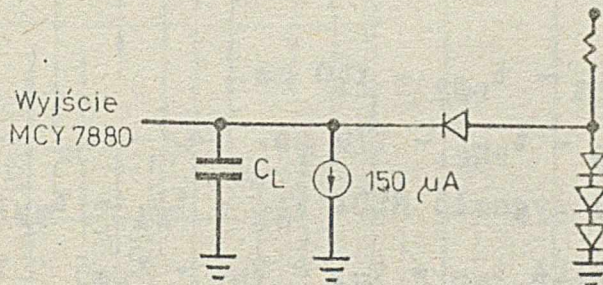
PARAMETRY CHARAKTERYSTYCZNE c.d.

1	2	3	4	5	6	7
Opóźnienie wejścia szyny danych i szyny adresowej w stan wysokiej impedancji względem $\overline{WR}=1$ /2//4/	t_{WF}	/9/				warunek ze str. 18
Czas podtrzymywania adresu po $\overline{DBIN} = 1$ podczas $\overline{HLDA}=1$ /2//4/	t_{AH}	-20			ns	

Wyjaśnienia i uwagi do tabeli PARAMETRY CHARAKTERYSTYCZNE /pkt. 1-9/ i rysunku 10 /pkt. 10-13/:

1. Dane wejściowe powinny pojawić się podczas sygnału \overline{DBW}

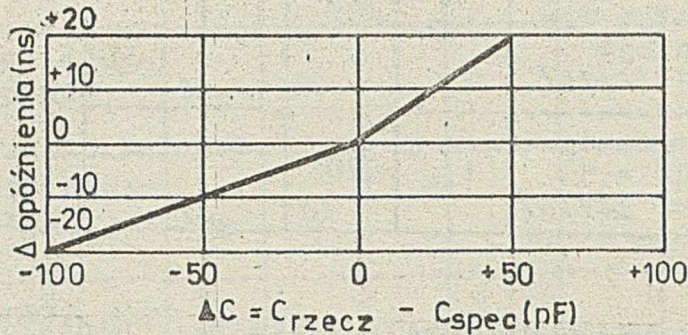
2.
$$t_{DH} = \begin{cases} 50_{ns} ; \text{ dla } t_{DF} \geq 50_{ns} \\ \text{lub} \\ t_{DF} ; \text{ dla } t_{DF} < 50_{ns} \end{cases}$$



Rys. 8. Układ obciążający

3. $t_{CY} = t_{D3} + t_{r\phi 2} + t_{\phi 2} + t_{r\phi 2} + t_{D2} + t_{r\phi 1} \geq 480_{ns}$.

4. Typowa zmiana opóźnienia sygnałów wyjściowych w funkcji zmiany pojemności przybierze postać, jak na rysunku 9.



Rys. 9. Typowa zmiana opóźnienia sygnałów wyjściowych w funkcji zmiany pojemności obciążenia

Zmiana ta dotyczy łączenia układu MCY 7880 z układami o $U_{IH} = 3,3$ V:

- a/ maksymalny czas narastania sygnału wyjściowego od 0,8 do 3,3 V jest równy 100 ns przy $C_L = C_{spec}$,
- b/ opóźnienie sygnału wyjściowego /mierzone do chwili osiągnięcia 3,0 V/ jest równe SPEC + 60 ns przy $C_L = C_{spec}$,
- c/ jeżeli $C_L > C_{spec}$ należy dodać 0,6 ns/pF; jeżeli $C_L < C_{spec}$ należy odjąć od zmodyfikowanego opóźnienia 0,3 ns/pF.

5. $t_{AW} = 2t_{CY} - t_{D3} - t_{r02} - 140$ ns.

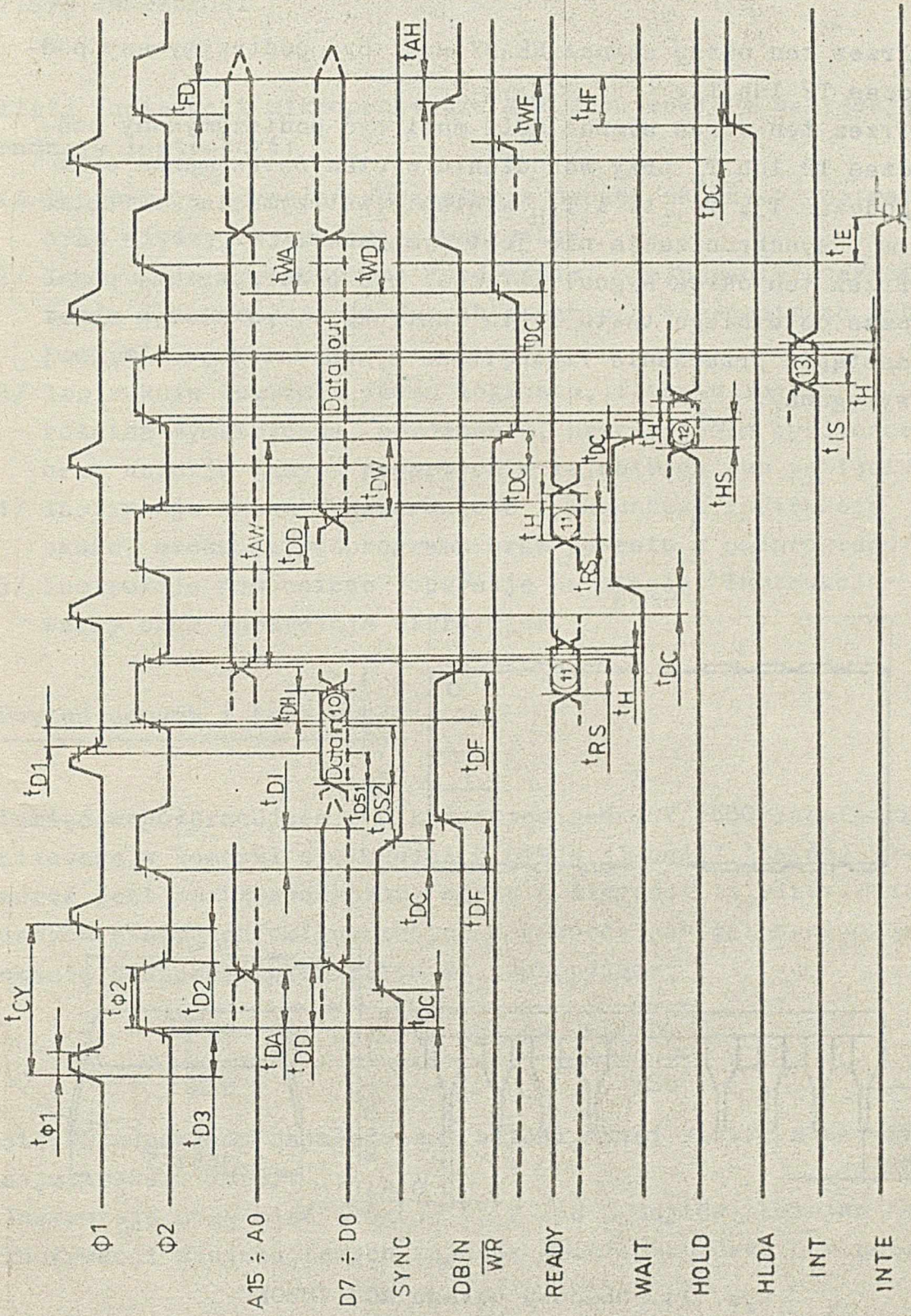
6. $t_{DW} = t_{CY} - t_{D3} - t_{r02} - 170$ ns.

7. Podczas trwania sygnału HLDA $t_{WD} = t_{WA} = t_{WF}$; podczas braku sygnału HLDA $t_{WD} = t_{WA} = t_{D3} + t_{r02} + 10$ ns.

8. $t_{HF} = t_{D3} + t_{r02} - 50$ ns.

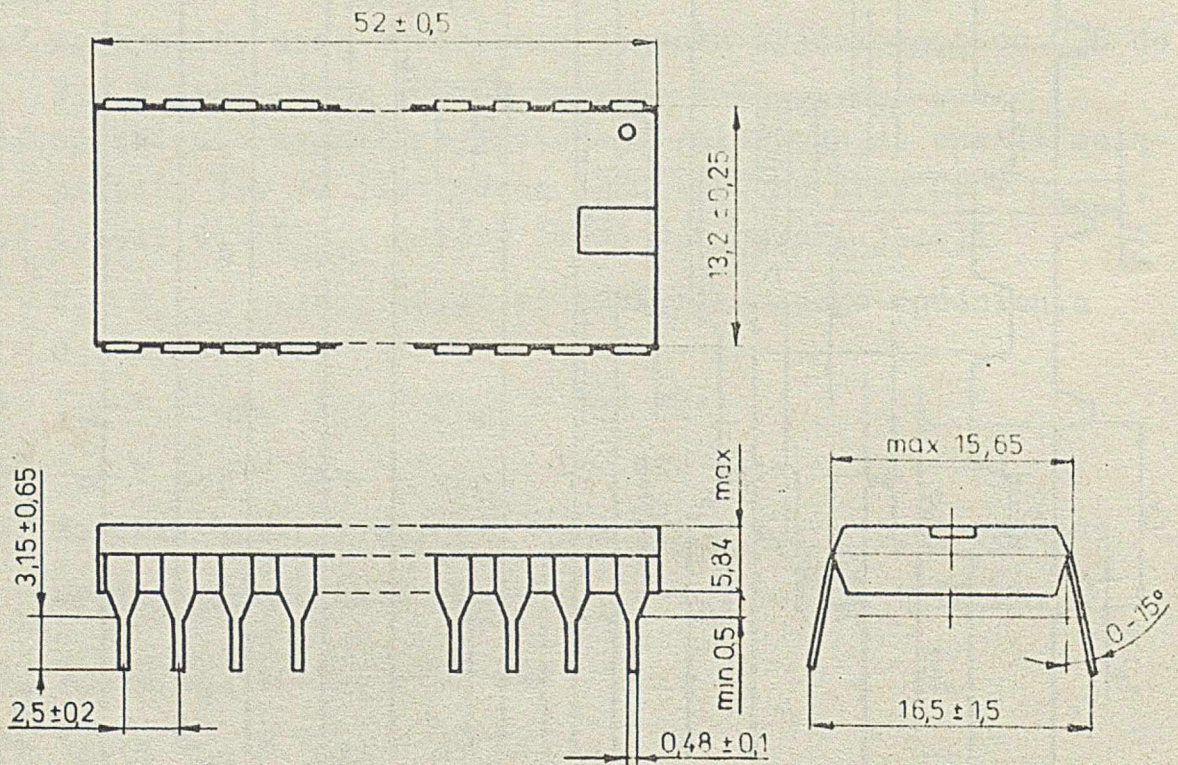
9. $t_{WF} = t_{D3} + t_{r02} - 10$ ns.

10. Przez ten okres dane muszą być podtrzymywane podczas DBIN i T3.



Rys. 10. Zależności czasowe sygnałów w układzie MCY 7880. Diagram ten nie reprezentuje żadnego konkretnego cyklu maszynowego

11. Przez ten okres sygnał READY musi być podtrzymywany podczas T_2 lub T_W .
12. Przez ten okres sygnał HOLD musi być podtrzymywany podczas T_2 lub T_W przy wchodzeniu w stan wstrzymania oraz podczas T_3 , T_4^* , t_5^* i T_{WH} w stanie wstrzymania /zewnętrzna synchronizacja nie jest wymagana/.
13. Przez ten okres sygnał INT musi być podtrzymywany podczas ostatniego taktu cyklu instrukcji, po którym ma nastąpić przerwanie /zewnętrzna synchronizacja nie jest wymagana/.



Rys. 11. Obudowa układu MCY 7880N

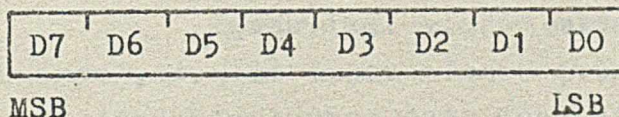
LISTA INSTRUKCJI

Lista instrukcji mikroprocesora MCY 7880 zawiera następujące rodzaje instrukcji:

- 1/ Instrukcje przesłań /przesłania danych między rejestrami oraz między rejestrami a pamięcią/,
- 2/ Instrukcje arytmetyczne /dodawanie, odejmowanie, dekrementacja i inkrementacja danych zawartych w rejestrach lub pamięci/,
- 3/ Instrukcje logiczne /suma logiczna, iloczyn logiczny, różnica symetryczna, porównanie, przesunięcie cykliczne oraz negacja danych zawartych w rejestrach lub pamięci/,
- 4/ Instrukcje skoku /bezwarunkowe i warunkowe instrukcje skoku, skoku do podprogramu oraz powrotu z podprogramu/,
- 5/ Instrukcje pomocnicze /operacje na stosie, instrukcje we/wy oraz instrukcje sterujące/.

Postać danych i instrukcji

Pamięć współpracująca z mikroprocesorem MCY 7880 jest zorganizowana w komórki o długości 8 bitów /1 bajt/. Każdej komórce jest przyporządkowany adres o długości 16 bitów. Postać danych binarnych umieszczonych w komórce pamięci oraz oznaczenia poszczególnych bitów są następujące:



Bit D0 odpowiada najmłodszemu bitowi danej /LSB/, a bit D7 najstarszemu /MSB/.

Instrukcje mogą mieć długość 1, 2 lub 3 bajtów. Kolejne bajty instrukcji wielobajtowych zajmują kolejne komórki pamięci.

Instrukcje skoku mają następujące sposoby określania adresu instrukcji, do której ma być wykonany skok:

- 1/ bezpośrednio: drugi i trzeci bajt instrukcji zawierają adres /z wyjątkiem instrukcji RST, drugi bajt zawiera młodsze bity adresu, a trzeci starsze/,
- 2/ pośrednio: instrukcja określa parę rejestrów, które zawierają adres /pierwszy rejestr pary zawiera starsze bity adresu, a drugi - młodsze/.

Instrukcja RST określa adres za pomocą liczby restartowej n /opis instrukcji str. 29/.

Rejestr flagowy

Podczas wykonywania pewnych instrukcji są modyfikowane poszczególne bity rejestru flagowego. W opisie każdej instrukcji podano, które z bitów rejestru flagowego są modyfikowane. Jeżeli nie zaznaczono inaczej, sposób modyfikacji i poszczególnych bitów jest następujący:

- Bit zera Z: Jeżeli wynik wykonania instrukcji przybiera wartość 0, wtedy Z jest ustawiany; w przeciwnym przypadku Z jest zerowany.
- Bit znaku S: Jeżeli najbardziej znaczący bit wyniku operacji ma wartość 1, wtedy S jest ustawiany; w przeciwnym przypadku S jest zerowany.
- Bit parzystości P: Jeżeli suma modulo 2 wszystkich bitów wyniku operacji jest równa 0, wtedy P jest ustawiany; w przeciwnym przypadku P jest zerowany.
- Bit przeniesienia CY: Jeżeli podczas wykonywania dodawania ma miejsce przeniesienie z najbardziej znaczącej pozycji lub podczas wykonywania odejmowania /porównywania/, pozycja dla najbardziej znaczącej pozycji

cji - CY jest ustawiany; w przeciwnym przypadku CY jest zerowany.

Bit przeniesienia
pomocniczego AC:

Jeżeli podczas wykonywania dodawania ma miejsce przeniesienie z pozycji czwartej lub podczas wykonywania odejmowania /porównania/ ma miejsce pożyczka dla pozycji czwartej - AC jest ustawiany; w przeciwnym przypadku AC jest zerowany.

Oznaczenia

Przy opisie instrukcji stosowane są następujące oznaczenia:

akumulator - rejestr A,
adr - adres 16-bitowy,
dane - dane 8-bitowe,
dane 16 - dane 16-bitowe,
bajt 2 - drugi bajt instrukcji,
bajt 3 - trzeci bajt instrukcji,
port - 8-bitowy rejestr buforowany układu we/wy,
r, r1, r2 - jeden z rejestrów A,B,C,D,E,H,L,
DDD, SSS - 3-bitowe oznaczenie jednego z rejestrów A,B,C,
D,E,H,L /DDD - rejestr, do którego dane są wpisywane; SSS - rejestr, z którego dane są pobierane/:

DDD	lub	SSS	rejestr
		111	A
		000	B
		001	C
		010	D
		011	E
		100	H
		101	L

rp - jedna z par rejestrów:
B - określa parę BC /B jest starszym rejestrem

- pary, a C młodszym/,
D - określa parę DE /D jest starszym rejestrem pary, a E młodszym/,
H - określa parę HL /H jest starszym rejestrem pary, a L młodszym/,
SP - 16-bitowy wskaźnik stosu,
RP - 2-bitowe oznaczenie jednej z par rejestrów
- | RP | para rejestrów |
|----|----------------|
| 00 | BC |
| 01 | DE |
| 10 | HL |
| 11 | SP |
- rh - pierwszy /starszy/ rejestr pary,
rl - drugi /młodszy/ rejestr pary,
PC - 16-bitowy licznik rozkazów,
PCH - 8 starszych bitów licznika rozkazów,
PCL - 8 młodszych bitów licznika rozkazów,
SP - 16-bitowy wskaźnik stosu,
SPH - 8 starszych bitów wskaźnika stosu,
SPL - 8 młodszych bitów wskaźnika stosu,
 r_m - m-ty bit rejestru r / $M = 0, 1 \dots 7$, $r_7 = \text{MSB}$,
 $r_0 = \text{LSB}$ /,
Z,S,P,CY,AC - bity rejestru flagowego:
Z - zera,
S - znaku,
P - parzystości,
CY - przeniesienia,
AC - przeniesienia pomocniczego,
[] - zawartość komórki pamięci,
() - zawartość rejestru,
← - jest przepisywane,
 \wedge - iloczyn logiczny /AND/,
 \oplus - różnica symetryczna /EXCLUSIVE OR/,
 \vee - suma logiczna /OR/,
+ - suma,

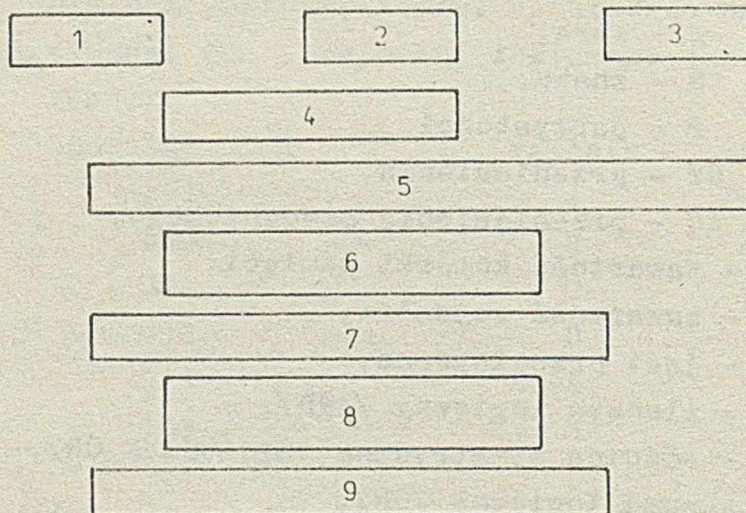
- różnica /w kodzie uzupełnienia do 2/,
- * iloczyn,
- wymień zawartość,
- negacja logiczna np. \bar{A} ,
- n liczba restartowa /n = 0,1,...7/,
- NNN liczba restartowa w kodzie binarnym
- CCC 3-bitowe oznaczenie warunku wykonania instrukcji. Warunkiem wykonania instrukcji może być stan jednego z bitów rejestru flagowego:

CCC	warunek
000	Z=0
001	Z=1
010	CY=0
011	CY=1
100	P=0
101	P=1
110	S=0
111	S=1

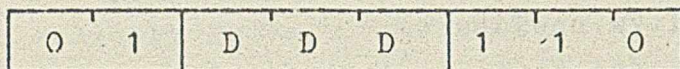
- MCI - i-ty cykl maszynowy, $i = 1, 2, \dots, 5$,
- Tj - j-ty takt cyklu maszynowego, $j = 1, 2, \dots, 5$,
- Tz - takt w stanie zatrzymania,
- (k) - liczba porządkowa słowa statusu wg tabeli 1 /str.8/. $k = 1, 2, \dots, 10$.

Sposób opisu

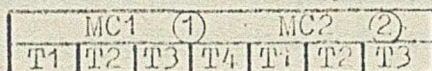
Każda instrukcja jest opisana według następującego schematu:



Zawartość komórki pamięci o adresie umieszczonym w parze rejestrów HL jest przepisywana do rejestru r.

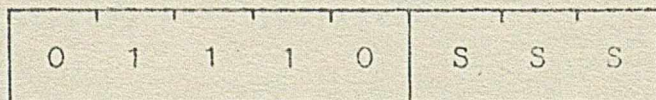


Modyfikowane bity rejestru flagowego: -

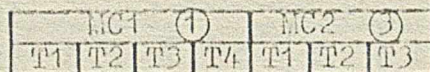


PRZ M, r /MOV M, r/ /przepisz rejestr
[/H//L/] ← /r/ do pamięci/

Zawartość rejestru r jest przepisywana do komórki pamięci o adresie umieszczonym w parze rejestrów HL.

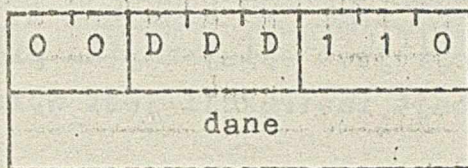


Modyfikowane bity rejestru flagowego: -

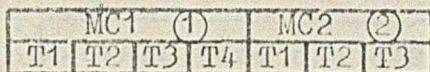


PRD r, dane /MVI r, data/ /przepisz dane do
/r/ ← /bajt 2/ rejestru/

Drugi bajt instrukcji jest przepisywany do rejestru r.

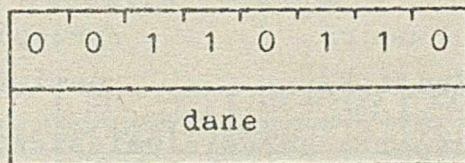


Modyfikowane bity rejestru flagowego: -

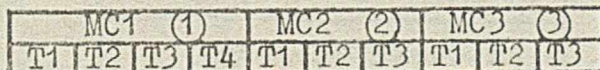


PRD M, dane /MVI M, data/ /przepisz dane do
[/H//L/] /bajt 2/ pamięci/

Drugi bajt instrukcji jest przepisywany do komórki pamięci o adresie umieszczonym w parze rejestrów HL.



Modyfikowane bity rejestru flagowego: -



Modyfikowane bity rejestru flagowego: -

MC1 (1)				MC2 (2)			MC3 (2)			MC4 (2)		
T1	T2	T3	T4	T1	T2	T3	T1	T2	T3	T1	T2	T3

PAP adr /STA addr/
 [/bajt 3//bajt 2/] ← /A/

/przepisz bezpo-
 średnio akumula-
 tor do pamięci/

Zawartość akumulatora jest przepisywana do komórki pamięci o adresie umieszczonym w drugim i trzecim bajcie instrukcji.

0	0	1	1	0	0	1	0
młodsze bity adr							
starsze bity adr							

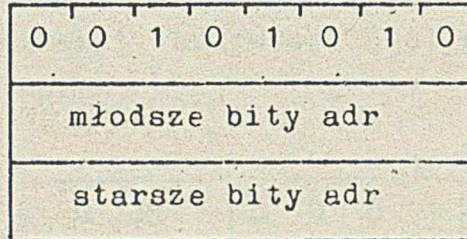
Modyfikowane bity rejestru flagowego: -

MC1 (1)				MC2 (2)			MC3 (2)			MC4 (3)		
T1	T2	T3	T4	T1	T2	T3	T1	T2	T3	T1	T2	T3

PPR adr /LHLD addr/ /przepisz pamięć do pary
 /L/ ← [/bajt 3//bajt 2/] rejestrów/
 /H/ ← [/bajt 3//bajt 2/ + 1]

Zawartość komórki pamięci o adresie umieszczonym w drugim

i trzecim bajcie instrukcji jest przepisywana do rejestru L. Zawartość następnej komórki pamięci jest przepisywana do rejestru H.

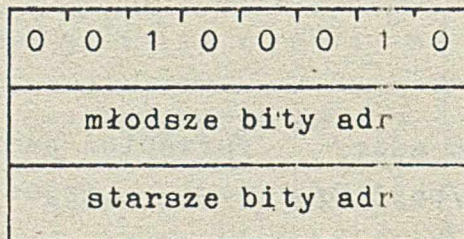


Modyfikowane bity rejestru flagowego: -



PRP adr /SHLD addr/ /przepisz bezpośrednio parę
[/bajt 3//bajt 2/] ← /L, rejestrów do pamięci/
[/bajt 3//bajt 2/+ 1] ← /R/

Zawartość rejestru L jest przepisywana do komórki pamięci o adresie umieszczonym w drugim i trzecim bajcie instrukcji. Zawartość rejestru H jest przepisywana do następnej komórki pamięci.



Modyfikowane bity rejestru flagowego: -

MC1 (1)				MC2 (2)			MC3 (2)			MC4 (3)			MC5 (3)		
T1	T2	T3	T4	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3

PDAP rp /LDAX rp/ /przepisz pośrednio pamięć do
akumulatora/

/A/ ← [/rp/]

Zawartość komórki pamięci o adresie umieszczonym w parze rejestrów rp jest przepisywana do akumulatora.

Uwaga: można wykorzystywać jedynie pary rejestrów BC i DE.

0	0	R	P	1	0	1	0
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)				MC2 (2)		
T1	T2	T3	T4	T1	T2	T3

PAPP rp /STAX rp/ /przepisz pośrednio akumulator
do pamięci/

[/rp/] ← /A/

Zawartość akumulatora jest przepisywana do komórki pamięci o adresie umieszczonym w parze rejestrów rp.

Uwaga: można wykorzystywać jedynie pary rejestrów BC i DE.

0	0	R	P	0	0	1	0
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)				MC2 (3)		
T1	T2	T3	T4	T1	T2	T3

WPR /XCHG/ /wymień parę rejestrów/

/H/ ↔ /D/

/L/ ↔ /E/

Zawartość rejestrów H i L jest wymieniana z zawartością rejestrów D i E.

1	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)			
T1	T2	T3	T4

Instrukcje arytmetyczne

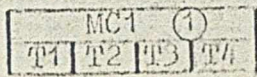
DDJ r /ADD r/ /dodaj rejestr do akumulatora/

/A/ ← /A/ + /r/

Zawartość rejestru r jest dodawana do zawartości akumulatora. Wynik zapisywany jest w akumulatorze.

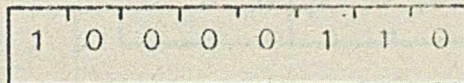
1	0	0	0	0	S	S	S
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

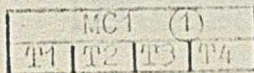


DDJ M /ADD M/ /dodaj pamięć do akumulatora/
 /A/ ← /A/ + [H/L]

Zawartość komórki pamięci o adresie umieszczonym w parze rejestrów HL jest dodawana do zawartości akumulatora. Wynik zapisywany jest w akumulatorze.

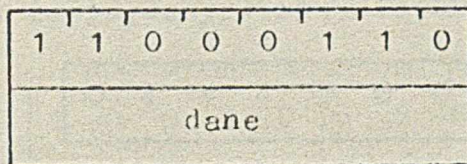


Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC



DDA dane /ADI data/ /dodaj dane do akumulatora/
 /A/ ← /A/ + /bajt 2/

Drugi bajt instrukcji jest dodawany do zawartości akumulatora. Wynik jest zapisywany w akumulatorze.



Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)				MC2 (2)		
T1	T2	T3	T4	T1	T2	T3

DDP dane /ACI data/ /dodaj dane i przeniesienie
do akumulatora/

/A/ ← /A/ + /bajt 2/ + /CY/

Drugi bajt instrukcji oraz bit CY rejestru flagowego są dodawane do zawartości akumulatora. Wynik jest zapisywany w akumulatorze.

1	1	0	0	1	1	1	0
dane							

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)				MC2 (2)		
T1	T2	T3	T4	T1	T2	T3

MDJ r /SUB r/ /odejmij rejestr od akumulatora/
/A/ ← /A/ - /r/

Zawartość rejestru r jest odejmowana od zawartości akumulatora. Wynik jest zapisywany w akumulatorze.

1	0	0	1	0	S	S	S
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)			
T1	T2	T3	T4

MDJ M /SUB M/ /odejmij pamięć od akumulatora/
/A/ ← /A/ - [H/L]

Zawartość komórki pamięci o adresie umieszczonym w parze rejestrów HL jest odejmowana od zawartości akumulatora. Wynik jest zapisywany w akumulatorze.

1	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)				MC2 (2)		
T1	T2	T3	T4	T1	T2	T3

MDA dane /SUB data/ /odejmij dane od akumulatora/
/A/ ← /A/ - /bajt 2/

Drugi bajt instrukcji jest odejmowany od zawartości akumulatora. Wynik jest zapisywany w akumulatorze.

1	1	0	1	0	1	1	0
dane							

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)				MC2 (2)		
T1	T2	T3	T4	T1	T2	T3

MZP r /SBB r/ /odejmij rejestr i pożyczkę od
akumulatora/

/A/ ← /A/ - /r/ - /CY/

Zawartość rejestru r oraz bit CY rejestru flagowego są odejmowane od zawartości akumulatora. Wynik jest zapisywany w akumulatorze.

1	0	0	1	1	S	S	S
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)			
T1	T2	T3	T4

MZP M /SBB M/ /odejmij pamięć i pożyczkę od aku-
mulatora/

/A/ ← /A/ - [H//L/] - /CY/

Zawartość komórki pamięci o adresie umieszczonym w parze rejestrów HL oraz bit CY rejestru flagowego są odejmowane od zawartości akumulatora. Wynik jest zapisywany w akumulatorze.

1	0	0	1	1	1	1	0
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, AC

MC1 (1)				
T1	T2	T3	T4	T5

ZWK M /INR M/ /zwiększ pamięć/

$$[/H//L/] \leftarrow [/H//L/] + 1$$

Zawartość komórki pamięci o adresie umieszczonym w parze rejestrów HL jest zwiększana o 1.

0	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, AC

MC1 (1)				MC2 (2)			MC3 (3)		
T1	T2	T3	T4	T1	T2	T3	T1	T2	T3

ZMN r /DCR r/ /zmniejsz rejestr/

$$/r/ \leftarrow /r/ - 1$$

Zawartość rejestru r jest zmniejszana o 1.

0	0	D	D	D	1	0	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, AC

MC1 (1)				
T1	T2	T3	T4	T5

ZMN M /DCR M/ /zmniejsz pamięć/
[/H//L/] ← [/H//L/] - 1

Zawartość komórki pamięci o adresie umieszczonym w parze rejestrów HL jest zmniejszana o 1.

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, AC

MC1 (1)				MC2 (2)			MC3 (3)		
T1	T2	T3	T4	T1	T2	T3	T1	T2	T3

ZWP rp /INX rp/ /zwiększ parę rejestrów/
/rh//rl/ ← /rh//rl/ + 1

Zawartość pary rejestrów rp jest zwiększana o 1.

0	0	R	P	0	0	1	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)				
T1	T2	T3	T4	T5

ZMP rp /DCX rp/ /zmniejsz parę rejestrów/
/rh//rl/ ← /rh//rl/ - 1

Zawartość pary rejestrów rp jest zmniejszana o 1.

0	0	R	P	1	0	1	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)				
T1	T2	T3	T4	T5

DPR rp /DAD rp/ /dodaj parę rejestrów/
/H//L/ ← /H//L/ + /rh//rl/

Zawartość pary rejestrów rp jest dodawana do zawartości pary rejestrów HL. Wynik jest zapisywany w parze rejestrów HL.

Uwaga: bit CY rejestru flagowego jest ustawiany, gdy wystąpi przeniesienie z najstarszego bitu rejestru H.

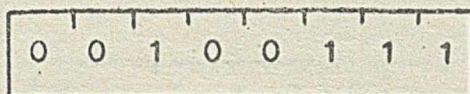
0	0	R	P	1	0	0	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: CY

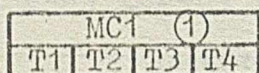
MC1 (1)				MC2 (-)			MC3 (-)		
T1	T2	T3	T4	T1	T2	T3	T1	T2	T3

ZDD /DAA/ /zamień na dwie cyfry dziesiętne/

Liczba zapisana w akumulatorze jest zamieniana z postaci w kodzie binarnym na postać w kodzie BCD w następujący sposób:
1/ jeśli zawartość czterech młodszych bitów akumulatora jest większa od 9 lub bit AC rejestru flagowego jest ustawiony, liczba 6 jest dodawana do zawartości akumulatora;
2/ jeśli po tej operacji zawartość czterech starszych bitów akumulatora jest większa od 9 lub bit CY rejestru flagowego jest ustawiony, liczba 6 jest dodawana do zawartości czterech starszych bitów akumulatora.



Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

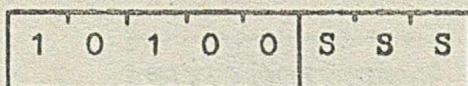


Instrukcje logiczne

PML r /ANA r/ /pomnóż logicznie rejestr i akumulator/

/A/ ← /A/ ^ /r/

Zawartość rejestru r jest mnożona logicznie z zawartością akumulatora. Wynik jest zapisywany w akumulatorze. Bit CY rejestru flagowego jest zerowany.



Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)			
T1	T2	T3	T4

PML M /ANA M/ /pomnóż logicznie pamięć i akumulator/
/A/ — /A/ ^ [H//L/]

Zawartość komórki pamięci o adresie umieszczonym w parze rejestrów HL jest mnożona logicznie z zawartością akumulatora. Wynik jest zapisywany w akumulatorze. Bit CY rejestru flagowego jest zerowany.

1	0	1	0	0	1	1	0
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)				MC2 (2)		
T1	T2	T3	T4	T1	T2	T3

PMD dane /ANI data/ /pomnóż logicznie dane i akumulator/
/A/ — /A/ ^ /bajt 2/

Drugi bajt instrukcji jest mnożony logicznie z zawartością akumulatora. Wynik jest zapisywany w akumulatorze. Bity CY i AC rejestru flagowego są zerowane.

1	1	1	0	0	1	1	0
dane							

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC 1	(1)	MC 2	(2)			
T1	T2	T3	T4	T1	T2	T3

RSM r /XRA r/ /odejmij symetrycznie rejestr od akumulatora/
/A/ ← /A/ ⊕ /r/

Zawartość rejestru r jest symetrycznie odejmowana od zawartości akumulatora. Wynik jest zapisywany w akumulatorze. Bity CY i AC rejestru flagowego są zerowane.

1	0	1	0	1	S	S	S
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC 1	(1)		
T1	T2	T3	T4

RSM M /XRA M/ /odejmij symetrycznie pamięć od akumulatora/
/A/ ← /A/ ⊕ [/H//L/]

Zawartość komórki pamięci o adresie umieszczonym w parze rejestrów HL jest odejmowana symetrycznie od zawartości akumulatora. Wynik jest zapisywany w akumulatorze. Bity CY i AC rejestru flagowego są zerowane.

1	0	1	0	1	1	1	0
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC 1	(1)	MC 2	(2)			
T1	T2	T3	T4	T1	T2	T3

RSD dane /XRI data/ /odejmij symetrycznie dane od akumulatora/

/A/ ← /A/ ⊕ /bajt 2/

Drugi bajt instrukcji jest odejmowany symetrycznie od zawartości akumulatora. Wynik jest zapisywany w akumulatorze. Bity CY i AC rejestru flagowego są zerowane.

1	1	1	0	1	1	1	0
dane							

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC 1 (1)				MC 2 (2)		
T1	T2	T3	T4	T1	T2	T3

DLG r /ORA r/ /dodaj logicznie rejestr do akumulatora/

/A/ ← /A/ ∨ /r/

Zawartość rejestru r jest dodawana logicznie do zawartości akumulatora. Wynik jest zapisywany w akumulatorze. Bity CY i AC rejestru flagowego są zerowane.

1	0	1	1	0	S	S	S
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC 1 (1)			
T1	T2	T3	T4

DLG M /ORA M/ /dodaj logicznie pamieć do akumulatora/

/A/ ← /A/ ∨ [/H//L/]

Zawartość komórki pamięci o adresie umieszczonym w parze rejestrów HL jest dodawana logicznie do zawartości akumulatora. Wynik jest zapisywany w akumulatorze. Bity CY i AC rejestru flagowego są zerowane.

1	0	1	1	0	1	1	0
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)				MC2 (2)		
T1	T2	T3	T4	T1	T2	T3

DLD dane /ORI data/ /dodaj logicznie dane do akumulatora/
/A/ ← /A/ ∨ /bajt 2/

Drugi bajt instrukcji jest dodawany logicznie do zawartości akumulatora. Wynik jest zapisywany w akumulatorze. Bity CY i AC rejestru flagowego są zerowane.

1	1	1	1	0	1	1	0
dane							

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 1				MC2 (2)		
T1	T2	T3	T4	T1	T2	T3

PWN r /CMP r/ /porównaj rejestr z akumulatorem/
/A/ - /r/

Zawartość rejestru r jest odejmowana od zawartości akumulatora. Zależnie od wyniku odejmowania ustawiane są odpowiednie bity rejestru flagowego. Bit Z jest ustawiany, gdy /A/ = /r/. Bit CY jest ustawiany, gdy /A/ < /r/. Zawartość

akumulatora nie ulega zmianie.

1	0	1	1	1	S	S	S
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)			
T1	T2	T3	T4

PWN M /CMP M/ /porównaj pamięć z akumulatorem/

/A/ ← [H//L/]

Zawartość komórki pamięci o adresie umieszczonym w parze rejestrów HL jest odejmowana od zawartości akumulatora. Zależnie od wyniku odejmowania ustawiane są odpowiednie bity rejestru flagowego. Bit Z jest ustawiany, gdy $A = [H//L/]$. Bit CY jest ustawiany, gdy $A < [H//L/]$. Zawartość akumulatora nie ulega zmianie.

1	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---

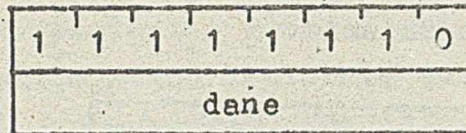
Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)				MC2 (2)		
T1	T2	T3	T4	T1	T2	T3

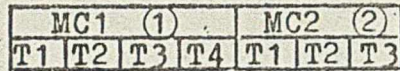
PWD dane /CPI data/ /porównaj dane z akumulatorem/

/A/ - /bajt 2/

Drugi bajt instrukcji jest odejmowany od zawartości akumulatora. Zależnie od wyniku odejmowania ustawiane są odpowiednie bity rejestru flagowego. Bit Z jest ustawiany, gdy $A = \text{/bajt 2/}$. Bit CY jest ustawiany, gdy $A < \text{/bajt 2/}$. Zawartość akumulatora nie ulega zmianie.



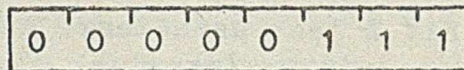
Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC



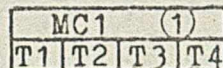
PCL /RLC/ /przesuń cyklicznie w lewo/

$/A_{n+1}/ \leftarrow /A_n/$, $/A_0/ \leftarrow /A_7/$, $/CY/ \leftarrow /A_7/$

Zawartość akumulatora jest przesuwana cyklicznie w lewo o 1 pozycję. Najstarszy bit akumulatora jest przepisywany na miejsce najmłodszego bitu oraz bitu CY rejestru flagowego.



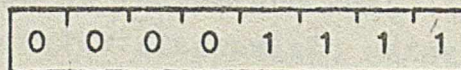
Modyfikowane bity rejestru flagowego: CY



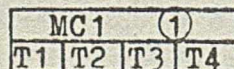
PCP /RRC/ /przesuń cyklicznie w prawo/

$/A_n/ \leftarrow /A_{n+1}/$, $/A_7/ \leftarrow /A_0/$, $/CY/ \leftarrow /A_0/$

Zawartość akumulatora jest przesuwana cyklicznie w prawo o 1 pozycję. Najmłodszy bit akumulatora jest przepisywany na miejsce najstarszego bitu oraz bitu CY rejestru flagowego.



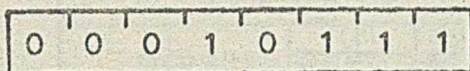
Modyfikowane bity rejestru flagowego: CY



PLP /RAL/ /przesuń cyklicznie w lewo z wykorzystaniem CY/

$$/A_{n+1}/ \leftarrow /A_n/, \quad /CY/ \leftarrow /A_7/, \quad /A_0/ \leftarrow /CY/$$

Zawartość akumulatora jest przesuwana cyklicznie w lewo z wykorzystaniem bitu CY rejestru flagowego. Najstarszy bit akumulatora jest przepisywany na miejsce bitu CY, a bit CY na miejsce najmłodszego bitu akumulatora.



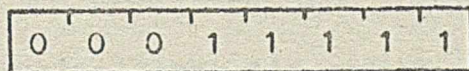
Modyfikowane bity rejestru flagowego: CY

MC1	(1)		
T1	T2	T3	T4

PLR /RAR/ /przesuń cyklicznie w prawo z wykorzystaniem CY/

$$/A_n/ \leftarrow /A_{n+1}/, \quad /CY/ \leftarrow /A_0/, \quad /A_7/ \leftarrow /CY/$$

Zawartość akumulatora jest przesuwana cyklicznie w prawo z wykorzystaniem bitu CY rejestru flagowego. Najmłodszy bit akumulatora jest przepisywany na miejsce bitu CY, a bit CY na miejsce najstarszego bitu akumulatora.



Modyfikowane bity rejestru flagowego: CY

MC1	(1)		
T1	T2	T3	T4

ZIA /CMA/ /zaneguj akumulator/
/A/ ← \bar{A} /

Zawartość akumulatora jest negowana.

0	0	1	0	1	1	1	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)			
T1	T2	T3	T4

ZLP /CMC/ /zaneguj bit przeniesienia/
/CY/ ← \bar{CY} /

Bit CY rejestru flagowego jest negowany.

0	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: CY

MC1 (1)			
T1	T2	T3	T4

USP /STC/ /ustaw bit przeniesienia/
/CY/ ← 1

Bit CY rejestru flagowego jest ustawiany.

0	0	1	1	0	1	1	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: CY

MC1 (1)			
T1	T2	T3	T4

Instrukcje skoku

SKC adr /JMP addr/ /skocz/
 /PC/ ← /bajt 3//bajt 2/

Następną wykonywaną instrukcją jest instrukcja o adresie podanym w drugim i trzecim bajcie instrukcji bieżącej.

1	1	0	0	0	0	1	1
młodsze bity adr							
starsze bity adr							

Modyfikowane bity rejestru flagowego: -

MC1 (1)				MC2 (2)			MC3 (2)		
T1	T2	T3	T4	T1	T2	T3	T1	T2	T3

S warunek adr /J condition addr/ /skocz warunkowo/

Jeśli /CCC/ = 1, /PC/ ← /bajt 3//bajt 2/

Jeśli wartość stanowiącego warunek bitu rejestru flagowego jest równa 1, następną wykonaną instrukcją jest instrukcja o adresie podanym w drugim i trzecim bajcie instrukcji bieżącej. W przeciwnym przypadku wykonywana jest kolejna instrukcja.

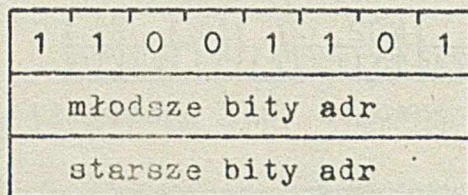
1	1	C	C	C	0	1	0
młodsze bity adr							
starsze bity adr							

Modyfikowane bity rejestru flagowego: -

MC1 (1)				MC2 (2)			MC3 (2)		
T1	T2	T3	T4	T1	T2	T3	T1	T2	T3

<u>KZS adr</u>	<u>/CALL addr/</u>	<u>/skocz do podprogramu/</u>
[/SP/ - 1]	← /PCH/	
[/SP/ - 2]	← /PCL/	
/SP/	← /SP/ - 2	
/PC/	← /bajt 3//bajt 2/	

Zawartość ośmiu starszych bitów rejestru PC jest przepisywana do komórki pamięci o adresie obliczonym przez zmniejszenie o 1 zawartości rejestru SP. Zawartość ośmiu młodszych bitów rejestru PC jest przepisywana do komórki pamięci o adresie obliczonym przez zmniejszenie o 2 zawartości rejestru SP. Zawartość rejestru SP jest zmniejszana o 2. Następną wykonywaną instrukcją jest instrukcja o adresie podanym w drugim i trzecim bajcie instrukcji bieżącej.



Modyfikowane bity rejestru flagowego: -

MC1 (1)					MC2 (2)			MC3 (2)			MC4 (5)			MC5 (5)		
T1	T2	T3	T4	T5	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3

<u>K warunek adr</u>	<u>/C condition addr/</u>	<u>/skocz warunkowo do podprogramu/</u>
----------------------	---------------------------	---

Jeśli /CCC/ = 1,

[/SP/ - 1]	← /PCH/
[/SP/ - 2]	← /PCL/
/SP/	← /SP/-2
/PC/	← /bajt 3//bajt 2/

Jeśli wartość stanowiącego warunek bitu rejestru flagowego jest równa 1, instrukcja jest wykonywana tak, jak instrukcja KZS/CALL/ /patrz wyżej/. W przeciwnym przypadku wykonywana

jest kolejna instrukcja.

1	1	C	C	C	1	0	0
młodsze bity adr							
starsze bity adr.							

Modyfikowane bity rejestru flagowego: -

MC1 (1)					MC2 (2)			MC3 (2)			MC4 (5)			MC5 (5)		
T1	T2	T3	T4	T5	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3

WRC /RET/ /powrót z programu/
 /PCL/ ← [/SP/]
 /PCH/ ← [/SP/ + 1]
 /SP/ ← /SP/ + 2

Zawartość komórki pamięci o adresie umieszczonym w rejestrze SP jest przepisywana do ośmiu młodszych bitów rejestru PC. Zawartość komórki pamięci o adresie o 1 większym jest przepisywana do ośmiu starszych bitów rejestru PC. Zawartość rejestru SP jest zwiększana o 2.

1	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)				MC2 (4)				MC3 (4)			
T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4

W warunek /R condition/ /powrót warunkowo z podprogramu/

Jeśli /CCC/ = 1,

$\text{/PCL/} \leftarrow \text{[/SP/]}$
 $\text{/PCH/} \leftarrow \text{[/SP/ + 1]}$
 $\text{/SP/} \leftarrow \text{/SP/ + 2}$

Jeśli wartość stanowiącego warunek bitu rejestru flagowego jest równa 1, instrukcja jest wykonywana tak, jak instrukcja WRC /RET/ /patrz wyżej/. W przeciwnym przypadku wykonywana jest kolejna instrukcja.

1	1	C	C	C	0	0	0
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)					MC2 (1)			MC3 (1)		
T1	T2	T3	T4	T5	T1	T2	T3	T1	T2	T3

$\text{RST } n \quad \text{/RST/} \quad \text{/restart/}$
 $\text{[/SP/ - 1]} \leftarrow \text{/PCH/}$
 $\text{[/SP/ - 2]} \leftarrow \text{/PCL/}$
 $\text{/SP/} \leftarrow \text{/SP/ - 2}$
 $\text{/PC/} \leftarrow 8 * \text{/NNN/}$

Zawartość ośmiu starszych bitów rejestru PC jest przepisywana do komórki pamięci o adresie obliczonym przez zmniejszenie o 1 zawartości rejestru SP. Zawartość ośmiu młodszych bitów rejestru PC jest przepisywana do komórki pamięci o adresie obliczonym przez zmniejszenie o 2 zawartości rejestru SP. Zawartość rejestru SP jest zmniejszana o 2. Następną wykonywaną instrukcją jest instrukcja o adresie obliczonym przez pomnożenie liczby restartowej n przez 8.

1	1	N	N	N	1	1	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)					MC2 (4)			MC3 (4)		
T1	T2	T3	T4	T5	T1	T2	T3	T1	T2	T3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	N	N	N	0	0	0

Licznik rozkazów po restarcie.

PCHL /PCHL/ /przepisz parę rejestrów do licznika programu/

/PCH/ ← /H/

/PCL/ ← /L/

Zawartość rejestru H jest przepisywana do ośmiu starszych bitów rejestru PC. Zawartość rejestru L jest przepisywana do ośmiu młodszych bitów rejestru PC.

1	1	1	0	1	0	0	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)				
T1	T2	T3	T4	T5

PST rp /PUSH rp/ /przepisz do stosu parę rejestrów/

[/SP/-1] ← /rh/

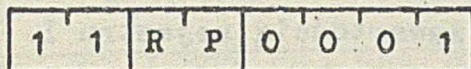
[/SP/-2] ← /r1/

/SP/ ← /SP/-2

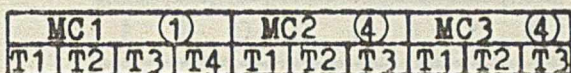
Zawartość pierwszego rejestru pary rp jest przepisywana do komórki pamięci o adresie obliczonym przez zmniejszenie o 1 zawartości rejestru SP. Zawartość drugiego rejestru pary rp jest przepisywana do komórki pamięci o adresie obliczonym przez zmniejszenie o 2 zawartości rejestru SP.

PBR rp /POP rp/ /pobierz parę rejestrów ze stosu/
/r1/ ← [/SP/]
/rh/ ← [/SP/+1]
/SP/ ← /SP/+2

Zawartość komórki pamięci o adresie umieszczonym w rejestrze SP jest przepisywana do drugiego rejestru pary rp. Zawartość komórki pamięci o adresie o 1 większym jest przepisywana do pierwszego rejestru pary rp. Zawartość rejestru SP jest zwiększana o 2. UWAGA: jako para rejestrów rp nie może być wykorzystywany rejestr SP.

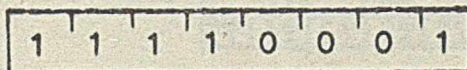


Modyfikowane bity rejestru flagowego: -



PBR PSW /POP PSW/ /pobierz ze stosu słowo stanu procesora/
/CY/ ← [/SP/]₀ , /P/ ← [/SP/]₂ , /AC/ ← [/SP/]₄ ,
/Z/ ← [/SP/]₆ , /S/ ← [/SP/]₇
/A/ ← [/SP/+1]
/SP/ ← /SP/+2

Zawartość komórki pamięci o adresie umieszczonym w rejestrze SP jest wykorzystywana do odtworzenia wartości bitów rejestru flagowego. Zawartość komórki pamięci o adresie o 1 większym jest przepisywana do akumulatora. Zawartość rejestru SP jest zwiększana o 2.



Modyfikowane bity rejestru flagowego: Z, S, P, CY, AC

MC1 (1)				MC2 (4)				MC3 (4)			
T1	T2	T3	T4	T1	T2	T3	T3	T1	T2	T3	

WHL /XTHL/ /wymień parę rejestrów HL z wierzchołkiem stosu/

/L/ ← [/SP/]
/H/ ← [/SP/+1]

Zawartość komórki pamięci o adresie umieszczonym w rejestrze SP jest wymieniana z zawartością rejestru L. Zawartość komórki pamięci o adresie o 1 większym jest wymieniana z zawartością rejestru H.

1	1	1	0	0	0	1	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)				MC2 (4)				MC3 (4)				MC4 (5)			MC5 (5)			
T1	T2	T3	T4	T1	T2	T3	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3	T4	T5

SPHL /SPHL/ /przepisz parę rejestrów HL do wskaźnika stosu/

/SP/ ← /H//L/

Zawartość pary rejestrów HL jest przepisywana do rejestru SP.

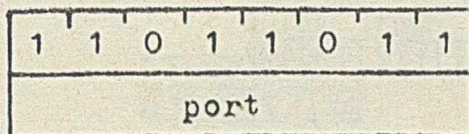
1	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---

Modyfikowane bity rejestru flagowego: -

MC1 (1)				
T1	T2	T3	T4	T5

WE port /IN port/ /wczytaj dane/
/A/ ←← /dane/

Drugi bajt instrukcji jest umieszczany na ośmiu starszych i na ośmiu młodszych liniach szyny adresowej. Dane umieszczone na szynie danych przez adresowane urządzenie wejściowe są wpisywane do akumulatora.

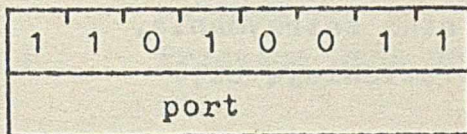


Modyfikowane bity rejestru flagowego: -

MC1 (1)				MC2 (2)			MC3 (6)		
T1	T2	T3	T4	T1	T2	T3	T1	T2	T3

WY port /OUT port/ /wyprowadź dane/
/dane/ →→ /A/

Drugi bajt instrukcji umieszczany jest na ośmiu starszych i na ośmiu młodszych liniach szyny adresowej. Zawartość akumulatora jest umieszczana na szynie danych w celu transmisji do adresowanego urządzenia wyjściowego.

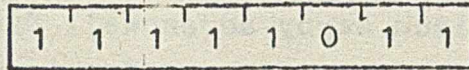


Modyfikowane bity rejestru flagowego: -

MC1 (1)				MC2 (2)			MC3 (7)		
T1	T2	T3	T4	T1	T2	T3	T1	T2	T3

UMS /EI/ /ustaw przerzutnik maski/

Przerzutnik maski przerwań jest ustawiany. Przerwanie będzie mogło nastąpić po wykonaniu następczej instrukcji.

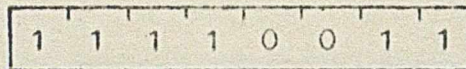


Modyfikowane bity rejestru flagowego: -

MC1		(1)	
T1	T2	T3	T4

ZMS /DI/ /zeruj przerzutnik maski/

Przerzutnik maski przerwań jest zerowany.

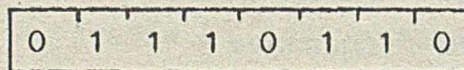


Modyfikowane bity rejestru flagowego: -

MC1		(1)	
T1	T2	T3	T4

STP /HLT/ /stop/

Mikroprocesor wchodzi w stan zatrzymania. Zawartości rejestrów nie zmieniają się.



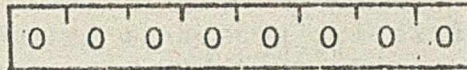
Modyfikowane bity rejestru flagowego: -

MC1		(1)		MC2		(9)		-	-
T1	T2	T3	T4	T1	T2	Tz	Tz	-	-

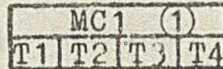
NNR /NOP/

/nic nie rób/

Mikroprocesor nie wykonuje żadnych operacji. Zawartości rejestrów nie zmieniają się.



Modyfikowane bity rejestru flagowego: -



LISTA INSTRUKCJI

Nazwa mnemoniczna	Odpowiadająca nazwa dla 8080A	Opis	Kod instrukcji	Ilość taktów
1	2	3	4	5
Instrukcje przesłań:				
PRZ r1,r2	MOV r1,r2	Przepisz rejestr do rejestru	01DDSSS	5
PRZ r,M	MOV r,M	Przepisz pamięć do rejestru	01DDD110	7
PRZ M,r	MOV M,r	Przepisz rejestr do pamięci	01110SSS	7
PRD r	MVI r	Przepisz dane do rejestru	00DDD110	7
PRD M	MVI M	Przepisz dane do pamięci	00110110	10
PDR B	LXI B	Przepisz dane do pary rejestrów BC	00000001	10
PDR D	LXI D	Przepisz dane do pary rejestrów DE	00010001	10
PDR H	LXI H	Przepisz dane do pary rejestrów HL	00100001	10

Lista instrukcji c.d.

1	2	3	4	5
PAPP B	STAX B	Przepisz pośrednio akumulator do pamięci	00000010	7
PAPP D	STAX D	Przepisz pośrednio akumulator do pamięci	00010010	7
PDAP B	LDAX B	Przepisz pośrednio pamięć do akumulatora	00001010	7
PDAP D	LDAX D	Przepisz pośrednio pamięć do akumulatora	00011010	7
PAP	STA	Przepisz bezpośrednio akumulator do pamięci	00110010	13
PDA	LDA	Przepisz bezpośrednio pamięć do akumulatora	00111010	13
PRP	SHLD	Przepisz bezpośrednio parę rejestrów do pamięci	00100010	16
PPR	LHLD	Przepisz bezpośrednio pamięć do pary rejestrów	00101010	16
WPR	XCHG	Wymień parę rejestrów DE z HL	11101011	4
Instrukcje związane ze stosem:				
PST B	PUSH B	Przepisz do stosu parę rejestrów BC	11000101	11
PST D	PUSH D	Przepisz do stosu parę rejestrów DE	11010101	11
PST H	PUSH H	Przepisz do stosu parę rejestrów HL	11100101	11
PST PSW	PUSH PSW	Przepisz do stosu słowo stanu procesora	11110101	11
PBR B	POP B	Pobierz ze stosu parę rejestrów BC	11000001	10

Lista instrukcji c.d.

1	2	3	4	5
PBR D	POP D	Pobierz ze stosu parę rejestrów DE	11010001	10
PBR H	POP H	Pobierz ze stosu parę rejestrów HL	11100001	10
PBR PSW	POP PSW	Pobierz ze stosu słowo stanu procesora	11110001	10
WHL	XTHL	Wymień parę rejestrów HL z wierzchołkiem stosu	11100011	18
SPHL	SPHL	Przepisz parę rejestrów HL do wskaźnika stosu	11111001	5
PDR SP	LXI SP	Wpisz dane do wskaźnika stosu	00110001	10
ZWP SP	INX SP	Zwiększ wskaźnik stosu	00110011	5
ZMP SP	DCX SP	Zmniejsz wskaźnik stosu	00111011	5
Instrukcje skoku:				
SKC	JMP	Skocz,	11000011	10
SC	JC	Skocz, jeżeli CY = 1	11011010	10
SNC	JNC	Skocz, jeżeli CY = 0	11010010	10
SZ	JZ	Skocz, jeżeli Z = 1	11001010	10
SNZ	JNZ	Skocz, jeżeli Z = 0	11000010	10
SP	JP	Skocz, jeżeli S = 0	11110010	10
SM	JM	Skocz, jeżeli S = 1	11111010	10
SPE	JPE	Skocz, jeżeli P = 1	11101010	10
SPO	JPO	Skocz, jeżeli P = 0	11100010	10
PCHL	PCHL	Przepisz parę rejestrów do licznika programu	11101001	5

Lista instrukcji c.d.

1	2	3	4	5
Instrukcje skoku do podprogramu:				
KZS	CALL	Skocz do podprogramu	11001101	17
KC	CC	Skocz do podprogramu, jeżeli CY = 1	11011100	11/17
KNC	CNC	Skocz do podprogramu, jeżeli CY = 0	11010100	11/17
KZ	CZ	Skocz do podprogramu, jeżeli Z = 1	11001100	11/17
KNZ	CNZ	Skocz do podprogramu, jeżeli Z = 0	11000100	11/17
KP	CP	Skocz do podprogramu, jeżeli S = 0	11110100	11/17
KM	PM	Skocz do podprogramu, jeżeli S = 1	11111100	11/17
KPE	CPE	Skocz do podprogramu, jeżeli P = 1	11101100	11/17
KPO	CPO	Skocz do podprogramu, jeżeli P = 0	11100100	11/17
Instrukcja powrotu z podprogramu:				
WRC	RET	Powrót z podprogramu	11001001	10
WC	RC	Powrót z podprogramu, jeżeli CY = 1	11011000	5/11
WNC	RNC	Powrót z podprogramu, jeżeli CY = 0	11010000	5/11
WZ	RZ	Powrót z podprogramu, jeżeli Z = 1	11001000	5/11
WNZ	RNZ	Powrót z podprogramu, jeżeli Z = 0	11000000	5/11
WP	RP	Powrót z podprogramu, jeżeli S = 0	11110000	5/11

Lista instrukcji c.d.

1	2	3	4	5
WM	RM	Powrót z podprogramu, jeżeli S = 1	11111000	5/11
WPE	RPE	Powrót z podprogramu, jeżeli P = 1	11101000	5/11
WPO	RPO	Powrót z podprogramu, jeżeli P = 0	11100000	5/11
Instrukcje inkrementacji i dekrementacji:				
ZWK r	INR r	Zwiększ rejestr	00000100	5
ZMN r	DCR r	Zmniejsz rejestr	00000101	5
ZWK M	INR M	Zwiększ pamięć	00110100	10
ZMN M	DCR M	Zmniejsz pamięć	00110101	10
ZWP B	INX B	Zwiększ parę rejestrów BC	00000011	5
ZWP D	INX D	Zwiększ parę rejestrów DE	00010011	5
ZWP H	INX H	Zwiększ parę rejestrów HL	00100011	5
ZMP B	DCX B	Zmniejsz parę rejestrów BC	00001011	5
ZMP D	DCX D	Zmniejsz parę rejestrów DE	00011011	5
ZMP H	DCX H	Zmniejsz parę rejestrów HL	00101011	5
Instrukcje dodawania:				
DDJ r	ADD r	Dodaj rejestr do akumulatora	10000SSS	4
DZP r	ADC r	Dodaj rejestr i przeniesienie do akumulatora	10001SSS	4
DDJ M	ADD M	Dodaj pamięć do akumulatora	10000110	7
DZP M	ADC M	Dodaj pamięć i przeniesienie do akumulatora	10001110	7
DDA	ADI	Dodaj dane do akumulatora	11000110	7
DDP	ACI	Dodaj dane i przeniesienie do akumulatora	11001110	7
DPR B	DAD B	Dodaj parę rejestrów BC do HL	00001001	10
DPR D	DAD D	Dodaj parę rejestrów DE do HL	00011001	10

Lista instrukcji c.d.

1	2	3	4	5
DPR H	DAD H	Dodaj parę rejestrów HL do HL	00101001	10
DPR SP	DAD SP	Dodaj wskaźnik stosu do HL	00111001	10
Instrukcje odejmowania:				
MDJ r	SUB r	Odejmij rejestr od akumulatora	10010SSS	4
MZP r	SBB r	Odejmij rejestr i pożyczkę od akumulatora	10011SSS	4
MDJ M	SUB M	Odejmij pamięć od akumulatora	10010110	7
MZP M	SBB M	Odejmij pamięć i pożyczkę od akumulatora	10011110	7
MDA	SUI	Odejmij dane od akumulatora	11010110	7
MDP	SBI	Odejmij dane i pożyczkę od akumulatora	11011110	7
Instrukcje operacji logicznych:				
PML r	ANA r	Pomnóż logicznie rejestr i akumulator	10100SSS	4
RSM r	XRA r	Odejmij symetrycznie rejestr od akumulatora	10101SSS	4
DLG r	ORA r	Dodaj logicznie rejestr do akumulatora	10110SSS	4
PWN r	CMP r	Porównaj rejestr z akumulatorem	10111SSS	4
PML M	ANA M	Pomnóż logicznie pamięć i akumulator	10100110	7
RSM M	XRA M	Odejmij symetrycznie pamięć od akumulatora	10101110	7
DLG M	ORA M	Dodaj logicznie pamięć do akumulatora	10110110	7
PWN M	CMP M	Porównaj pamięć z akumulatorem	10111110	7
PMD	ANI	Pomnóż logicznie dane i akumulator	11100110	7
RSD	XRI	Odejmij symetrycznie dane od akumulatora	11101110	7

Lista instrukcji c.d.

1	2	3	4	5
DLD	ORI	Dodaj logicznie dane do akumulatora	11110110	7
PWD	CPI	Porównaj dane z akumulatorem	11111110	7
Instrukcje rotacji:				
PCL	RIC	Przesuń cyklicznie w. lewo	00000111	4
PCP	RRC	Przesuń cyklicznie w. prawo	00001111	4
PLP	RAL	Przesuń cyklicznie w lewo z wykorzystaniem CY	00010111	4
PLR	RAR	Przesuń cyklicznie w prawo z wykorzystaniem CY	00011111	4
Instrukcje specjalne:				
ZLA	CMA	Zaneguj akumulator	00101111	4
USP	STC	Ustaw bit przeniesienia	00110111	4
ZLP	CMC	Zaneguj bit przeniesienia	00111111	4
ZDD	DAA	Zamień na dwie cyfry dziesiętne	00100111	4
Instrukcje we/wy:				
WE	IN	Wczytaj dane	11011011	10
WY	OUT	Wyprowadź dane	11010011	10
Instrukcje sterujące:				
UMS	EI	Ustaw przerzutnik maski	11111011	4
ZMS	DI	Zeruj przerzutnik maski	11110011	4
NNR	NOP	Nic nie rób	00000000	4
STP	HLT	Stop	01110110	7
RST		Restart	11AAA111	11

INSTYTUT TECHNOLOGII ELEKTRONOWEJ
Al. Lotników 32/46
02-668 Warszawa

tel. 43-54-01
tlx 315647

Cena 720 zł

1936

DRUK ZOINTE IPE zam. 4 / 36 n.

PRAWO REPRODUKCYJI ZASTRZEŻONE