

SYSTEM GEO-20

Dokumentacja Techniczno Ruchowa

Tom 2

Centralny procesor UMC-20

Opracował: mgr inż. Andrzej Papliński
mgr inż. Piotr Parewicz

Zatwierdził: dr Jerzy Szewczyk

Egzemplarz Nr

Wersja

Aktualizowane dn

Opracowanie redakcyjne:
mgr inż. Maria Kochman
mgr inż. Zbigniew Dudek
mgr inż. Andrzej Papliński

Warszawa 1977

Centralny Procesor Maszyny Cyfrowej UMC-20

Spis treści

1. Opis funkcjonalny centralnego procesora.
 - 1.1. Rejestr stanu procesora SR.
 - 1.2. Rozkład informacji systemowej w pamięci operacyjnej i rejestrach R.
 - 1.3. Pulpit procesora.
 - 1.4. Adresowanie.
 - 1.5. Rejestracja przerwania.
 - 1.6. Zasilanie.
 - 1.7. Zbiór instrukcji.
2. Budowa centralnego procesora.
 - 2.1. Wprowadzenie.
 - 2.2. Część wykonawcza.
 - 2.2.1. Rejestry R(N). Warunki AS,DS.
 - 2.2.2. Rejestry BR,RQ.
 - 2.2.3. Multipleksery ABQ,CQ,DQ.
 - 2.2.4. Jednostka arytmetyczno-logiczna ALU. Warunki NQ,TV,OKH.
 - 2.2.5. Rejestr instrukcji IR. Licznik kroków CN. Licznik CA.
 - 2.2.6. Układ adresowania.
 - 2.2.7. Licznik okresów czasowych TM.
 - 2.2.8. Rejestr stanu SR. Przerzutniki RT,TMP.
 - 2.2.9. Warunki efektywności skoków JUW.
 - 2.3. Układ przerwania.
 - 2.4. Mikroprogram sterowania.
 - 2.4.1. Mikrorozkazy fazy operacji pulpitowych MA1,...,MA5.
 - 2.4.2. Mikroprogram rejestracji przerwania: MB1,...,MB9.
 - 2.4.3. Mikrorozkaz MCH przesyłania rozkazu do kanałów we-wy.

- 2.4.4. Mikrorozkazy fazy pobierania instrukcji MC1,...,MC5.
- 2.4.5. Mikrorozkazy MD1,...,MD9, ME1,...,ME4, MJ1,...,MJ4.
- 2.4.6. Mikrorozkazy podstawowych operacji arytmetyczno-logicznych MH1,MH2,MS1,MS2,MS3,MG1,MG2,MG3.
- 2.4.7. Mikrorozkazy mnożenia i dzielenia MP1,...,MP9.

CENTRALNY PROCESOR MASZINY CYFROWEJ UMC20

1. Opis funkcjonalny centralnego procesora

Przed czytaniem niniejszego opisu należy zapoznać się z opisem "Jednostka Centralna UMC20. Opis ogólny", oraz z dodatkiem "Oznaczenia".

Podstawowe funkcje Centralnego Procesora (CP) polegają w ogólności na:

- pobieraniu z pamięci operacyjnej kolejnych instrukcji,
- określaniu adresów argumentów instrukcji w pamięci operacyjnej,
- wykonywaniu operacji opisywanych instrukcjami,
- określeniu adresu następnej instrukcji;
- rejestrowaniu przerw,
- wykonywaniu operacji pulpitowych.

W zależności od typu instrukcji pewne czynności mogą być pominięte. Przykładowo dla instrukcji bezargumentowych pomijane są czynności związane z określaniem argumentu, a dla instrukcji we-wy następuje przekazanie wykonania instrukcji do właściwego kanału transmisji informacji.

Szczegółowe działanie CP opisane będzie podanym w dalszej części mikroprogramem sterowania.

Wszelkie zależności czasowe w CP wyznaczone są przez tzw. układ synchronizacji, który określa czas trwania poszczególnych mikrorozkazów.

1.1. Rejestr stanu procesora SR

Rejestr stanu procesora jest rejestrem 20-bitowym $SR_{19:0}$. Poszczególne bity rejestru stanu identyfikowane są przez swój numer lub jednoliterową nazwę symboliczną np. $U = SR_0 = SR_{15}$. Szczegółowy

sposób ustawiania i wykorzystywania bitów rejestru stanu omawiany będzie sukcesywnie w dalszych częściach opisu. Ogólne przeznaczenie poszczególnych bitów jest następujące:

- 19 T Wskaźnik śledzenia programu.
- 18 M Sposób relokacji adresu.
- 17 P Uwzględnianie przeniesienia wejściowego sumatora
- 16 I }
15 U } Maskowanie przerwania
- 14 S Wskaźnik stanu programu nadrzędnego.
- 13 A }
12 B } Wydłużenia numerów rejestrów R.
- 11 Q Niewykorzystany.
- 10 D Sposób relokacji adresu.
- 9 L }
8 E } Wynik porównywania arytmetycznego.
- 7 Z₃ }
6 Z₂ } Warunki dowolnie ustawiane w programie.
5 Z₁ }
4 Z₀ }
- 3 V Nadmiar w operacjach arytmetycznych.
- 2 C Przeniesienie wyjściowe z sumatora.
- 1 X }
0 Y } Wynik porównywania logicznego oraz
 odpowiedź kanałów we-wy na instrukcję we-wy.

W przypadku gdy T=1 zgłaszane jest przerwanie TRP. Bit T ustawiany jest w stan 1 instrukcjami XTR oraz XITR, a zerowany po odebraniu jakiegokolwiek przerwania.

Bity M oraz P ustawiane są w stan 1 instrukcjami SMF, SPF oraz SMP tylko na czas trwania następnej instrukcji, a po jej wykonaniu są zerowane. W przypadku gdy MVP = 1

- nie są rejestrowane przerwania z wyjątkiem przerwania INE (błędna instrukcja),
- w trybie pracy krokowej nie następuje zatrzymanie po fazie pobierania instrukcji,
- użycie instrukcji SMF, SPF lub SMP powoduje zgłoszenie przerwania INE.

1.2. Rozkład informacji systemowej w pamięci operacyjnej i rejestrach R.

W pamięci operacyjnej określone są poprzez elektronikę systemu następujące adresy:

- adres APR miejsca pamięci, w którym przechowywany jest adres początku pola, przeznaczonego do zapamiętania zawartości rejestrów R podczas rejestracji przerwań APR = 0 ;
- adres APP miejsca pamięci, w którym przechowywany jest adres początku programu obsługi przerwań APP = 1 ;
- adres parametrów transmisji kanału multiplekserowego
licznik transmisji LEN = 0600 + 2 \cdot UN ;
adres pola transmisji ADR = 0601 + 2 \cdot UN ;
- adres AIPL początku programu początkowego (inicjującego) AIPL = 0100000 ;

Obszar pamięci o adresach 0100000 : 0100377 zrealizowany jest w postaci pamięci stałej.

Rejestry R o numerach 0 : 07 są uniwersalnymi rejestrami programowymi dostępnymi dla instrukcji gdy SR_A lub SR_B równe jest 0. W przypadku gdy SR_A lub SR_B równe jest 1 dostępne są dla instrukcji programu rejestry R o numerach 010 : 017.

Wyróżnione są następujące rejestry R:

R(10) = IC licznik instrukcji

R(12) = $QDT \cdot 2^8 + QLT$
R(13) = $RDT \cdot 2^8 + RLT$ } rejestry relokacji i ograniczeń

R(16) = CS przyczyna przerwania
R(17) = SPC specyfikacja przerwania

1.3. Pulpit procesora

- Na pulpicie procesora (rys.B.4) znajduje się:
- stacyjka z kluczykiem do włączania i wyłączania zasilania oraz blokady pulpitu;
 - niestabilny klucz CLR powodujący wysłanie sygnału zerowania systemu;
 - lampka POWER sygnalizująca awarię zasilacza;
 - klawiatura numeryczna złożona z dziewięciu klawiszy oznaczonych C, 0, 1, 2, 3, 4, 5, 6, 7;
 - cztery klucze interpretacyjne oznaczone R, 0, I, B określające sposób (format) wprowadzania i wyświetlania informacji na pulpicie;
 - osiem niestabilnych kluczy operacji pulpitowych:
FETCH, SET, STORE, PUT, GET, IPL, START, CONTINUE;
 - stabilny klucz STEP określający tryb pracy procesora:
praca krokowa - klucz wciśnięty
praca automatyczna - klucz zwolniony
 - dwa niestabilne klucze przerwania pulpitowych oznaczone RESTART, REQUEST;
 - trzy zespoły wyświetlaczy numerycznych oznaczonych AR, K, R służących do wyświetlania zawartości rejestrów:
 - AR adresowego (6 cyfr oktalnych)
 - RK kluczy (6 cyfr oktalnych)
 - RN numeru rejestrów R. (2 cyfry oktalne)
 - dziewięć lampek oznaczonych I, U, S, A, B, Q, D, X, Y wyświetlających odpowiednie bity rejestru stanu;
 - dwie lampki oznaczone WAIT, RUN sygnalizujące ogólny stan

procesora;

- lampki oraz klucze BLANK, TRACE wykorzystywane przy technicznym uruchamianiu i sprawdzaniu procesora.

Kluczyk w stacyjce może znajdować się w trzech stabilnych położeniach oznaczonych OFF, LOCK, ON oraz w położeniu niestabilnym. Kluczyk może być wyjmowany i wkładany tylko w pozycjach OFF oraz LOCK.

Pozycja niestabilna służy do włączania, a pozycja OFF do wyłączenia zasilania. W pozycji LOCK pulpit jest zablokowany i aktywny jest jedynie klucz REQUEST (normalna eksploatacja systemu). Pozostałe klucze na pulpicie aktywne są w położeniu ON kluczyka stacyjki.

Lampka RUN zapala się na czas pobierania i wykonywania instrukcji, a lampka WAIT gdy procesor jest w stanie WAIT. Zgaszenie lampek RUN i WAIT jest sygnalizacją awarii technicznej w systemie.

Klucze przerwań pulpitu mogą być użyte w dowolnym momencie pracy procesora i powodują zgłoszenie przerwań RES oraz ORQ. Przy przerwaniu ORQ stan rejestru kluczy RK przenoszony jest jako specyfikacja przerwania.

Wciśnięcie klucza trybu pracy procesora STEP w czasie wykonywania programu powoduje przejście do stanu WAIT po fazie pobierania kolejnej instrukcji i przejście do trybu pracy krokowej. W rejestrze AR wyświetlany jest fizyczny adres pobranej instrukcji, a sama instrukcja wyświetlana jest w rejestrze kluczy RK. Wciskanie i wyciskanie klucza STEP w stanie WAIT nie powoduje żadnej akcji.

Klucze operacji pulpitu aktywne są jedynie w stanie WAIT procesora. Funkcje poszczególnych kluczy są następujące:

KP	NAZWA	FUNKCJA	OPIS
0.	IPL	R(10) := 0100000	Umieszczenie w R(10) adresu początku programu inicjującego i rozpoczęcie cyklu pracy maszyny od fazy pobrania instrukcji.

KP	NAZWA	FUNKCJA	OPIS
1.	STORE	$S(AR) := RK$ $AR := AR+1$	Wpisanie zawartości rejestru RK do miejsca pamięci o adresie ustawionym w AR, a następnie zwiększenie zawartości AR o 1;
2.	FETCH	$RK := S(AR)$ $AR := AR+1$	Wpisanie do RK (a więc wyświetlenie) zawartości miejsca pamięci o adresie ustawionym w AR, a następnie zwiększenie zawartości AR o 1;
3.	SET	$AR := RK$	Przepisanie zawartości rejestru kluczy RK do rejestru adresowego AR;
4.	PUT	$R(RN) := RK$	Wpisanie do rejestru R którego numer, ustawiony jest w pulpituowym rejestrze numerów RN zawartości rejestru kluczy RK;
5.	GET	$RK := R(RN)$	Wpisanie do RK (a więc wyświetlenie) zawartości rejestru R o numerze RN;
6.	START		Rozpoczęcie cyklu pracy maszyny od fazy pobierania instrukcji, której adres podany jest w $IC=R(10)$. W trybie pracy krokowej po fazie pobierania instrukcji procesor przejdzie do stanu WAIT;
7.	CONTINUE		Rozpoczęcie cyklu pracy maszyny od fazy wykonania aktualnie pobranej instrukcji. W trybie pracy krokowej po wykonaniu tej instrukcji i pobraniu następnej procesor przejdzie do stanu WAIT.

Klawiatura numeryczna wraz z czterema kluczami R, O, I, B służy do ustawiania zawartości rejestrów RN oraz RK.

Przy wciśniętym kluczu R naciśnięcie klawisza C powoduje wyzerowanie rejestru RN a następnie dwa kolejne uderzenia klawiszy numerycznych spowodują umieszczenie odpowiadającej im liczby oktalnej w rejestrze RN.

Przy wciśniętym jednym z kluczy O,I,B naciśnięcie klawisza C spowoduje wyzerowanie rejestru RK. Przy kolejnych naciskaniach klawiszy rejestr RK przesuwany jest w lewo o właściwą liczbę pozycji binarnych, a nowa cyfra umieszczona jest na najmniej znaczących pozycjach. Liczba kolejnych przesunięć i liczba bitów tworzących kolejne wyświetlane cyfry zależy od położenia kluczy O,I,B i wynosi w formie:

oktalnym	O	1, 3, 3, 3, 3, 3
instrukcji	I	2, 3, 3, 3, 2, 3
bajtowym	B	2, 3, 3, 2, 3, 3.

1.4. Adresowanie

Adresy instrukcji, argumentów i wyników używane w programach są adresami względnymi, tzn. że każdy program adresuje swoje dane i instrukcje począwszy od adresu 0. Mechanizm automatycznej relokacji adresów zapewnia, że w czasie wykonywania programu do adresu względnego AW użytego w programie dodawana jest wielkość DT (adres bazowy), a tak utworzony adres rzeczywisty AR, porównywany jest z wielkościami LT (górne ograniczenie) i DT. Poprawnie wyliczony adres rzeczywisty AR spełnia warunek $DT \leq AR < LT$. Jeżeli ten warunek nie jest spełniony generowane jest przerwanie ADE (przekroczenie adresu). Wielkości DT i LT przechowywane są w rejestrach R(12) oraz R(13).

Szczegółowy sposób adresowania zależy od bitów S,D,M rejestru stanu może być opisany zależnościami:

$$AR = AW + W \cdot DT$$

$$ADE = W \cdot (AR < DT \vee AR \geq LT)$$

$$\text{gdzie } DT = 2^8 \cdot R_{15:8}(N)$$

$$LT = 2^8 \cdot R_{7:0}(N)$$

a warunek W oraz numer rejestru N określone są w tabeli.

Tabela relokacji adresów

TRYB	SDM	W	N
1	0XX	1	12
	1XX	0	X
2	000	1	12
	001	1	13
	010	1	13
	011	1	12
	100	1	12
	101	1	13
	110	0	X
111	1	13	

W tabeli wyróżniono dwa tryby relokacji:

Tryb 1 dla adresów związanych z licznikiem instrukcji i dotyczący pobierania:

- instrukcji;
- części adresowej instrukcji długich;

Tryb 2 dla adresów danych (argumentów instrukcji).

Należy zwrócić uwagę, że adresy instrukcji nie są relokowane w przypadku gdy $S=1$, natomiast adresy argumentów - w przypadku gdy $(S,D,M) = (110)$. W tych obydwu przypadkach nie powstaje¹⁾ przekroczenie adresu ($AD=1$).

¹⁾ Dokładniej przekroczenie adresu w przypadku gdy $(S,D,M) = (110)$ powstanie gdy $AR \gg 0177400$ (ostatni fizycznie istniejący adres pamięci jest równy 0100377).

Jeżeli przykładowo $S=D=M=0$ to jak wynika, z tabeli adresy względne instrukcji i danych relokowane są zawartością rejestru $R(12)$.
Jeśli teraz

$$R(12) = 3/8 + 5$$

$$\text{czyli } DT = 3 \cdot 2^8, \quad IT = 5 \cdot 2^8$$

to wszystkie adresy rzeczywiste zawarte będą w przedziale

$$[01400, 02377]$$

1.5. Rejestracja przerwania

Dla każdego typu przerwania tworzone jest 4-bitowe słowo kodowe przyczyny przerwania $CS_3:0$. Poniżej zestawiono przyczyny przerwania podając kolejno: słowo kodowe, nazwę symboliczną przerwania oraz przyczynę przerwania.

017	BSC	Instrukcja skoku do programu nadrzędnego (SVEA, SVEB)
016	SVC	
015	INE	Instrukcja nieistniejąca lub nielegalna
014	ADE	Błąd w adresowaniu (przekroczenie adresu)
012	FPE	Instrukcja zmiennoprzecinkowa
011	PFL	Awaria zasilania (zanik napięć zasilających)
010	RES	Restart (przerwanie z pulpitu procesora)
6	ORQ	Zgłoszenie operatora (przerwanie z pulpitu procesora)
5	TIM	Wyzerowanie licznika okresów czasowych ($TM=0$) Zawartość TM jest zmniejszana o 1 co $92\frac{4}{7}$ us gdy $SR_S=0$
4	TRP	Śledzenie programu ($SR_T=1$)
2	CHM	Przerwanie z kanału multiplekserowego
0	CHS	Przerwanie z kanału selektorowego

Przejdźcie do obsługi przerwania następuje:

- po określeniu adresu rzeczywistego kolejnej instrukcji dla przerwania o kodach $CS=0:011$ o ile poprzednia instrukcja nie była instrukcją $KR, XIR, XTR, XITR$ albo legalną instrukcją SMP, SPF, SMP (w

- tych przypadkach sygnał $\overline{TMP}=1$) oraz dla przzerwania ADE gdy określany adres rzeczywisty instrukcji jest błędny;
- po zdekodowaniu instrukcji SVEA, SVER (przerwania BSC, SVC) albo błędnej (przerwanie INE) albo zmiennoprzecinkowej (przerwanie FPE);
 - w czasie wykonywania instrukcji przy stwierdzeniu błędu w adresowaniu ADE;
 - w stanie WAIT dla przerwania o kodach CS=0,2,5,6,010,011 i przy odpowiednich bitach maskowania przerwania I oraz U;
- dokładnie gdy wystąpi sygnał $\overline{INT}=1$ równy:

$$\begin{aligned} \overline{INT} = & \overline{BSC} \vee \overline{SVC} \vee \overline{INE} \vee \overline{ADE} \vee \overline{FPE} \vee \\ & \vee \overline{TMP} \cdot (\overline{TRP} \vee \overline{SR}_I \cdot (\overline{PFL} \vee \overline{RES} \vee \overline{ORQ}) \vee \\ & \vee \overline{SR}_I \cdot \overline{SR}_U \cdot (\overline{STEP} \cdot \overline{TIM} \vee \overline{CBI} \vee \overline{OHS}) \end{aligned}$$

W przypadku jednoczesnego wystąpienia kilku przyczyn przerwania jako pierwsze obsłużone będzie przerwanie o najwyższym kodzie przyczyny CS.

Przy rejestracji przerwania wykonywane są następujące czynności:

1. zapamiętanie stanu maszyny

$S(S(0)) \quad := \overline{TM} \ ;$
 $S(S(0)+1) \quad := \overline{SR} \ ;$
 $S(S(0)+2) \quad := \overline{IC} \ ;$
 $S(S(0)+3) \quad := \overline{R}(11) \ ;$
...
 $S(S(0)+11) \quad := \overline{R}(17) \ ;$

przy czym w $S(S(0)+2)$ zapamiętuje się:

- aktualny licznik instrukcji $\overline{IC}=\overline{R}(10)$, gdy procesor jest w stanie WAIT
- adres względny instrukcji, która wykonałaby się gdyby nie przyszło przerwanie (z wyjątkiem przerwania z grupy błędów)
- adres względny części operacyjnej błędnej instrukcji przy przerwaniach: FPE, ADE, INE.

2. zapamiętanie informacji o przerwaniu

R(16) := CS ;

R(17) := SPC ;

przy czym bliższa specyfikacja przerwania zależy od typu przerwania i tak:

SPC = IR dla przerwania BSC,SVC,FPE

SPC = SPCF $2^8 + UN$ dla przerwania CHM i CHS

SPC = RK dla przerwania ORQ

3. ustawienie rejestru stanu w stanie "początkowym"

(SR_{19:15},SR₁₁,SR_{9:0}) := 0 ;

(SR_{14:12},SR₁₀) := 1 ;

4. przejście do programu obsługi przerwania

R(10) := S(1);

1.6. Zasilanie

Włączanie zasilania odbywa się w niestabilnej pozycji kluczyka stacyjki na pulpicie. Po włączeniu zasilania generowany jest automatycznie sygnał CLEAR identyczny jak przy naciśnięciu klucza CLR na pulpicie. Po sygnale CLEAR procesor przechodzi do stanu WAIT, a rejestr stanu ustawiany jest następująco:

(SR_{19:17},SR₁₅,SR₁₁,SR_{9:0}) := 0 ;

(SR₁₆,SR_{14:12},SR₁₀) := 1 ;

Wyłączenie zasilania następuje w 2 przypadkach:

- przy przerwie bądź obniżeniu napięcia sieciowego,
- w pozycji OFF kluczyka stacyjki.

W obydwu przypadkach zgłaszane jest przerwanie PFL, a po nim przez około 10 ms utrzymywane są poprawne warunki pracy maszyny. Po tym czasie następuje blokada pamięci operacyjnej, a następnie wyłączenie zasilania.

Przy włączeniu i wyłączeniu zasilania zawartość pamięci opera-

cyjnej nie ulega zniszczeniu.

1.7. Zbiór instrukcji

Listę instrukcji przedstawiono w tabeli "ZBIÓR INSTRUKCJI". W poszczególnych kolumnach tabeli podano postać symboliczną instrukcji w assemblerze MAAS, kod wewnętrzny instrukcji oraz funkcję realizowaną przez instrukcję. W kolumnie INE podane są warunki, przy których dane słowo instrukcyjne reprezentuje instrukcję nielegalną bądź nie istniejącą.

Słowo reprezentujące instrukcję oznaczono przez IR. Znaczenie poszczególnych grup słowa instrukcyjnego zależy w ogólności od instrukcji, ale zazwyczaj jest następujące:

- | | | |
|------------------|---|---|
| $M = IR_{15:14}$ | } | 5-bitowa główna część operacyjna instrukcji |
| $N = IR_{13:11}$ | | |
| $A = IR_{10:8}$ | } | 3-bitowe pola numerów rejestrów R. Pełny 4-bitowy numer rejestru tworzony jest z pola A albo B słowa instrukcyjnego i bitów SR_A albo SR_B rejestru stanu: $(SR_A, IR_{10:8})$, $(SR_B, IR_{7:5})$. |
| $B = IR_{7:5}$ | | |
| $D = IR_{4:3}$ | | postać argumentu instrukcji |
| $E = IR_{2:0}$ | | wydłużenie części operacyjnej instrukcji |

Ponadto oznaczono

$L = IR_{7:0}$ - krótki argument bądź adres argumentu instrukcji

$$0 \leq L \leq 0377$$

$I = -IR_7 \cdot 2^7 + \sum_{i=0}^6 IR_i \cdot 2^i$ - krótki argument instrukcji

$$-0200 \leq I \leq 0177$$

$U = S(IC+1)$ - długi argument bądź adres argumentu instrukcji

$$0 \leq U \leq 0177777$$

IC - licznik instrukcji (adres instrukcji aktualnie wykonywanej)

W opisie wszystkie liczby za wyjątkiem numerów pozycji słów binarnych są liczbami oktalnymi

$S(a:a+1) = (S(a), S(a+1), \dots, S(a+11));$

$R(A) := B / \text{WAR} \Leftrightarrow \underline{\text{if}} \text{WAR} \underline{\text{then}} R(A) := B.$

2. Budowa centralnego procesora

2.1. Wprowadzenie

1.2.1

Schemat blokowy centralnego procesora (CP) pokazano na rys.2.1. Część wykonawczą (informacyjną) procesora obejmującą między innymi blok 16 rejestrów programowych R, jednostkę arytmetyczno-logiczną i układy adresowania przedstawiono w uproszczonej formie na rys.B2 w opisie "Jednostka Centralna UMC20. Opis ogólny".

Centralny procesor współpracuje z kanałami transmisji informacji (procesorami peryferyjnymi) oraz z pamięcią operacyjną za pośrednictwem centralnego łącza systemu CLS. Opis linii CLS oraz mikrooperacji zespołu pamięci operacyjnej znaleźć można w części "Zespół pamięci operacyjnej".

Wewnętrzne łącze procesora oznaczone na rys.2.1 symbolem CPB jest przeznaczone do przesyłania:

- informacji między pulpitem rejestru kluczy i pulpitem rejestru adresowym a blokiem rejestrów procesora;
- argumentów operacji, zmiennoprzecinkowej z rejestrów procesora do jednostki FPU, oraz wyniku operacji w kierunku odwrotnym;
- kodu przyczyny przerwania $CS_{3:0}$ z układu przerwań do bloku rejestrów procesora.

Część wykonawcza przesyła do części sterującej tzw. warunki wewnętrzne oznaczone symbolem X_w . Warunkami tymi, wpływającymi na pracę części sterującej są między innymi słowo z rejestru instrukcji oraz z rejestru stanu, sygnały z bloków części wykonawczej np. z sumatora (liczba na wyjściu jest ujemna, równa zeru itp.).

Do części wykonawczej układ sterowania przesyła słowa S_w zwane wysterowaniem oraz pojedynczy przebieg synchronizujący SXP. Wysterowanie doprowadzane jest do wejść sterujących bloków funkcjonalnych części wykonawczej i określa rodzaj mikrooperacji wykonywanej przez

blok sekwencyjny (np. przesuwanie w lewo, wpisywanie informacji itp.) lub kombinacyjny (dodawanie, odejmowanie itp.). Przebieg synchronizujący SXP doprowadzony jest do wejść synchronizujących (zegarowych) wszystkich bloków sekwencyjnych części wykonawczej.

W części sterującej wyróżnia się trzy podstawowe bloki funkcyjne: układ synchronizacji, rejestr nazwy mikrorozkazu oraz układy kombinacyjne nazwane dekoderni przejść i wyjść. Rejestr nazwy mikrorozkazu przechowuje słowo kodowe M nazwy mikrorozkazu w kodzie "1 z n", tzn. każdemu mikrorozkazowi przyporządkowana jest jedna pozycja w słowie M.

Dla każdego mikrorozkazu są określone:

- nazwa mikrorozkazu,
- czas trwania lub sposób jego zakończenia,
- zbiór mikrooperacji wykonawczych,
- mikrooperacja przejść.

Nazwy mikrorozkazów dobrane są tak, aby łatwo było identyfikować do jakiej grupy należy dany mikrorozkaz. Poniżej podano w sposób orientacyjny do jakiej grupy operacji należą dane mikrorozkazy:

- | | |
|--------------------|--|
| MA1, MA2, ..., MA5 | - operacje pulpitowe |
| MB1, MB2, ..., MB9 | - rejestracja przerw |
| MC1, MC2, ..., MC5 | - pobieranie instrukcji |
| MD1, MD2, ..., MD9 | - { faza określania argumentów instrukcji oraz
faza wykonawcza instrukcji współpracujących z pamięcią |
| ME1, ME2, ME3, ME4 | |
| MG1, MG2, MG3, MG4 | - wykonywanie instrukcji wymiany bajtów i słów w rejestrach |
| MJ1, MJ2, MJ3, MJ4 | - instrukcje skoku i niektóre instrukcje arytmetyczno-logiczne |
| MH1, MH2 | - instrukcje arytmetyczno-logiczne |
| MP1, MP2, ..., MP9 | - operacje mnożenia i dzielenia |

- MS1, MS2, MS3 - instrukcje przesuwania słów
 MCH - mikrorozkaz przesyłania rozkazów do kanałów
 MFL1, MFL2 - współpraca z jednostką zmiennoprzecinkową.

Czas trwania każdego mikrorozkazu wyznaczony jest odstępem pomiędzy dwoma kolejnymi zboczami narastającymi w przebiegu synchronizującym SXP. Narastające zbocze przebiegu synchronizującego powoduje wpisanie nowego słowa WM do rejestru nazwy mikrorozkazu M, a więc przejście do następnego mikrorozkazu oraz wykonanie wszystkich synchronicznych mikrooperacji przez bloki części wykonawczej. Przebieg synchronizujący SXP pokazano na rys.2.2. Na rysunku przedstawiono postać SXP dla trzech różnych typów mikrorozkazów tzw. mikrorozkazów krótkich (M_k) trwających 2·T, długich (M_d) trwających 3·T oraz asynchronicznych (M_z) trwających n·T gdzie M_k, M_d, M_z oznacza wyjścia z rejestru nazwy mikrorozkazu, a T jest okresem podstawowego przebiegu zegarowego: $T = \frac{1}{74} \mu s \approx 71.4 \text{ ns}$. Czas trwania bądź sposób zakończenia mikrorozkazu jest funkcją M, przy czym mikrorozkazy asynchroniczne (M_z) kończone są po pojawieniu się zewnętrznego sygnału synchronizującego oznaczonego na rys.2.2 symbolem ZZ. Asynchronicznymi są mikrorozkazy współpracujące z pamięcią, mikrorozkazy współpracujące z kanałami, oraz mikrorozkaz pobierający wynik operacji zmiennoprzecinkowej z jednostki FPU. Czas trwania mikrorozkazu współpracującego z pamięcią operacyjną, bądź z kanałem, uzależniony jest od czasu oczekiwania na przydział łącza CFS i wykonania żądanej operacji przez pamięć bądź kanał. Zakończenie mikrorozkazu następuje po otrzymaniu sygnału zakończenia operacji z pamięci (strob OUCN) bądź z kanału (strob TRCHN). Natomiast czas trwania mikrorozkazu pobierającego wynik operacji zmiennoprzecinkowej uzależniony jest od czasu wykonywania tej operacji przez jednostkę FPU; zakończenie mikrorozkazu następuje po otrzymaniu sygnału zakończenia operacji (sygnał FPS).

Mikrorozkazem asynchronicznym jest również mikrorozkaz MA1 (stan

WAIT procesora). Wyjście z tego stanu następuje po otrzymaniu z pulpitu polecenia wykonania operacji pulpitowej (sygnał OPN) bądź sygnału INT z układu przerwań.

Zbiór mikrooperacji wykonawczych bloków funkcjonalnych części wykonawczej określony jest przez słowo S_W doprowadzane do wejść sterujących. Występowanie S_W wyznaczane jest dla każdego mikrorozkazu M na podstawie warunków zewnętrznych X_Z i wewnętrznych X_W przez układ kombinacyjny zwany dekodere wyjść. Poza występowaniem układ ten wyznacza również tzw. wyjściowe sygnały sterujące S_Z . Tak więc działanie dekodera wyjść opisuje zależność:

$$(S_Z, S_W) = FW(M, X_W, X_Z)$$

gdzie FW oznacza tzw. funkcje wyjść.

Mikrooperacja przejść określa dla każdego mikrorozkazu M i warunków X_Z, X_W nazwę następnego mikrorozkazu WM . Realizuje to układ kombinacyjny zwany dekodere przejść, którego działanie można opisać równaniem:

$$WM = FP(M, X_W, X_Z)$$

gdzie FP oznacza tzw. funkcje przejść.

Centralny procesor zmontowany jest na czterech pakietach drukowanych o nazwach ACPU, BCPU, CCPU, PCPU. Na pakiecie ACPU zmontowana jest część wykonawcza, część sterująca wraz z układem przerwań zmontowana jest na pakietach BCPU oraz CCPU, a układy pulpitu na płycie PCPU.

Wyjścia wszystkich układów procesora identyfikowane są poprzez nazwę złożoną z trzech części opisujących kolejno:

- pakiet (litery A, B, C, P)
- numer układu scalonego (od 1A do 14K)
- numer końcówki układu scalonego (od 1 do 16).

Przykładowo: B12C11 oznacza 11 końcówkę układu scalonego o numerze 12C na pakiecie BCPU.

Dodatkowo opis sygnału może być poprzedzony numerem arkusza np. 3A 10B oznacza układ 10B na pakiecie ACPU przedstawiony na arkuszu 3 schematów ideowych.

2.2. Część wykonawcza procesora

2.3

Schemat blokowy części wykonawczej centralnego procesora pokazano na rys. 2.3. W części wykonawczej procesora można wyróżnić następujące podstawowe bloki:

1. Blok rejestrów programowych R(N) (układy 2A13G, 2A12G, 2A11G, 2A10G). Układy adresowania rejestrów przedstawione są na ark. 1A.
2. Rejestry przesuwne BR (układy 6A6E, 6A8E) oraz RQ (układy 6A7E, 6A9E) wraz z dodatkową pozycją wydłużającą rejestr RQ z prawej strony RQM1 (przerzutnik 6A8F).
3. 16-bitowy licznik okresów czasowych TM (układy 2A13H, 2A12H, 2A11H, 2A10H). Zawartość licznika jest poprzednikowana co $92\frac{4}{7}$ ns (sygnał TMCK). Po wyzerowaniu wysyłany jest sygnał $TMZ:=(TM=0)$ powodujący przerwanie TIM.
4. 20-bitowy rejestr stanu procesora SR. Pozycje $SR_{0:15}$ przedstawiono na ark. 12A, 13A, a układy pomocnicze na ark. 11A, natomiast pozycje $SR_{19:16}$ na ark. 7B.
5. Jednostka arytmetyczno-logiczna ALU wraz z multiplekserem DQ (ark. 5A). ALU wytwarza trzy warunki:
CAD - przeniesienie wyjściowe,
VAD - nadmiar,
ADZ - zerowy wynik operacji ($AD=0$).
6. Multipleksery ABQ, CQ (ark. 3A, 4A)
7. Sumator relokacji REL (układy 8A2D, 8A3C), modyfikujący 8 najbardziej znaczących bitów adresu programowego (względnej) wielkością DT (rozdz. 1.4), stanowiącą rzeczywisty adres słowa pamięci, względem którego liczone są adresy programowe.

8. 16-bitowy rejestr adresowy AR (układy 7A3G, 7A3F, 7A3E, 7A3D). Przed nawiązaniem współpracy z pamięcią operacyjną, w rejestrze AR umieszczony jest fizyczny adres słowa pamięci operacyjnej, którego dotyczyć będzie przesłanie. Nadajniki adresu na łączy CLS znajdują się na ark. 7A.
9. Układ kontroli adresu złożony z komparatorów ADE1 i ADE2 (układy 8A2C, 8A3A, 8A2B, 8A3B). Zadaniem tego układu jest sygnalizowanie sytuacji polegającej na zaadresowaniu słowa znajdującego się poza wyznaczonym obszarem pamięci (określonym wielkościami DT i LT). Sygnał ADE1 bądź ADE2 powoduje powstanie przerwania ADE.
10. Rejestr instrukcji IR (układy 7A1D, 7A1C, 7A1A, 7A1B), do którego wpisywane jest pobrane z pamięci słowo instrukcyjne.
11. Nadajniki informacji z multipleksera CQ na szyny CPB oraz IFN (ark. 9A).

Przesłania informacji w części wykonawczej procesora odbywają się za pośrednictwem multiplekserów DQ, CQ oraz ABQ. Multiplekser ABQ powtarza na wyjściach AQ_{15:0} słowo z wybranego wejścia, a na wyjściach BQ_{15:0} negację tego słowa. Multiplekser CQ powtarza na wyjściu słowo z wybranego wejścia.

Sterując odpowiednio multiplekserami ABQ i CQ można na szynie CQ (wyjściu multipleksera CQ) otrzymać w szczególności:

1. Prostą lub zanegowaną zawartość dowolnego z rejestrów R(N)
2. Prostą lub zanegowaną zawartość rejestru BR lub rejestru RQ;
3. Zawartość pozycji SR_{15:0} rejestru stanu;
4. Zawartość licznika chwil czasowych TM;
5. Słowo z linii informacyjnych IFN łączy CLS;
6. Słowo przesłane z rejestru kluczy pulpitu RK, pulpitowego rejestru adresowego ARP lub z arytmometru zmiennoprzecinkowego FPU za pośrednictwem szyny CPB;
7. Słowo, którego prawy bajt (pozycje 7:0) pochodzi z rejestru RQ, a

bajt lewy (15:8) - z wyjść sumatora relokacji. Możliwość ta wykorzystywana jest przy relokowaniu adresu: adres umieszczony jest przed relokacją w rejestrze RQ, skąd mniej znaczący bajt (7:0) przesyłany jest bez zmiany na szynę CQ, a bajt bardziej znaczący (15:8) przed wprowadzeniem na szynę CQ modyfikowany jest w sumatorze relokacji wielkością DT;

8. Słowo (w postaci prostej lub zanegowanej), którego prawy bajt pochodzi z rejestru RQ, a bajt lewy - z rejestru BR; możliwość ta wykorzystywana jest przy operacjach wymiany bajtów między rejestrami;
9. Stała 077777 odpowiadająca adresowi ostatniego miejsca pamięci operacyjnej wykorzystywana jest do utworzenia adresu początku programu IPL.

Możliwe jest również przekazanie na szynę CQ tylko prawego lub tylko lewego bajtu z wybranego źródła, przy czym drugi bajt jest wtedy wypełniany zerami bądź jedynekami, zależnie od drogi przesłania. Można również maskować 12 bardziej znaczących bitów przesyłanego słowa, przekazując na szynę CQ tylko 4 najmniej znaczące bity; na wyjściach $CQ_{15:4}$ są wówczas zera. Wreszcie możliwe jest otrzymanie na szynie CQ słowa złożonego z samych zer, lub z samych jedynek.

Z szyny CQ informacja może być przekazana:

1. na linie informacyjne łącza CLS;
2. do rejestru kluczy lub do arytmometru zmiennoprzecinkowego, za pośrednictwem szyny CPB;
3. do jednostki arytmetyczno-logicznej ALU, jako jeden z argumentów;
4. na szynę DQ z pominięciem jednostki ALU;
5. do rejestru adresowego AR.

Do rejestrów R(N) oraz do rejestrów BR i RQ informacja wpisywana jest z wyjść multipleksera DQ, przekazującego:

- Wynik operacji arytmetycznej lub logicznej z wyjść jednostki ALU,

bądź

- Słowo z szyny CQ (na szynie DQ otrzymuje się to słowo w postaci zanegowanej).

Wpis informacji do przerzutników rejestru stanu kontrolowany jest przez przeznaczone dla poszczególnych przerzutników sygnały sterujące. Każdy przerzutnik obudowany jest układem kombinacyjnym, który wybiera, na podstawie sygnałów sterujących, źródło bitu wprowadzane go do danego przerzutnika. Możliwych źródeł informacji wpisywanej do rejestru stanu jest wiele, nie zostały więc one uwidocznione na schemacie blokowym. Zapis informacji synchronizowany jest przebiegiem SXP.

Rejestry BR i RQ są rejestrami przesuwными. Bit wprowadzany podczas przesunięcia na skrajną pozycję rejestru BR (BR_{15} przy przesunięciu w prawo i BR_0 przy przesunięciu w lewo) wyznaczany jest na podstawie sygnałów sterujących przez odpowiedni układ kombinacyjny (wejścia BRR oraz BRL). To samo dotyczy pozycji RQ_{15} rejestru RQ przy przesunięciu w prawo (wejście RQR), natomiast podczas przesunięcia w lewo na pozycję RQ_0 wprowadzany jest bit z przerzutnika RQM1. Bit wpisywany do przerzutnika RQM1 wyznaczany jest przez układ kombinacyjny współpracujący z tym przerzutnikiem, na podstawie odpowiednich sygnałów sterujących.

Wymiana informacji między rejestrami pulpitemi i pozostałymi blokami procesora dokonywana jest za pośrednictwem szyny CPB. Podobnie przesłanie argumentów operacji zmiennoprzecinkowej do jednostki FPU, oraz przesłanie wyniku z jednostki FPU odbywa się poprzez szynę CPB. Szyna CPB pośredniczy również przy przesłaniu kodu przyczyny przerwania z układu przerwań do rejestru R(17) podczas rejestracji przerwania.

Rejestr instrukcji IR, ładowany jest z wyjść zanegowanych, $BQ_{15:0}$ multipleksera ABQ, przekazującego w tym przypadku słowo instrukcyjne

z linii informacyjnych IN łączy CEG.

Jednostka arytmetyczno-logiczna wykonuje operacje dwuargumentowe na argumentach podanych odpowiednio na wejście A i wejście B, oraz operacje jednoargumentowe na argumencie podanym na wejście A. Argumentem A jest słowo pobrane z szyny CQ, a argumentem B może być wartość dowolnego z rejestrów R(N). Wynik operacji przekazywany jest poprzez multiplekser DQ do rejestru BR lub RQ.

Poniżej omówiono szczegółowo działanie poszczególnych bloków części wykonawczej procesora. Do opisu każdego bloku dołączona jest tablica mikrooperacji realizowanych przez dany blok.

2.2.1. Rejestry R(N). Warunki AS,DS.

Rolę bloku rejestrów R(N) pełni pamięć półprzewodnikowa o pojemności 16 słów 16-bitowych (ark.2A). Wybór żadanego rejestru dokonywany jest przez podanie numeru słowa pamięci $N = RFS_{3:0}$. Mikrooperacje R(N) podane są w tab.1.

Informacja wpisywana do wybranego rejestru pobierana jest z szyny DQ. Sterowanie zapisem do lewej i do prawej połówki wybranego rejestru jest niezależne, możliwy jest więc wpis zarówno całego słowa, jak i jednego bajtu (drugi bajt nie ulega zmianie). Zapis informacji dokonywany jest w stanie SXP=0 przebiegu zegarowego przy RWEL=1 dla lewego bajtu i RWER=1 dla bajtu prawego.

Przy RWEL = 0, RWER = 0 na wyjściach pamięci, NR_{15:0} pojawia się zanegowana zawartość wybranego rejestru (odczyt informacji).

Układ adresowania rejestrów R(N) przedstawiono na ark.1A. Słowo RFN_{2:0} umożliwia wprowadzenie numeru rejestrów przez jedną z ośmiu dróg. W czterech pierwszych przypadkach najbardziej znaczący bit numeru pobierany jest z rejestru stanu z pozycji SR_A oraz SR_B. Trzy mniej znaczące bity pochodzą z pola A lub pola B instrukcji (pole A poprzez licznik CA) przy czym numer N może być zwiększony o 1 (dla

RFN_{2:0} = 1 lub 3). Dla RFN = 4 lub 5 numer może być stałą określoną sygnałami RFC_{1:0}. W przypadku gdy RFN = 6 wszystkie cztery bity numeru pochodzą z licznika CA, a w przypadku gdy RFN = 7 - z pulpitowego rejestru numerów RFP.

Pomocnicze rejestry warunków AS i DS zapamiętują znakowy bit dzielnej i dzielnika.

2.2.2. Rejestry BR, RQ.

Rejestry BR (tabela 2) i RQ (tabela 3) synchronizowane są sygnałem SXP, a sterowane sygnałami BRCL, BRS_{1:0}, RQS_{1:0}. Rejestry BR, RQ wykonują mikrooperacje: nic nie rób, przesun w prawo lub w lewo słowo w rejestrze, wpisz słowo z szyn DQ, a rejestr BR dodatkowo mikrooperację zerowania zawartości.

Z rejestrami BR i RQ współpracują układy kombinacyjne BRR, BRL, RQR oraz rejestr RQM1. Zadaniem każdego z tych układów jest wyznaczenie bitu wprowadzanego na skrajną pozycję rejestru BR bądź RQ przy przesunięciu jego zawartości. Jak wynika z tab.2 oraz z tab.3 przy przesunięciu zawartości rejestru:

- BR w prawo, na pozycję BR₁₅ wpisywany jest stan wyjścia BRR
- BR w lewo, na pozycję BR₀ wpisywany jest stan wyjścia BRL
- RQ w prawo, na pozycję RQ₁₅ wpisywany jest stan wyjścia RQR
- RQ w lewo, na pozycję RQ₀ wpisywany jest stan rejestru RQM1.

Wybór bitu przekazywanego na wyjście układu BRR bądź BRL uzależniony jest od wykonywanej instrukcji. Dla instrukcji nie wymienionych w tabeli 2 stan wyjść układów BRR, BRL jest nieistotny, ponieważ podczas ich wykonywania nie występują przesunięcia zawartości rejestru BR.

Układ RQR sterowany jest sygnałem ME4:

$$RQR = BR_0 \cdot \overline{ME4}$$

Wpisywaniem informacji do przerzutnika RQM1 sterują sygnały MP1,

MP2, MP3, ME4. Wpis wybranego bitu do RQM1 dokonywany jest dodatnim zboczem przebiegu zegarowego. Ponadto przerzutnik RQM1 zerowany jest sygnałem MC5.

2.2.3. Multipleksery ABQ, CQ, DQ.

Mikrooperacje multiplekserów zebrane są w tab.4.

ab.4

Multiplekser ABQ przekazuje na wyjścia AQ, BQ słowo ze źródła wybranego kombinacją sygnałów:

$$BQS = B(BQC, BQB, BQA).$$

Wyjście AQ powtarza słowo podane na wybrane wejście, natomiast na wyjściu BQ otrzymuje się negację tego słowa.

Sygnał BQL umożliwia maskowanie lewego bajtu przekazywanego słowa; przy BQL=1 wykonywane jest przesłanie:

$$AQ_{15:8} := 0, \quad BQ_{15:8} := 377$$

Podobnie przy BQR=1 maskowany jest prawy bajt:

$$AQ_{7:0} := 0, \quad BQ_{7:0} := 377$$

Podając jednocześnie BQL=1 i BQR=1 otrzymuje się na wyjściu AQ słowo złożone z samych zer, a na wyjściu BQ - słowo złożone z samych jedynek.

Wybór jednego z 4-ch wejść multipleksera CQ dokonywany jest kombinacją sygnałów

$$CQS = B(CQB, CQA).$$

Wyjście CQ powtarza słowo z wybranego wejścia.

Lewy bajt przekazywanego słowa może być maskowany przez podanie sygnału CQL; przy CQL=1 jest mianowicie:

$$CQ_{15:8} := 0$$

Natomiast sygnał CQRL umożliwia maskowanie bitów 7:4 przekazywanego słowa; przy CQRL=1 jest:

$$CQ_{7:4} := 0$$

Gdy jest jednocześnie CQL=1, CQRL=1, multiplekser CQ przekazuje

tylko 4 najmniej znaczące bity z wybranego źródła, natomiast na wyjściach $CQ_{15:4}$ są zera.

Słowo z wyjścia multipleksera CQ może być przesłane na linie CPB oraz IFN (CLS). Nadajniki pokazane są na ark.9A przy czym:

$$CPB_{15:0} := CQ_{15:0} / CPBE,$$

$$IFN_{15:0} := CQ_{15:0} / INPE.$$

Multiplekser DQ sterowany jest sygnałem DQS, który wybiera jako źródło informacji wyjście ADN jednostki arytmetyczno-logicznej, bądź wyjście multipleksera CQ. Na wyjście DQ przekazywana jest negacja słowa z wybranego źródła.

2.2.4. Jednostka arytmetyczno-logiczna ALU. Warunki NQ,TV,OKR.

Jednostka arytmetyczno-logiczna (tabela 5) wykonuje jedno- i dwuargumentowe operacje arytmetyczne i logiczne na słowach 16-bitowych. Wybór żądanej operacji dokonywany jest słowem sterującym

$$ADS = B(ADM, ADS_3, ADS_2, ADS_1, ADS_0).$$

Ponadto przy operacjach arytmetycznych z układu sterowania podawane jest przeniesienie wejściowe CIN.

Argumentem podawanym na wejście B jednostki ALU (ark.5A i rys.2.3) jest zawartość wybranego rejestru $R(N)$. Natomiast na wejście A podawane jest, za pośrednictwem szyny CQ, słowo z rejestru BR, bądź z rejestru RQ. Na szynę CQ może być przekazana prosta bądź zanegowana zawartość rejestru BR albo RQ; ponadto, sygnały BQL, BQR umożliwiają ustawienie na szynie CQ słowa składającego się z samych zer lub z samych jedynek. W rezultacie więc rodzaj wykonywanej operacji arytmetycznej lub logicznej zależy nie tylko od sygnałów sterujących jednostką ALU, lecz również od sygnałów sterujących multiplekserami ABQ, CQ.

W tab.5 podano wszystkie wykorzystywane operacje i odpowiadające im słowa sterujące. Argumenty operacji podawane są na wejścia A i B

w postaci zanegowanej. Na wyjściach $ADN_{15:0}$ jednostki ALU otrzymuje się zanegowane słowo wyniku. Wynik operacji w postaci prostej oznaczony jest AD przy czym $AD_{15:0} = \overline{ADN_{15:0}}$.

Na wyjściu CAD (5A13E16) pojawia się przeniesienie wyjściowe z najbardziej znaczącej pozycji sumatora, a na wyjściu ADZ (5A10E14) sygnalizowany jest zerowy wynik operacji:

$$ADZ := (AD = 0).$$

Układ badania nadmiaru operacji arytmetycznej zrealizowany jest przy użyciu elementów 5A14E i 5A14F przy czym sytuacje gdy powstaje nadmiar opisane są w tabeli 5. Przez A_{15} , B_{15} , AD_{15} oznaczono odpowiednio bit znakowy argumentu A, bit znakowy argumentu B, oraz bit znakowy wyniku. Oznaczenia te odnoszą się do argumentów i wyniku w postaci prostej (nie zanegowanej). W lewej części tabeli opisano sytuacje w których może powstać nadmiar VAD, a w prawej odpowiadające tym sytuacjom działanie dekodera 5A14E i multipleksera 5A14F. Warunki ADZ oraz VAD zapamiętywane są w przerzutnikach NQ (5B7E8) oraz TV (2A8F8) zgodnie z tab.5.

2.2.5. Rejestr instrukcji IR. Licznik kroków CN. Licznik CA.

0.6 Mikrooperacje IR, CA oraz CN zebrane są w tab.6. Wpisanie słowa instrukcyjnego do rejestru IR (ark.7A) dokonywane jest opadającym zboczem przebiegu zegarowego SXP w mikrorozkazie MC4. Słowo instrukcyjne, przesłane z pamięci liniami IFN, przekazywane jest do rejestru IR za pośrednictwem szyny BQ:

$$BQ_{15:8} := IFN_{15:8}$$

Pole A słowa reprezentującego instrukcję ($IR_{10:8}$) jest w mikrorozkazie MC4 umieszczane w liczniku CA (element 1A14C), który wykorzystywany jest w układach adresowania rejestrów R(N). Mikrooperacja następnikowania słowa w liczniku CA $IN(CA) = B(CA) + 1$ wykorzystywana jest przy:

- wykonywaniu instrukcji grupowego przesyłania słów pomiędzy pamięcią operacyjną i rejestrami R(N):

34A XXX
35A XXX

- zapamiętywaniu stanu procesora przy przerwaniach i odtwarzaniu stanu przy wykonywaniu instrukcji powrotu z programu obsługi przerwania. W tych przypadkach zliczanie rozpoczyna się od wartości CA=05 przy czym CA=06 wskazuje, że przesyłane jest słowo pomiędzy TM a pamięcią, przy CA=07 przesyłane jest słowo pomiędzy SR a pamięcią, natomiast dla CA \geq 10 przesyłane są rejestry R(CN). Koniec przesyłania sygnalizowany jest warunkiem:

CA17:=(CA=17)

Pole E słowa instrukcyjnego ($IR_{2:0}$) umieszczane jest w czasie trwania mikrorozkazu MC5 w liczniku kroków CN (element 8B7C).

Licznik CN wykorzystywany jest do zliczania:

1. kolejnych kroków przy wykonywaniu mnożenia lub dzielenia,
2. kolejnych przesunięć słów w rejestrach R(A) przy wykonywaniu jednej z instrukcji:

00A 6XE
00A 7XE

3. liczby przesłanych słów przy przesyłaniu zawartości kilku rejestrów do pamięci lub na odwrót, podczas wykonywania jednej z instrukcji:

34A XXE
35A XXE

4. liczby przesłanych słów podczas przesyłania do jednostki FPU argumentów operacji zmiennoprzecinkowej oraz przy pobieraniu wyniku operacji zmiennoprzecinkowej (liczba zmiennoprzecinkowa reprezentowana jest przez dwa słowa).

Licznik kroków wytwarza warunek:

CNZ:=(CN=0)

2.2.6. Układ adresowania.

Układ adresowania składa się z rejestru adresowego AR wraz z nadajnikiem adresu ARN (ark.7A), sumatora relokacji REL i układu kontroli adresu ADE1, ADE2 (ark.8A).

Sumator relokacji wyznacza 8-bitowe słowo $REL_{15:8}$ przy czym:

$$B(REL_{15:8}) := B(RQ_{15:8}) + B(BQ_{15:8}) + CREL$$

W rejestrze RQ umieszczany jest adres względny. W przypadku gdy adres ten nie podlega relokacji (rozdz.1.4) $CREL=1$, a na wyjściach $BQ_{15:8}$ ustawiane są same jedynki. Tak więc w tym przypadku na wyjściach REL pojawia się niezmodyfikowany adres:

$$REL_{15:8} = RQ_{15:8}$$

W przypadku gdy adres podlega relokacji $CREL=0$, na wyjściach $BQ_{15:8}$ pojawia się wielkość DT (rozdz.1.4).

Do rejestru adresowego adres przesyłany jest poprzez multiplexer CQ (tab.4,7) który umożliwia przesłanie:

$$CQ_{15:0} := (REL_{15:8}, RQ_{7:0})$$

Mikrooperacja zwiększania o 1 zawartości rejestru AR wykonywana jest przy operacjach grupowego przesyłania słów pomiędzy pamięcią operacyjną oraz rejestrami R(N).

Układ kontroli adresu składa się z dwóch części wytwarzających odpowiednio sygnały ADE1 i ADE2. Sygnał ADE1 wykorzystywany jest w mikrorozkazach, w których adres rzeczywisty tworzony jest przez sumowanie adresu względnego przy pomocy sumatora relokacji, a sygnał ADE2 - w mikrorozkazach w których adres rzeczywisty tworzony jest poprzez zwiększanie o 1 zawartości rejestru AR.

$$ADE1 := RQ_{15} \cdot \overline{CREL} \vee B(REL_{15:8}) \geq B(BQ_{7:0})$$

$B(BQ_{7:0})$ reprezentuje górne ograniczenie adresu LT, a RQ_{15} jest pozycją znakową adresu względnego. W przypadku gdy nie ma relokacji $CREL=1$ $BQ_{7:0}=377$ (same jedynki) i przekroczenie adresu powstanie gdy:

$$ADE1 := B(RQ_{15:8}) \geq 377$$

a więc dla adresów względnych $RQ \geq 177400$. W przypadku gdy relokacja zachodzi $CPRL=0$ przekroczenie adresu nastąpi gdy adres względny jest ujemny (wtedy $AR < DT$) lub adres rzeczywisty jest nie mniejszy niż DT . Podobnie

$$ADE2 := B(AR_{15:8}) \geq B(BQ_{7:0})$$

W przypadku gdy $AR_{15:8} = BQ_{7:0}$ na wyjście ADE2 przenoszony jest stan wejścia 8A3B4, a zatem $ADE2=1$.

Przy sygnale $SADR=0$ zanegowane zawartość rejestru AR przesyłana jest na linie ARN należące do CFS:

$$ARN := \overline{AR/SADR}$$

2.2.7. Licznik okresów czasowych TM.

Licznik okresów czasowych TM (ark.2A) jest zapełniany słowem z pamięci z miejsca $S(S(0))$ przy wykonywaniu instrukcji powrotu z przerwania (sygnał TMLD - tabela 8).

W przypadku gdy $SR_{14}=0$ sygnał TMCK powoduje zmniejszenie zawartości licznika TM o 1 co $92\frac{4}{7}$ μs . Gdy zawartość licznika wynosi 0 generowany jest sygnał TMZ powodujący zgłoszenie przerwania TIM. Przy ciągłym zliczaniu maksymalny przedział czasu pomiędzy dwoma kolejnymi sygnałami TMZ wynosi $2^{16} \cdot 92\frac{4}{7} \text{ us} = 6.07 \text{ sek}$.

Przy rejestracji dowolnego przerwania słowo z TM zapamiętywane jest w miejscu $S(S(0))$ pamięci. Po każdym przerwaniu $SR_{14}=1$, a więc zawartość TM nie jest poprzednikowana.

Sygnał TMZ nie powoduje zgłoszenia przerwania TIM w przypadku gdy procesor jest w stanie pracy krokowej.

2.2.8. Rejestr stanu SR. Przerzutniki BT, TMP.

Mikrooperacje wykonywane przez poszczególne pozycje rejestru stanu podane są w tab.9 i 10.

Pozycje $SR_{15:0}$ rejestru stanu pokazano na ark. 13A i 12A, a związane z nimi układy wejściowe na ark. 11A. Pozycje $SR_{19:16}$ oraz przerzutniki pomocnicze RT i TMP na ark. 7B.

Sygnal zerowania początkowego CLRN oraz ostatni mikrorozkaz fazy rejestracji przerwań MB9 powoduje umieszczenie w rejestrze stanu słowa:

$SR_{15:0} := 07200/SRCR$; $SR_{19:17} := 0/SRCR$
(jedynki na pozycjach S, A, B, D).

Na pozycji I wpisywana jest jedynka po zerowaniu, a zero po rejestracji przerwań.

Informacja na pozycje $SR_{15:0}$ może być wpisywana równolegle z linią DQ w czasie wykonywania instrukcji powrotu z przerwania (00X 4X1 oraz instrukcji PUT A (00A 513):

$SR_{15:0} := DQ_{15:0}$

Stany przerzutników C, V, D, Q, B, A, U, I, P, M mogą być ponadto ustawiane instrukcjami 377 XX7. W tym celu przewidziano możliwość:

- dla przerzutników C, U, D, Q, B, V, I - wpisania bitu IR_3 słowa instrukcyjnego, a dla przerzutnika P - negacji bitu IR_3 ;
- dla przerzutnika A - wpisania bitu IR_4 słowa instrukcyjnego, a dla przerzutnika M - wpisania negacji bitu IR_4 .

Poniżej zestawiono instrukcje przy których wykonywane są pozostałe mikrooperacje poszczególnych przerzutników rejestru stanu

- Przerzutniki Y, X (SR_0, SR_1)

$(Y, X) := (BQ_{15}, BQ_{15} \oplus BQ_{14}) / (00A 512)$

$(Y, X) := (\overline{ADZ}, ADZ) / (17A XXX \vee 27A XXX \vee 36A XX7)$

$X := \overline{MZN} / (20_7^6 XXX \vee 36_7^6 X20_3^1)$

$\overline{ADZ}=1$ gdy słowo z sumatora jest równe zero.

$MZN=0$ gdy słowo pobrane z pamięci jest równe zero.

- Przerzutnik C (SR_2)

$C := BQ_0 / (00A 510)$

$$C := BQ_{15} / (00X 511)$$

$$C := CAD / IART \text{ gdzie } IART = (13A \overset{1}{XXX} \vee \overset{1}{23A} \overset{1}{XXX} \vee \overset{1}{36A} \overset{1}{XX3})$$

$$C := BR_0 / (00A 71X)$$

$$C := ER_{15} / (00A \overset{1}{62X})$$

CAD jest przeniesieniem wyjściowym z sumatora

- Przerzutnik V (SR₃)

$$V := V \vee VAD / IART$$

$$V := V \vee BR_{14} \oplus BR_{15} / (00A 62X)$$

$$V := V \vee ADZ / (37A B30)$$

$$V := V \vee AS \cdot DS \cdot AD_{15} / (37A B20)$$

$$V := V \vee \overline{ADZ} / (37A B30)$$

- Przerzutniki Z_{3:0} (SR_{7:4})

dla i = 0, 1, 2, 3

$$Z_i := (Z_i \cdot \overline{IR}_1 \vee BQ_1 \cdot IR_1) / (00A 5_3^2 X)$$

$$Z_i := (Z_i \cdot \overline{IR}_1 \vee IR_{1+4} \cdot IR_1) / (077 XXX)$$

- Przerzutniki E, L (SR₈, SR₉)

$$(E, L) := (ADZ, DQ_{15}) / (12A \overset{1}{XXX} \vee \overset{1}{22A} \overset{1}{XXX} \vee \overset{1}{36A} \overset{1}{XX2})$$

$$(E, L) := (\overline{WZN}, DQ_{15}) / (202 \overset{1}{XXX} \vee \overset{1}{362} \overset{1}{X20})$$

Przerzutnik TMP (tabela 10) zapalany jest po pobraniu legalnej instrukcji ustawiającej przerzutniki M i P:

0
377 717
2

lub legalnej instrukcji powrotu z przerwania: 00X 4X1 i jest gaszony po pobraniu instrukcji następnej.

Sygnal TMP=1 uniemożliwia przejście do obsługi ewentualnego przerwania (z wyjątkiem przerwania ADE, FPE, BSC, SVC, INE), przed wykonaniem instrukcji następującej bezpośrednio po wymienionych instrukcjach. Podobnie w przypadku pracy krokowej sygnał TMP =1 uniemożliwia również przejście do stanu WAIT przed wykonaniem następnej instrukcji. Wreszcie TMP=1 nie dopuszcza do wyzerowania przerzutników M i P

przed wykonaniem następnej instrukcji (gdy $TMP=0$, przerzutniki te są zerowane w czasie fazy pobrania kolejnej instrukcji). Przerzutnik TMP jest zerowany przy przejściu do obsługi przerwania (jest to konieczne w przypadku gdy wystąpiło przerwanie ADE, BSC, SVC, INB lub PEP przy $TMP=1$).

Przerzutnik RT zapalany jest po pobraniu legalnej instrukcji powrotu z przerwania

00X 4X1

Sygnal z tego przerzutnika wykorzystywany jest do wytworzenia warunków w mikrooperacjach fazy rejestracji przerwania.

2.2.9. Warunki efektywności skoków JUV.

Gdy pobrana instrukcja jest instrukcją skoku, układ JUV (ark.10A) bada odpowiednie bity rejestru stanu (zgodnie z listą instrukcji) w celu stwierdzenia, czy właściwy dla danej instrukcji warunek skoku jest spełniony. Układ JUV ma trzy wyjścia odpowiadające trzem grupom instrukcji skoku.

Na wyjściu JUV5 otrzymuje się sygnał efektywności skoku dla instrukcji

05X XXX

37X XX5

na wyjściu JUV6 - sygnał efektywności skoku dla instrukcji

06X XXX

37X XX6

na wyjściu JUV7 - sygnał efektywności skoku dla instrukcji

07X XXX

37X XX7

Warunki JUV5, JUV6, JUV7, doprowadzone są do wejść układu realizujących funkcje przejść.

2.3. Układ przerwania

2.4 Zadaniem układu przerwania (rys.2.4) jest zarejestrowanie sygnałów przerwania programowych i zewnętrznych, utworzenie 4-bitowego numeru przyczyny przerwania $CS_{3:0}$ dla przerwania o najwyższym priorytecie i wyzerowanie przerzutnika zgłoszenia przerwania po zarejestrowaniu przerwania (wykonaniu mikroprogramu rejestracji przerwania).

W skład układu przerwania wchodzi następujące bloki (rys.2.4):

- układy zgłaszania przerwania zewnętrznych
- rejestr przerwania zewnętrznych i przerwania programowych
- rejestr numeru kanału
- koder przyczyny przerwania
- nadajniki przyczyny przerwania na linii CFB
- demultiplekser przyczyny przerwania.

Układy zgłoszeń przerwania zewnętrznych składają się z:

- układów formowania impulsów synchronicznych z zegarem $Z1$ trwających $2xT = 140$ ns dla przerwania PFL (elementy 6B13I) oraz dla przerwania TMZ (elementy 6B14K) - impulsy PFL2N i TIM2N
- przerzutniki zgłoszeń przerwania zewnętrznych wraz z układem maskowania przerwania (tab.10) wytwarzającym sygnały zgłoszeń przerwania PFLW, RESW, REQW, TIMW, CHIW (elementy 7B13D, 7B11B9, 7B11A9, 7B10C11, 7B10C3, 7B9B8, 7B10B6, 7B10B8).

Po zakończeniu rejestracji przerwania, w mikroczkasku MB9, zerowany jest przerzutnik zgłoszenia przerwania za pomocą jednego z sygnałów $CS_{11}, CS_{10}, CS_{06}, CS_{05}$ z demultipleksera przyczyny przerwania (element 8B10A).

Stan przerzutników zgłoszeń przerwania przepisany jest do rejestru przerwania zewnętrznych (element 7B10D) na początku mikro-rozkazu MC2 i w czasie trwania mikro-rozkazu MA1 (sygnał INCK).

Na wyjściu rejestru przerwań wytwarzany jest warunek:

$$\text{INT} = \overline{\text{PFLZ}} \vee \overline{\text{RESZ}} \vee \overline{\text{ORQZ}} \vee \overline{\text{TINZ}} \vee \overline{\text{TRZ}} \quad (\text{element 7B9D})$$

przy czym $\overline{\text{TRZ}} = \text{SR}_T \cdot \overline{\text{TMP}}$ (przerwanie TRP).

Działanie poszczególnych przerzutników rejestru przerwań programowych (elementy 2C3K i 2C2B) opisane jest w tab.11.

Przerwania BSC, SVC, INE, zgłaszane są odpowiednio sygnałami WBSC, WSVC, WINE, pochodzącymi z dekodera rejestru instrukcji. Zgłoszenie wpisywane jest do właściwego przerzutnika rejestru zgłoszeń w czasie trwania mikrorozkazu MC5. Przerwanie ADE zgłaszane jest sygnałem WADE. Sygnał ten może powstać w czasie wykonywania jednego z mikrorozkazów sprowadzających wyznaczony adres przed nawiązaniem współpracy z pamięcią (są to mikrorozkazy MC2, MD2, MD5, MD7, MD9). Wpisanie zgłoszenia do przerzutnika ADE dokonywane jest sygnałem

$$\text{MRQS} = \text{MC2} \vee \text{MD2} \vee \text{MD5} \vee \text{MD7}$$

albo sygnałem MD9.

Przerwanie FPE zgłaszane jest:

- w wyniku zdekodowania instrukcji $10A B_3^2 E$ przy braku jednostki zmiennoprzecinkowej w procesorze (sygnał FPD),
- w wyniku otrzymania z jednostki FPU sygnału błędu operacji zmiennoprzecinkowej (sygnał FPE1).

W pierwszym przypadku zgłoszenie wpisywane jest do przerzutnika FPE w czasie wykonywania mikrorozkazu MC5, a w drugim przypadku - w czasie wykonywania mikrorozkazu MF2.

Rejestr numeru kanału CHR składa się z przerzutników CHR1, CHR2, CHR3, CHR4, odpowiadających poszczególnym kanałom. W czasie wykonywania mikrorozkazu MB4, gdy pobierana jest z kanału specyfikacja przerwania, do przerzutników CHR1, CHR2, CHR3, CHR4, zostają wpisane odpowiednio negacje stanów linii GRN1, GRN2(MUX), GRN3, GRN4(SEL). Zostaje więc zapalony przerzutnik o numerze odpowiadającym

jącym numerowi kanału zgłaszającego przerwanie.

2 Koder przerwania wybiera przerwanie o najwyższym numerze (priorytecie) i tworzy dla niego 4-bitowe słowo $CS_{3:0}$ zgodnie z tab.12 (elementy 8B6D, 8B7B, 8B8B).

Numer przerwania wybranego przez koder priorytetowy jest przekazywany do układu sterowania procesora, a podczas wykonywania mikrorozkazu MB7, numer przerwania jest ponadto przekazywany na szynę CPB poprzez nadajniki 8B7A. Oprócz numeru wybranego przerwania, układ kodera priorytetowego przekazuje do sterowania sygnał ICZ (8B8B8). $ICZ = 1$ gdy na wejścia kodera podane jest co najmniej jedno zgłoszenie; w przeciwnym wypadku $ICZ = 0$. Sygnał ICZ jest warunkiem dla mikroprogramowania przerwania, będzie to omówione dokładniej przy opisie odpowiednich mikrorozkazów.

Demultiplekser numeru przerwania wykorzystywany jest w przypadku, gdy przerwaniami przyjętym do obsługi (tzn. przerwaniami o najwyższym priorytecie) jest jedno z przerwania PFL, RES, ORQ, TIM). W takim przypadku sygnał z dekodera zeruje pozycję rejestru zgłoszeń przerwania odpowiadającą wybranemu zgłoszeniu. Sygnał zerujący z dekodera przesyłany jest do rejestru wejściowego w czasie wykonywania mikrorozkazu MB9.

Sygnał zerowania systemu, CLRN, zeruje wszystkie przerzutniki rejestru zgłoszeń przerwania.

W stanie WAIT (mikrorozkaz MA1) zerowane są przerzutniki BSC, SVC, INB, ADE, FPE rejestru przerwania programowych.

Rejestr numeru kanału jest zerowany w czasie wykonywania mikrorozkazu MB2.

2.4. Mikroprogram sterowania

Przy opisie mikroprogramu sterowania (w tabeli) kolejne mikro-

rozkazy przedstawiono w formie:

MN/ZZ : < mikrooperacje wykonawcze > — (mikrooperacja przejść);
MN oznacza nazwę mikrorozkazu, ZZ — zewnętrzny sygnał synchronizujący. Brak ZZ oznacza mikrorozkaz o stałym czasie trwania (patrz 2.1). Kolejne mikrooperacje wykonawcze oddzielane są przecinkami.

Mikrooperacją może być nazwa wyjściowego sygnału zewnętrznego
np. PGN:

MB4/TRCH : PGN, CHR:=GR,...

Jeśli mikrooperacja lub grupa mikrooperacji wykonywana jest warunkowo to warunek pisany jest po kresce ukośnej np.:

MC2: (ARW=RQ,WADE=0)/SR_S;...

Mikrooperacje synchroniczne (wykonywane na końcu mikrorozkazu przy narastającym zboczu przebiegu synchronizującego SXP) wyróżnione są znakiem podstawienia := natomiast asynchroniczne (realizowane głównie przez bloki kombinacyjne) znakiem podstawienia =
np.

ARW=RQ, AR:=ARW

W tabeli podano również kody mikrooperacji wykonawczych (słowo sterujące blokami części wykonawczej) przy czym w ostatniej kolumnie podano nazwy pozostałych sygnałów aktywnych w danym mikrorozkazie.

Mikrooperację zgłoszenia się do pamięci operacyjnej (patrz rys. 2.7) oznaczono symbolicznie MRQ(MOP,AR) gdzie MOP jest słowem kodowym operacji pamięci, a AR — adresem miejsca pamięci.

Dla bezwarunkowej mikrooperacji przejścia nazwa następnego mikrorozkazu pisana jest w nawiasie np.

MA3: ... →(MA2);

Warunkowe mikrooperacje przejścia przedstawione są w formie listy przełączanej warunkiem W:

$\rightarrow (MN_0, MN_1, \dots, MN_{k-1})^W$

gdzie warunek W przyjmuje k wartości $0, 1, \dots, k-1$ wskazując MN_1 .
 MN_1 jest nazwą następnego mikrorozkazu bądź listą np.

$MB6/GRC:\dots \Rightarrow ((MB7, MC1)RT, MB6)(CA > 17)$

co oznacza, że gdy $CA > 17$ następnym mikrorozkazem będzie mikrorozkaz $MB6$ w przeciwnym przypadku dla $RT=0$ nastąpi przejście do $MB7$ a dla $RT=1$ przejście do $MC1$.

2.4.1. Mikrorozkazy fazy operacji pulpitowych $MA1, MA2, MA3, MA4, MA5$

MA1

Stan wykonywania mikrorozkazu $MA1$ nazywany jest stanem WAIT centralnego procesora. Przejście do $MA1$ następuje po sygnale CLEAR a ponadto:

- po mikrorozkazie $MC4$ w obecności sygnału $PK8 = 1$, pod warunkiem, że $TMP = 0$. Sygnał $PK8=1$ występuje, gdy wciśnięty jest klucz STEP (tryb pracy krokowej), natomiast $TMP=0$ oznacza, że ostatnio wykonywana instrukcja nie była instrukcją ustawiającą przerzutniki M i P rejestru stanu, ani instrukcją powrotu z przerwania. Jeśli $TMP=1$, przejście do $MA1$ z $MC4$ nastąpi dopiero po wykonaniu następnej instrukcji,
- po mikrorozkazie $MC5$, gdy zdekodowana została instrukcja $COx 4x0$ legalna tylko w stanie programu nadrzędnego.

Sterowanie procesora pozostaje w stanie mikrorozkazu $MA1$ aż do ponownego wyzwolenia przebiegu zegarowego SXP , co następuje pod wpływem sygnału OPN z pulpitu; bądź sygnału INT z układu przerwania.

Sygnał OPN powstaje w wyniku naciśnięcia jednego z kluczy operacji pulpitowych. Najbliższe dodatnie zbocze przebiegu SXP po jego odblokowaniu powoduje przejście do właściwego dla danej operacji pulpitowej mikrorozkazu. Przejściem steruje przesyłany z

pulpitu kod operacji pulpitowej KP:

- gdy $KP=0$ (operacja pulpitowa IPL) do licznika instrukcji $IC=R(10)$ wpisywany jest zmniejszony o 1 adres pola pamięci stałej zawierającej program początkowy i następuje przejście do mikrorozkazu MC1,
- gdy $KP=1,2$ (operacje pulpitowe STORE, FETCH) następuje przejście do mikrorozkazu MA2. W chwili przejścia do rejestru adresowego procesora AR, wpisywana jest zawartość rejestru adresowego pulpitu ARP. Słowo z rejestru ARP przesyłane jest do AR za pośrednictwem szyny CPB,
- dla $KP=3$, następuje przejście do mikrorozkazu MA3, wykonującego operację SET,
- $KP=4$ powoduje przejście do mikrorozkazu MA4, wykonującego operację PUT,
- $KP=5$ powoduje przejście do mikrorozkazu MA5, wykonującego operację GET,
- $KP=6$ jest sygnałem START, powodującym przejście do mikrorozkazu MC1, rozpoczynającego fazę pobrania instrukcji o adresie określonym aktualną zawartością licznika instrukcji,
- gdy $KP=7$ (operacja CONTINUE), następuje przejście do mikrorozkazu MC5, rozpoczynającego wykonanie ostatnio pobranej instrukcji.

Ponadto w mikrorozkazie MA1 zerowane są przerzutniki przerw programowych (tabela 11) oraz zapamiętywane są zgłoszenia pozostałych przerw (sygnał INCK - tabela 10). Przy pojawieniu się sygnału INT z układu przerw następuje przejście do mikrorozkazu MB2 fazy rejestracji przerw.

MA2

Dla $KP=1$ mikrorozkaz MA2 wykonuje operację STORE. Mikroopera-

oja $MRD(03,AR)$ oznacza, że do koordynatora kanałów wysyłane jest zgłoszenie przydziału szyny głównej, $CERN0=0$. Po nadejściu zezwolenia, $GRNO=0$, na liniach MOPN umieszczony jest kod operacji odczytu słowa ($MOP=03$), a na liniach adresowych AR negacja słowa z rejestru adresowego (ustawiona w MA1). Na linie informacyjne IFN wprowadzane jest słowo przesłane za pośrednictwem szyny CPB z rejestru kluczy pulpitu RK.

Dla $MP=2$ mikrorozkaz MA2 wykonuje operację FENCH. Wysyłany jest sygnał $CERN0=0$; po otrzymaniu sygnału $GRNO=0$ wprowadzany jest na linie MOPN kod operacji odczytu słowa, a na liniach AR umieszczona jest zawartość rejestru adresowego AR. Po otrzymaniu sygnału OUCN słowo pobrane z pamięci znajdującej się na liniach IFN, przesyłane jest do pulpitowego rejestru kluczy RK za pośrednictwem szyny CPB.

Mikrorozkaz MA2 trwa do czasu pojawienia się sygnału $GRC = GRNO \cdot OUCN$. Wykonanie mikrorozkazu kończy zwiększenie o 1 liczby reprezentowanej słowem w rejestrach AR i AR. Po wykonaniu MA2 następuje powrót do stanu WAIT.

MA3

W ramach mikrorozkazu MA3 wykonywany jest wpis zawartości rejestru kluczy do pulpitowego rejestru adresowego:

$ARF : = RK$

Po wykonaniu MA3 następuje powrót do stanu WAIT.

MA4

Mikrorozkaz MA4 przesyła zawartość rejestru kluczy RK do rejestru $B(i)$ o numerze i określonym zawartością pulpitowego rejestru numerów RFP. Słowo z rejestru RK przesyłane jest do $B(i)$ za pośrednictwem szyny CPB.

Po wykonaniu następuje powrót do stanu WAIT.

MA5

Mikrorozkaz MA5 przesyła do rejestru kluczy RK za pośrednictwem szyny CPB, zawartość rejestru R(i) o numerze i określonym przez zawartość pulpitowego rejestru numerów RFP.

Po wykonaniu MA5 następuje powrót do stanu WAIT.

2.4.2. Mikroprogram rejestracji przerwania: MB1,...,MB9

Mikrorozkazy MB1,...,MB9 tworzą mikroprogram rejestracji przerwania. Ponadto mikrorozkazy MB5 i MB6 wykorzystywane są jako faza wykonania instrukcji powrotu z przerwania (sygnał RT=1):

00x 4x1

Rejestrację przerwania rozpoczyna mikrorozkaz MB1 bądź MB2, w zależności od rodzaju przerwania oraz od mikrorozkazu, po którym następuje przejście do obsługi przerwania.

Rejestracja przerwania ADE rozpoczyna się bezpośrednio po mikrorozkazie w którym zostało stwierdzone przekroczenie adresu (może to być mikrorozkaz MC2,MD2,MD5,MD7 i MD9). Pierwszym wykonywanym mikrorozkazem jest dla tego przerwania MB1 z wyjątkiem przypadku, gdy przekroczenie adresu wykryte zostało w mikrorozkazie MD7, a wykonywana była jedna z instrukcji:

30x xxx

31x xxx

32x xxx

33x xxx

W tym przypadku rejestrację przerwania ADE rozpoczyna mikrorozkaz MB2. Ponieważ w przypadku przerwania ADE zainicjowana została praca pamięci, to cykl współpracy jest podtrzymywany w czasie mikrorozkazów MB1 i MB2 przy czym zleconą operację pamięci będzie odczyt słowa (MOP=07). Mikrorozkaz MB2 będzie zakończony w takim przypadku po zakończeniu współpracy z pamięcią.

Przerwania PFL, RES, ORQ, TIM, TRP, CHM, CH1, CH2, CH3 zgłoszone do układu przerwań gdy procesor jest w stanie pracy (RUN), rejestrowane są po najbliższym wykonaniu mikrorozkazu MC2, jeśli spełnione są następujące warunki:

- przerzutnik TMP=0,
- przerzutnik I rejestru stanu ustawiony jest w stan 1 (warunek ten nie dotyczy przerwania TRP),
- ponadto dla przerwań z kanałów oraz przerwań TIM, ORQ przerzutnik U rejestru stanu ustawiony jest w stan 1.

Przejście z mikrorozkazu MC2 do mikrorozkazu MB1, rozpoczynającego rejestrację przerwań omawianej grupy, następuje pod wpływem sygnału INT=1, ustawianego na początku mikrorozkazu MC2 w przypadku, gdy oczekuje na obsługę jedno z wymienionych przerwań i spełnione są podane wyżej warunki.

Przerzutnik TMP ustawiany jest w stan 1 w mikrorozkazie MC2, jeśli ostatnio wykonaną instrukcją była instrukcja ustawiania bitów M, P rejestru stanu:

377 737

bądź instrukcja powrotu z przerwania

00x 4x1.

W takiej sytuacji ewentualne przerwanie z omawianej grupy będzie rejestrowane dopiero po wykonaniu następnej instrukcji (przerzutnik TMP ustawiany jest w stan 0 w czasie mikrorozkazu MC5).

Stan 0 przerzutnika I maskuje wszystkie przerwania omawianej grupy z wyjątkiem TRP, a stan 0 przerzutnika U - przerwania z kanału oraz przerwania ORQ i TIM.

Przejście do rejestracji przerwania BSC lub SVC następuje po mikrorozkazie MC5, gdy zdekodowana zostanie jedna z instrukcji

(00x xxx) - \overline{IR}_6

Rejestrację tych przerwań rozpoczyna mikrorozkaz MB1.

Zdekodowanie w mikrorozkazie MC5 instrukcji nielegalnej, bądź wykrycie kodu nie odpowiadającego żadnej instrukcji powoduje przejście do mikrorozkazu MB2 - rejestrowane jest przerwanie INE.

Rejestracja przerwania FPE rozpoczyna się po mikrorozkazie MF2, jeśli w czasie wykonywania tego mikrorozkazu zasygnalizowany został błąd operacji zmiennoprzecinkowej (FPE1=1), albo po mikrorozkazie MC5 po zdekodowaniu instrukcji

$(10x \text{ xxx}) \cdot IR_4$

przy braku jednostki zmiennoprzecinkowej. Pierwszym mikrorozkazem dla przerwania FPE jest mikrorozkaz MB2.

MB1

Mikrorozkaz MB1 rozpoczyna mikroprogram rejestracji dla przerwania: BSC, SVC, ADE (z wyjątkiem wymienionego już wyżej przypadku) oraz dla zgłoszonych w stanie RUN procesora przerwania: PFL, RES, ORQ, TIM, TRP, CHM, CH1, CHS, CH3.

W przypadku przerwania innych niż BSC i SVC mikrorozkaz MB1 wpisuje do licznika instrukcji zapamiętany w rejestrze BR adres części operacyjnej instrukcji ostatnio pobranej (adres ten umieszczony został w rejestrze BR w czasie mikrorozkazu MC1).

W przypadku przerwania BSC lub SVC do licznika instrukcji wpisywana jest zwiększona o 1 zawartość rejestru BR, czyli adres następnej instrukcji po instrukcji BSC lub SVC która spowodowała przerwanie. Jest to adres powrotu do przerwanej programu.

Jak już wyjaśniono, w przypadku gdy przejście do MB1 następuje po jednym z mikrorozkazów MC2, MD2, MD5, MD7 bądź MD9, w chwili wejścia do MB1 jest sygnał żądania współpracy z Centralnym Łącznikiem Systemu CBRNO = 0. W takiej sytuacji sygnał CBRNO = 0 jest podtrzymywany w czasie wykonywania mikrorozkazu MB1 aż do otrzymania sygnału zezwolenia GRNO = 0 z koordynatora kanałów; jeśli

natomiast zezwolenie koordynatora nie nadejdzie w czasie trwania MB1, sygnał CBRNO = 0 podtrzymywany jest również w czasie następnego mikrorozkazu, którym jest MB2.

Po otrzymaniu sygnału GRNO = 0, na liniach MOPN ustawiany jest kod operacji odczytu słowa (07).

MB2

Przejście do mikrorozkazu MB2 następuje:

- po mikrorozkazie MB1,
- ze stanu WAIT (mikrorozkaz MA1) po zgłoszeniu przerwania powodującego powstanie sygnału INT=1,
- po mikrorozkazie MD7, jeśli stwierdzone zostało przekroczenie adresu, a wykonywana była jedna z instrukcji:

BNX XXX dla N=0,1,2,3

- po mikrorozkazie MC5 w przypadku przerwania INE, albo FPE,
- po mikrorozkazie MF2 w przypadku przerwania FPE.

Jeśli w chwili przejścia do MB2 obecny jest sygnał CBRNO = 0 bądź sygnał GRNO = 0, co świadczy o tym, że w czasie jednego z poprzednich mikrorozkazów zainicjowana została współpraca z pamięcią, wstrzymywany jest przebieg zegarowy procesora SXP, aż do otrzymania strobu OUCN z pamięci (GRC).

Mikrorozkaz MB2 powoduje wyzerowanie rejestru numeru kanału CHR w układzie przerwań, oraz ustawienie wyjść CS_{3:0}, ICS priorytetowego kodera przyczyny przerwania (tab.12).

Jeśli rejestrowanym przerwaniem jest przerwanie z kanału, po wyzerowaniu rejestru numeru kanału priorytetowy koder przyczyny przerwania sygnalizuje brak zgłoszeń, ustawiając ICS=0.

Wpis numeru kanału zgłaszającego przerwanie oraz, co za tym idzie, ustawienie przyczyny przerwania na wyjściach kodera, następuje w ramach mikrorozkazu MB4, wykonywanego po MB2 w przypadku

przerwań z kanałów (ICS=0).

Po mikrorozkazie MB2 następuje, w zależności od rejestrowanego przerwania, mikrorozkaz MB3, MB4, bądź MB5.

- Gdy na wyjściach priorytetowego kodera przyczyny przerwania ustawiany jest kod przerwania ORQ (CS=06) następuje przejście do mikrorozkazu MB3.
- Jeśli w chwili zakończenia mikrorozkazu MB2 koder przyczyny przerwania sygnalizuje brak zgłoszeń (sygnał ICS=0) następuje przejście do mikrorozkazu MB4.
- W przypadku gdy ICS=1, a kod przyczyny przerwania ustawiony na wyjściach kodera priorytetowego, jest różny od kodu przerwania ORQ (CS≠06), a więc gdy rejestrowane jest jedno z przerwania: BSC, SVC, INE, ADE, FPE, PFL, RES, TIM, TRP, następnym mikrorozkazem jest MB5.

MB3

Mikrorozkaz MB3 następuje po MB2 w przypadku przerwania ORQ. Mikrorozkaz ten wpisuje do rejestru RQ specyfikację przerwania pulpitu. Specyfikacja pobierana jest z rejestru kluczy RK i przesyłana szyną CPB z pulpitu.

Po mikrorozkazie MB3 wykonywany jest mikrorozkaz MB5.

MB4

Mikrorozkaz MB4 wykonywany jest po MB2 gdy rejestrowane jest przerwanie z kanału. Zadaniem mikrorozkazu MB4 jest pobranie z kanału specyfikacji przerwania.

Zależności czasowe przy wykonywaniu mikrorozkazu MB4 pokazano na rys. 2.5. Cykl współpracy z kanałem z którego wysłane było zadanie CHIN (CBRN₁, IEN₁) rozpoczyna się od wysłania sygnału zezwolenia na przerwanie PGN=0. Przebieg zegarowy procesora SXP jest

45

wstrzymywany aż do nadejścia strobu TRCHN z kanału. Po otrzymaniu strobu TRCHN, specyfikacja przerwania umieszczona przez kanał na liniach informacyjnych Łacza CLS wpisywana jest do rejestru RQ.

Ponadto w rejestrze numeru kanału CHR zapamiętywany jest stan linii GRN_1 . W wyniku wpisu do rejestru numeru kanału, na wyjściach kodera CS pojawi się numer przerwania odpowiadający numerowi kanału który zgłosił przerwanie.

Po zakończeniu mikrorozkazu MB4 następuje przejście do mikrorozkazu MB5.

MB5

Przejście do mikrorozkazu MB5 następuje w czasie rejestracji przerwania (RT=0) z mikrorozkazu MB2, MB3, albo MB4, bądź przy wykonywaniu instrukcji 00x 4x1 powrotu z przerwania - z mikrorozkazu MC5 (RT=1).

Mikrorozkaz MB5 powoduje wpisanie do rejestru adresowego AR słowa z zerowego miejsca pamięci:

$AR := S(0)$

AR jest adresem pola pamięci w którym zapamiętany zostanie stan procesora przy przerwaniu albo z którego odtworzony będzie stan procesora przy powrocie z obsługi przerwania.

Ponadto licznik CA ustawiany jest na wartość początkową

$CA := 06$.

Następnym mikrorozkazem po MB5 będzie mikrorozkaz MB6.

MB6

Przejście do mikrorozkazu MB6 następuje z mikrorozkazu MB5. Mikrorozkaz MB6 wykonywany jest 12 razy dla kolejnych wartości $CA = 06, 07, 10, \dots, 17$.

Przy rejestracji przerwania (RT=0) do kolejnych miejsc pamięci $S(0), S(0)+1, S(0)+2, \dots$ przesyłana jest zawartość licznika okre-

sów czasowych TM, rejestru stanu $SR_{15:0}$, licznika instrukcji $R(10)$ oraz rejestrów $R(11), \dots, R(17)$.

Przy instrukcji powrotu z obsługi przerwania ($RT=1$) słowa z kolejnych miejsc pamięci $S(0), S(0)+1, S(0)+2, \dots$ umieszczane są kolejno w liczniku TM, rejestrze stanu $SR_{15:0}$, liczniku instrukcji $R(10)$ oraz w pozostałych rejestrach $R(11), \dots, R(17)$.

Każde wykonanie mikrorozkazu MB6 rozpoczyna się wysłaniem żądania dostępu do szyny CLS $CBRN0=0$. Po otrzymaniu zezwolenia z koordynatora kanałów $GRN0=0$, ustawiany jest na liniach MOPN kod operacji pamięci (03 dla $RT=0$ (zapis słowa), 07 dla $RT=1$ (odczyt słowa)), a na liniach adresowych ARN, umieszczana jest aktualna zawartość rejestru adresowego AR. Rejestr adresowy AR, zawiera w czasie pierwszego wykonania mikrorozkazu MB6 ustawiony przez mikrorozkaz MB5 adres początkowy pola pamięci przeznaczonego na przechowanie zawartości rejestrów $S(0)$. Przy każdym wykonaniu MB6 zawartość rejestru adresowego zwiększana jest o 1.

Każde wykonanie mikrorozkazu MB6 zwiększa o 1 zawartość licznika CA. W czasie pierwszego wykonania $CA=06$ (stan ten został ustawiony przez mikrorozkaz MB5). Ten stan licznika CA powoduje, że informacja przesyłana będzie pomiędzy TM a pamięcią. Przy drugim wykonaniu MB6 gdy $CA=07$ informacja przesyłana będzie pomiędzy SR i pamięcią, a dla następnych wykonań gdy $10 \leq CA \leq 17$ -- pomiędzy rejestrami $R(CA)$ a pamięcią.

Gdy stan licznika $CA=17$ z mikrorozkazu MB6 następuje przejście do MB7 gdy $RT=0$, a do MC1 gdy $RT=1$.

MB7

Mikrorozkaz MB7 następuje po MB6, gdy wykonywana jest rejestracja przerwania. Zadaniem mikrorozkazu MB7 jest wpisanie kodu przyczyny przerwania CS do rejestru $R(16)$. Kod ten, pobierany

z wyjść priorytetowego kodera przyczyny przerwania, jest przesyłany do rejestru R(16) za pośrednictwem szyny głównej procesora CPB.

Po MB7 następuje mikrorozkaz MB8.

MB8

Mikrorozkaz MB8 wpisuje do rejestru R(17) zawartość rejestru RQ. Gdy rejestrowane jest przerwanie wymagające specyfikacji, słowo zapamiętane w rejestrze RQ, przesyłane do R(17), jest właśnie specyfikacją obsługiwanego przerwania.

- W przypadku przerwania BSC, SVC oraz FPE przy braku jednostki zmiennoprzecinkowej specyfikacją jest słowo kodujące instrukcję.
- W przypadku przerwania ORQ specyfikacją jest stan rejestru kluczy RK na pulpicie procesora.
- W przypadku przerwania z kanałów specyfikacja przesyłana jest z kanałów.
- W przypadku przerwania FPE specyfikacja jest przesyłana z jednostki zmiennoprzecinkowej (jeśli jest).

W pozostałych przypadkach zawartość RQ jest obojętna.

Po mikrorozkazie MB8 wykonywany jest mikrorozkaz MB9.

MB9

Mikrorozkaz MB9 pobiera z pamięci operacyjnej adres początku programu obsługi przerwania; adres ten wpisywany jest do rejestrów BR i RQ z miejsca pamięci o adresie 1.

Wykonanie mikrorozkazu MB9 rozpoczyna się wysłaniem żądania dostępu do szyny głównej systemu CBRN0=0. Gdy koordynator kanału prześle zezwolenie GRN0=0, na liniach MOPN ustawiany jest kod operacji odczytu słowa (07). Na linie adresowe, ARN wprowadzany jest adres 1, w którym zapamiętany jest początkowy adres programu obsługi przerwania. Przebieg zegarowy SXP wstrzymywany jest aż do nadejścia strobu OUCN. Po otrzymaniu tego strobu, słowo

odczytane z pamięci wpisywane jest do rejestrów BR i RQ.

W czasie trwania MB9 zerowany jest przerzutnik rejestru zgłoszeń przerwania zewnętrznych (tab.10) odpowiadający obsługiwanemu przerwaniu. Ponadto w rejestrze stanu na pozycje S,A,B,D wpisywane są jedynki, a na pozostałe pozycje zera. Zerowany jest również przerzutnik TMP.

Następnym mikrorozkazem po MB9 jest MC2.

2.4.3. Mikrorozkaz MCH przesyłania rozkazu do kanałów we-wy

Mikrorozkaz MCH jest fazą wykonawczą instrukcji wejścia-wyjścia. Przejście do niego następuje po mikrorozkazie MC5 dla instrukcji

00x 50x

Zadaniem mikrorozkazu MCH jest przesłanie słowa zapisanego w rejestrze R(CA), do wskazanego kanału (numer kanału określony jest przez bity 0, 1 słowa instrukcyjnego IR) oraz pobranie odpowiedzi z kanału.

2.6

Wykonanie mikrorozkazu MCH (rys.2.6) rozpoczyna się wysłaniem żądania dostępu do szyny głównej CBRN0=1, IRN0=0 i zgłoszenia procesora do kanału o numerze $i=IR_1:0$ $CHN_i=0$. Ponadto na linię TESTN wysyłany jest \overline{IR}_2 (rozkaz testowania kanału gdy TESTN=0).

Po otrzymaniu zezwolenia z koordynatora kanałów GRN0=0, na linii informacyjnej IFN przekazywana jest zawartość rejestru R(CA). Wysyłany jest do kanału strob TRPN. Po pojawieniu się strobu TRCHN, z linii informacyjnych IFN pobierana jest odpowiedź kanału; zostaje ona wpisana do rejestru R(CA). Bity 1,0 odpowiedzi wpisane zostają ponadto odpowiednio do przerzutników X,Y rejestru stanu.

Mikrorozkaz MCH jest mikrorozkazem asynchronicznym - przebieg zegarowy SXP blokowany jest aż do nadejścia strobu TRCHN.

Następnym mikrorozkazem po MCH jest mikrorozkaz MC1.

2.4.4. Mikrorozkazy fazy pobierania instrukcji MC1,...,MC5

Mikrorozkazy MC1,...,MC5 tworzą fazę pobrania instrukcji.

MC1

Mikrorozkaz MC1 umieszcza w rejestrach BR i RQ programowy adres instrukcji, która ma być pobrana do wykonania z pamięci operacyjnej.

Przejście do mikrorozkazu MC1 następuje:

- z mikrorozkazu MA1 (stan WAIT) po naciśnięciu kluczy IPL albo START (sygnał OPN i $KP=0,6$),
- z mikrorozkazu MC5 dla instrukcji

$(00x\ 51x) \cdot \overline{IR}_2$	377 4x7
00x 514	377 5x7
$(00x\ 5xx) \cdot \overline{IR}_4$	377 6x7
077 xxx	$(377\ 737) \cdot \overline{SR}_M \cdot \overline{SR}_P$
$(377\ xx7) \cdot \overline{IR}_7 \cdot \overline{SR}_S$	377 737

oraz dla 16-bitowych instrukcji skoków warunkowych przy niespełnionym warunku JUW (skoki krótkie nieefektywne)

$(05x\ xxx) \cdot \overline{JUW}_5$	$(37x\ x05)$
$(06x\ xxx) \cdot \overline{JUW}_6$	$(37x\ x06)$
$(07x\ xxx) \cdot \overline{JUW}_7$	$(37x\ x07)$

- po ostatnim mikrorozkazie fazy wykonania instrukcji za wyjątkiem instrukcji skoków efektywnych.

Jeżeli przejście do MC1 nastąpiło ze stanu WAIT przy $KP=6$ (operacja START), bądź gdy ostatnio wykonaną instrukcją była instrukcja powrotu z przerwania ($RT=1$), do rejestrów BR i RQ wpisywana jest, jako adres instrukcji która ma być pobrana, aktual-

na zawartość licznika instrukcji. W pozostałych przypadkach załadowana zostaje do rejestrów BR i RQ zawartość licznika instrukcji zwiększona o 1.

Po mikrorozkazie MC1 następuje zawsze mikrorozkaz MC2.

MC2

W ramach tego mikrorozkazu adres instrukcji znajdujący się w rejestrze RQ (a także w rejestrze BR) jest relokowany według trybu 1, po czym zostaje wpisany do rejestru adresowego AR. Układ kontroli adresu sprawdza, czy relokowany adres nie wykracza poza obszar pamięci ograniczony wielkościami

$$DT(12) = 256 \cdot R_{15:8}(12)$$

$$LT(12) = 256 \cdot R_{7:0}(12)$$

(dla $SR_5=0$). Stwierdzenie przekroczenia adresu powoduje powstanie sygnału $WADE=ADE1=1$ oraz ustawienie w stan 1 przerzutnika ADE w rejestrze zgłoszeń przerw.

Ze względu na występujące w mikrorozkazie MC4 odwołanie do pamięci, układ współpracy z Centralnym Łączem systemu ustawia w czasie trwania mikrorozkazu MC2 sygnał $CBRN0=0$, stanowiący żądanie dostępu do GLS (zob. mikrorozkaz MB2).

Mikrorozkaz MC2 powoduje zapis ewentualnych zgłoszeń przerw do rejestru przerw (zob. tab.10 i mikrorozkaz MA1) oraz zeruje przerzutniki SR_M i SR_P gdy $TMP=0$ (ostatnio wykonana instrukcja nie ustawiała SR_M i SR_P).

Wejście do mikrorozkazu MC2 następuje:

- po mikrorozkazie MC1;
- po mikrorozkazie MB9, kończącym mikroprogram rejestracji przerw - mikrorozkaz MB9 umieszcza w rejestrach BR i RQ adres początkowy programu obsługi przerwania;
- po mikrorozkazach MJ1, MJ4, MD6 oraz MC5 w których może być wyz-

naczany adres skoku dla instrukcji skoków efektywnych. W dwóch ostatnich przypadkach fazę pobrania nowej instrukcji rozpoczyna mikrorozkaz MC2 (mikrorozkaz MC1 jest pomijany).

Po mikrorozkazie MC2 wykonywany jest mikrorozkaz MC3 lub mikrorozkaz MB1. Przejście do mikrorozkazu MB1, rozpoczynającego mikroprogram rejestracji przerwań, następuje w przypadku wystąpienia sygnału przerwania INT:

$$\text{INT} = \text{WADE} \vee \overline{\text{PPLZ}} \vee \overline{\text{RESZ}} \vee \overline{\text{ORQZ}} \vee \overline{\text{TIMZ}} \vee \overline{\text{CHIZ}} \vee \overline{\text{TRZ}}$$

W przeciwnym przypadku po mikrorozkazie MC2 następuje mikrorozkaz MC3.

MC3

Mikrorozkaz MC3 wpisuje do licznika instrukcji znajdujący się w rejestrze RQ (i w rejestrze BR) adres programowy instrukcji, która będzie pobrana do wykonania.

W czasie trwania mikrorozkazu MC3 procesor może otrzymać z koordynatora kanałów sygnał przydziału szyny głównej (GRNO) jako odpowiedź na ustawiony w czasie mikrorozkazu MC2 i podtrzymywany po przejściu do MC3 sygnał żądania dostępu do szyny. Po pojawieniu się sygnału przydziału szyny głównej, układ współpracy z szyną CLS wycofuje sygnał CBRNO i ustawia na liniach kodu operacji pamięci (MOPN) kod odczytu słowa z pamięci (07) a na liniach ARN - zawartość rejestru adresowego.

Wejście do mikrorozkazu MC3 możliwe jest tylko z mikrorozkazu MC2. Po mikrorozkazie MC3 wykonywany jest zawsze mikrorozkaz MC4.

MC4

Mikrorozkaz MC4 pobiera słowo instrukcyjne z pamięci operacyjnej. Jest to mikrorozkaz asynchroniczny: czas jego trwania uzależniony jest od czasu oczekiwania na przesłanie słowa z pamięci.

Jeżeli w czasie trwania mikrorozkazu MC3 nie pojawił się sygnał przydziału szyny głównej (GRNO), podtrzymywany jest po wejściu

52

ciu do MC4 sygnał żądania dostępu (CERN0) aż do otrzymania sygnału przydziału, po czym ustawiany jest kod operacji odczytu słowa na liniach MOPN i adres, według zawartości rejestru adresowego, na liniach ARN.

Jeżeli sygnał GRN0 został przyjęty już w czasie trwania mikrorozkazu MC3, po wejściu do MC4 układ współpracy z szyną główną utrzymuje na liniach MOPN i ARN kod operacji i adres, ustawione w czasie mikrorozkazu MC3.

Przebieg zegarowy procesora wstrzymywany jest aż do otrzymania strobu OUCN. Po pojawieniu się strobu OUCN słowo instrukcyjne przesyłane po liniach IFN_{15:0}, wpisywane jest do rejestru instrukcji, do rejestru RQ oraz do rejestru kluczy pulpitu RK. Na liniach IFN znajduje się zanegowane słowo instrukcyjne; do rejestru instrukcji zostaje więc wpisane (poprzez wyjścia BQ przełącznicy ABQ) słowo instrukcyjne w postaci prostej, a do rejestru RQ - w postaci zanegowanej.

Ponadto do licznika numeru rejestru CA zapisywane są bity 8, 9, 10 słowa instrukcyjnego (na pozycję CA₃ wpisywane jest zero).

Mikrorozkaz MC4 następuje po mikrorozkazie MC3. Po zakończeniu mikrorozkazu MC4 możliwe jest przejście do stanu WAIT (MA1), bądź do mikrorozkazu MC5. Stan WAIT następuje po mikrorozkazie MC4 jeżeli z pulpitu przesyłany jest sygnał PK8=1 (wciśnięty klucz STEP - praca krokowa) a ponadto TMP=0. W przeciwnym wypadku wykonywany jest po MC4 mikrorozkaz MC5.

MC5

Przejście do mikrorozkazu MC5 może nastąpić z mikrorozkazu MC4 bądź z mikrorozkazu MA1 (ze stanu WAIT) po naciśnięciu klucza CONTINUE na pulpicie.

Czynności wykonywane w ramach mikrorozkazu MC5 zależą od pob-

ranej instrukcji. Ogólnie można powiedzieć, że rola mikrorozkazu MC5 polega na zdekodowaniu instrukcji oraz umieszczeniu w rejestrze RQ bądź BR informacji wykorzystywanej przez późniejsze mikrorozkazy do wyznaczenia argumentu, bądź do wykonania instrukcji. Ponadto mikrorozkaz MC5 ustawia wstępnie zawartość licznika kroków CN oraz stany przerzutników pomocniczych (warunków) AS, DS, TV, TMP, NQ, RT.

Istnieje grupa instrukcji dla których mikrorozkaz MC5 stanowi fazę wykonania, są to instrukcje związane z ustawieniem bądź odczytywaniem zawartości rejestru stanu oraz instrukcji krótkich warunkowych skoków nieefektywnych dla których z MC5 następuje przejście do MC1, a także instrukcje krótkich skoków efektywnych z argumentem w rejestrze R(B), dla których z MC5 następuje przejście do MC2.

Omówione zostaną teraz dokładniej czynności wykonywane w ramach mikrorozkazu MC5 i następujące po nim przejścia dla poszczególnych grup instrukcji. Należy pamiętać, że zgodnie z tym co powiedziano przy opisie mikrorozkazu MC4, w chwili wejścia do MC5 w rejestrze RQ znajduje się zanegowane słowo instrukcyjne $RQ = \overline{IR}$.

- W przypadku instrukcji

$(00x\ xxx) \cdot \overline{IR}_7$

zanegowana zostaje zawartość rejestru RQ. W rejestrze RQ znajduje się więc po zakończeniu mikrorozkazu MC5 słowo instrukcyjne w postaci nie zanegowanej $RQ = IR$. Ponadto ustawiane są przerzutniki BSC i SVC rejestru zgłoszeń przerw programowych tak, że

BSC=1 dla instrukcji $\begin{pmatrix} 00x\ 0xx \\ 00x\ 1xx \end{pmatrix}$

SVC=1 dla instrukcji $\begin{pmatrix} 00x\ 2xx \\ 00x\ 3xx \end{pmatrix}$

Po mikrorozkazie MC5 następuje przejście do mikrorozkazu MC1

rozpoczynającego rejestrację przerwania BSC lub SVC.

- W przypadku instrukcji przesuwania słów w rejestrach

00A 6xx
00A 7xx

zostaje załadowana do rejestru BR zawartość rejestru R(CA).

Jeżeli pobrana instrukcja jest instrukcją przesuwania zawartości jednego rejestru tzn. gdy $IR_{4:3} \neq 3$, po mikrorozkazie MC5 następuje przejście bezpośrednio do mikrorozkazu MS2, przesuwającego zawartość rejestru BR. Jeśli natomiast ma być wykonana instrukcja łącznego przesuwania zawartości dwóch rejestrów, czyli gdy $IR_{4:3} = 3$, następuje przejście do mikrorozkazu MS1, w celu wpisania do rejestru RQ zawartości rejestru R(CA+1) przed mikrorozkazem MS2, który w tym przypadku przesuwa łącznie zawartość BR i RQ.

- W przypadku instrukcji:

01x xxx
02x xxx
03x xxx
(2xx xxx) · ($IR_{13:8} \neq 00, 01$)

liczba dodatnia reprezentowana przez mniej znaczący bajt słowa instrukcyjnego stanowi adres argumentu. Dla tych instrukcji rola mikrorozkazu MC5 polega na umieszczeniu tej liczby w rejestrze RQ. Tak uatawiona zawartość RQ wykorzystywana jest przez mikrorozkazy pobierające lub zapisujące argument. Dla instrukcji

01x xxx
02x xxx

następnym po MC5 mikrorozkazem jest MD7, relokujący zapisany w RQ adres przed następującym w mikrorozkazie MD8 pobraniem argumentu do rejestru R(CA), bądź zapisaniem zawartości R(CA) pod wskazany adres. Natomiast dla instrukcji:

03x xxx
(2xx xxx) · ($IR_{13:8} \neq 00, 01$)

kolejnym po MC5 mikrorozkazem jest MD5, relokujący zawarty w RQ adres, przed pobraniem argumentu z pamięci.

- W przypadku instrukcji skoków

- 04x xxx
- 05x xxx
- 06x xxx
- 077 xxx

mniej znaczący bajt słowa instrukcyjnego reprezentuje w kodzie U2 względny adres skoku. Adres ten może być dodatni lub ujemny. Mikrorozkaz MC5 umieszcza tę liczbę w rejestrze RQ - zależnie od jej znaku, określonego pozycją IR₇, bardziej znaczący bajt w RQ uzupełniany jest zerami bądź jedynkami. Dla instrukcji

- 04x xxx

oraz gdy spełnione są warunki efektywności skoku

- (05x xxx)·JUW5
- (06x xxx)·JUW6
- (077 xxx)·JUW7

z mikrorozkazu MC5 następuje przejście do mikrorozkazu MJ1, a w przeciwnym przypadku (skoki nieefektywne) - do mikrorozkazu MC1, w którym rozpoczyna się faza pobrania następnej instrukcji.

- W przypadku instrukcji arytmetycznych

- 10x xxx

argument podany jest na mniej znaczącym bajcie instrukcji. Mikrorozkaz MC5 przekształca go na słowo 16-bitowe i umieszcza je w rejestrze BR. Dla instrukcji

- 11x xxx
- 12x xxx
- 13x xxx

pozycja IR₇ określa znak argumentu (w kodzie U2) tak więc na bardziej znaczącym bajcie rejestru BR umieszczane są zera gdy IR₇=0 albo jedynki gdy IR₇=1. Natomiast dla instrukcji

- 14x xxx
- 15x xxx
- 16x xxx
- 17x xxx

na bardziej znaczącym bajcie BR umieszczane są zera.

Dla instrukcji

11x xxx

po MC5 wykonywany jest mikrorozkaz MJ3, wpisujący ustawioną w MC5 zawartość rejestru BR do rejestru R(CA), natomiast dla pozostałych instrukcji tej grupy następnym po MC5 mikrorozkazem jest MH1, wykorzystujący zawartość rejestru BR jako argument odpowiedniej operacji logicznej lub arytmetycznej (drugim argumentem operacji jest zawartość R(CA)).

- W przypadku instrukcji

10x xxx
30x xxx
31x xxx
32x xxx
33x xxx

wymagane jest sumowanie zawartości rejestru R(B) i liczby (dodatniej lub ujemnej), reprezentowanej w kodzie U2 przez cztery najmniej znaczące bity słowa instrukcyjnego. Liczba ta umieszczana jest przez mikrorozkaz MC5 w rejestrze RQ; zależnie od znaku liczby (IR_3), bity $RQ_{15:4}$ zapełniane są zerami lub jedynkami. Wspomniane sumowanie wykonywane jest w mikrorozkazie ME1, następującym po MC5 w przypadku wymienionych instrukcji.

- Argumentem instrukcji

34x x0x
35x x0x

jest zawartość rejestru R(B). Mikrorozkaz MC5 umieszcza tę zawartość w rejestrze RQ. Po mikrorozkazie MC5 następuje przejście do wykonania przesłania blokowego z pola pamięci lub do pola pamięci o ustawionym w RQ adresie początkowym. Pierwszym wykonywanym po MC5 mikrorozkazem jest MD7, dokonujący relokacji adresu.

- Argumentem instrukcji

36x x00
37x x00
37x x10

jest zawartość rejestru R(B). Wpisywana jest ona w czasie mikrorozkazu MC5 do rejestru BR; dla mikrorozkazu MH1, następującego po MC5 w przypadku tych instrukcji, zawartość BR jest argumentem odpowiedniej operacji arytmetycznej lub logicznej (drugim argumentem operacji jest zawartość rejestru R(CA)).

-- W przypadku instrukcji

36x x1x z wyjątkiem $\begin{pmatrix} 360 \text{ x10} \\ 361 \text{ x10} \end{pmatrix}$

zawartość rejestru R(B) jest adresem argumentu. Mikrorozkaz MC5 umieszcza ten adres w rejestrze RQ. Następuje przejście do mikrorozkazu MD5, relokującego adres przed odwołaniem do pamięci, następującym w mikrorozkazie MD6.

-- W przypadku instrukcji mnożenia i dzielenia:

37x x20
37x x30

mnożna lub dzielna, zapisana w rejestrze R(A) umieszczana jest w czasie mikrorozkazu MC5 w rejestrze RQ. Rejestr BR oraz przerzutnik TV jest zerowany. W przypadku instrukcji dzielenia do przerzutnika AS wpisany zostaje bit znakowy dzielnej. Licznik kroków ustawiany jest w początkowym stanie tak aby $\overline{CN}=17$.

Jeśli wykonywane jest mnożenie, po mikrorozkazie MC5 następuje mikrorozkaz MP3; w przypadku dzielenia - mikrorozkaz MP1.

-- Jeżeli pobrana została jedna z instrukcji:

37x xx1

mikrorozkaz MC5 wpisuje zawartość rejestru R(B) do rejestru BR. Wykonywany po MC5 mikrorozkaz MG1 umieszcza z kolei zawartość rejestru R(CA) w rejestrze RQ, a późniejsze mikrorozkazy MG2 i MG3 ustawiające nową zawartość rejestrów R(A) i R(B), korzystają z zapamiętanej w rejestrach BR i RQ dotychczasowej zawartości tych rejestrów. W przypadku instrukcji 37x x31 mikrorozkaz MG1 jest pomijany: z MC5 następuje przejście bezpośrednio do MG2.

- W przypadku instrukcji:

37x x03
37x x04
37x x05
37x x06
 $\overline{377}$ x07

adres skoku znajduje się w rejestrze R(B). W czasie mikrorozkazu MC5 adres ten zostaje wpisany do rejestrów RQ i BR.

Dla instrukcji

37x x03
37x x04

następnym mikrorozkazem po MC5 jest MJ2. Natomiast dla instrukcji skoków efektywnych

(37x x05)·JUW5
(37x x06)·JUW6
($\overline{377}$ x07)·JUW7

następuje przejście do mikrorozkazu MC2, co rozpoczyna operację pobrania instrukcji o adresie wskazanym przez instrukcję skoku. Jeżeli natomiast odpowiedni warunek skoku JUW nie jest spełniony, po MC5 wykonywany jest mikrorozkaz MC1, co rozpoczyna fazę pobrania kolejnej instrukcji.

- Dla instrukcji

(00x 51E) dla E= 0,1,2,3
00x 52x
00x 53x
077 xxx
(377 Bx7) dla (B= 0,1,2,3 i $SR_S=1$ albo B= 4,5,6)
 $\overline{377}$ 737

wykonanie mikrorozkazu MC5 polega na ustawieniu wskazanych pozycji rejestru stanu. Pozycje SR_M i SR_P ustawiane są gdy do przetrzutnika TMP wpisywana jest jedynka:

$$TMP := (\overline{377} \overline{737}) \cdot \overline{SR_M} \cdot \overline{SR_P}$$

Instrukcja

377 737

53

jest instrukcją pustą ("nierób nic"). W przypadku instrukcji
00x 514

mikrorozkaz MC5 przesyła słowo $SR_{15:0}$ z rejestru stanu do rejestru
R(CA).

Mikrorozkaz MC5 stanowi fazę wykonania tych instrukcji; kolej-
nym mikrorozkazem jest MC1, rozpoczynający pobranie następnej ins-
trukcji.

- Jeżeli pobrana została legalna instrukcja XWT (WAIT)

00x 4x0 przy $SR_S=1$

po mikrorozkazie MC5 następuje przejście do mikrorozkazu MA1.

- W przypadku gdy pobrana została legalna instrukcja powrotu z
przerwania:

00x 4x1 przy $SR_S=1$

w MC5 ustawiany jest przerzutnik $RT:=1$ oraz $SR_T:=IR_4$, $SR_I:=IR_3$.
Następnym wykonywanym jest mikrorozkaz MB5. Mikrorozkazy MB5, MB6
wchodzące w skład mikroprogramu rejestracji przerwań, wykorzysty-
wane są również jako faza wykonania instrukcji powrotu z przerwa-
nia.

- W przypadku legalnej instrukcji:

00x 50x przy $SR_S=1$

po mikrorozkazie MC5 następuje przejście do mikrorozkazu MCH,
stanowiącego fazę wykonania tej instrukcji.

- Gdy pobrana instrukcja jest instrukcją operacji zmiennoprzecin-
kowej:

(10x x2x)
(10x x3x)

następnym po MC5 mikrorozkazem jest MF1, przesyłający argumenty
operacji do jednostki zmiennoprzecinkowej FPU. Jeśli brak jest
jednostki zmiennoprzecinkowej w procesorze (sygnał FPD), zdekodo-
wanie tej instrukcji powoduje zapalenie przerzutnika FPE rejestru

zgłoszeń przerwania, oraz przejście do mikrorozkazu MB2, który rozpoczyna rejestrację przerwania.

- Część adresowa instrukcji długich

- 34x $\overline{x0x}$
- 35x $\overline{x0x}$
- ($\begin{matrix} 36A & \overline{x20} \\ 36A & \overline{x30} \end{matrix}$) dla A= 2,3,4,5,6,7
- 36x $\overline{x20}$
- 36x $\overline{x30}$
- ($\begin{matrix} 37x & \overline{x0E} \end{matrix}$) dla E= 2,3,4,5,6
- 377 $\overline{x07}$

stanowi oddzielne słowo pamięci. Adres tego słowa jest o 1 większy od adresu części operacyjnej instrukcji. Gdy pobrana zostanie instrukcja długa, po MC5 następuje mikrorozkaz MD1 rozpoczynający fazę pobrania części adresowej z pamięci operującej. Warto zwrócić uwagę, że gdy instrukcja ta jest instrukcją skoku warunkowego, a więc jedną z instrukcji:

- 37x $\overline{x05}$
- 37x $\overline{x06}$
- 377 $\overline{x07}$

przejście do MD1 następuje również wtedy, gdy odpowiedni warunek skoku nie został spełniony. W tym przypadku jednak rola mikrorozkazu MD1 ogranicza się do zwiększenia o 1 zawartości licznika instrukcji przed fazą pobrania kolejnej instrukcji; ma to na celu ominięcie słowa pamięci operacyjnej zawierającego część adresową instrukcji skoku.

W przypadku instrukcji długich, mikrorozkaz MC5 pozostawia zawartość rejestrów BR i RQ bez zmian.

- Przerwanie INE (błędna instrukcja) powstaje gdy zdekodowana zostanie jedna z instrukcji:

- 00x 4x0
- 00x 4x1
- 00x 50x
- 377 Bx7 dla B=0,1,2,3

w tzw. stanie programu użytkowego tzn. gdy $SR_S=0$, instrukcja

$\overline{377} \overline{737}$ gdy $SR_M \vee SR_P = 1$

(tzw. instrukcje nielegalne) oraz jedno ze słów

00x 4xE dla E=2,3,4,5,6,7

00x 51E dla E=5,6,7

(20A xxx)
(36A xx0) dla A=0,1

36x x00

37x x02

nie reprezentujące żadnej instrukcji (tzw. instrukcje nieistniejące). W tych przypadkach sygnał WINE=1, ustawiany jest przerzutnik zgłoszenia przerwania

INE:=WINE

i następuje przejście do mikrorozkazu MB1 rozpoczynającego rejestrację przerwania.

Licznik kroków ustawiany jest następująco:

$\overline{CN}_{3:0} := (0, IR_{2:0})$ dla instrukcji $\begin{pmatrix} 34x \text{ xxx} \\ 35x \text{ xxx} \end{pmatrix}$

$\overline{CN}_{3:0} := 17$ dla instrukcji $\begin{pmatrix} 10x \text{ xDx} \\ 37x \text{ xD0} \end{pmatrix}$ przy D=2,3

dla pozostałych instrukcji stan licznika kroków jest obojętny.

2.4.5. Mikrorozkazy MD1, ..., MD9, ME1, ..., ME4, MJ1, ..., MJ4

Mikrorozkazy MD1, MD2, MD3 tworzą fazę pobrania części adresowej instrukcji długiej. Rola mikrorozkazu MD4 polega na modyfikacji pobranej części adresowej zawartością rejestru R(B).

MD1

Mikrorozkaz MD1 następuje po mikrorozkazie MC5 w przypadku instrukcji o długiej części adresowej:

34x $\overline{x0x}$

35x $\overline{x0x}$

(36A x20)
(36A x30) dla A=2,3,4,5,6,7

36x x20
36x x30
37x x0E dla E=2,3,4,5,6
377 x07

Mikrorozkaz MD1 wpisuje do rejestru RQ oraz do licznika instrukcji programowy adres części adresowej instrukcji długiej. Adres ten jest o 1 większy od adresu części operacyjnej, znajdującego się, w chwili wejścia do mikrorozkazu MD1, w rejestrze BR.

Po mikrorozkazie MD1 następuje mikrorozkaz MD2, z wyjątkiem przypadku, gdy wykonywana jest instrukcja skoku warunkowego przy niespełnionym warunku tzn. gdy:

(37x xx5) • $\overline{JUV5}$
(37x xx6) • $\overline{JUV6}$
(37x xx7) • $\overline{JUV7}$

W takiej sytuacji po mikrorozkazie MD1 następuje przejście do MC1. Mikrorozkaz MD1 jest w tym przypadku wykonywany tylko w celu zwiększenia o 1 zawartości licznika instrukcji, aby ominąć słowo zawierające część adresową instrukcji skoku.

MD2

Wejście do mikrorozkazu MD2 może nastąpić tylko z mikrorozkazem MD1.

W ramach mikrorozkazu MD2 adres znajdujący się w rejestrze RQ jest relokowany według trybu 1, a następnie zostaje zapisany do rejestru adresowego AR (zob. mikrorozkaz MC2). Układ kontroli adresu bada relokowany adres; przekroczenie adresu sygnalizowane jest przez ustawienie WADE=ADE1=1 oraz wpisanie ADE:=WADE.

Jeżeli nie zostało stwierdzone przekroczenie adresu, po mikrorozkazie MD2 następuje, zależnie od wykonywanej instrukcji, mikrorozkaz MD3 lub MD8. Z uwagi na występujące w obu tych mikrorozkazach odwołania do pamięci, w czasie mikrorozkazu MD2 ustawiony

zostaje sygnał CBRN0=0 (żądanie dostępu do pamięci). W przypadku instrukcji

37x x12

po mikrorozkazie MD2 następuje przejście do MD8 w celu pobrania do rejestru R(CA) części adresowej instrukcji, co kończy jej wykonanie. Przejście to następuje pod warunkiem, że nie jest sygnalizowane przekroczenie adresu. W przypadku pozostałych instrukcji po mikrorozkazie MD2 następuje, również pod warunkiem, że nie zostało stwierdzone przekroczenie adresu, mikrorozkaz MD3 pobierający z pamięci część adresową.

Jeśli układ badania adresu sygnalizuje przekroczenie wyznaczonego obszaru pamięci (ADE1=1), mikrorozkazem wykonywanym po MD2 jest MB1 rozpoczynający rejestrację przerwania ADE.

MD3

Wejście do mikrorozkazu MD3 może nastąpić tylko z mikrorozkazem MD2.

Zadaniem mikrorozkazu MD3 jest pobranie z pamięci operacyjnej do rejestru RQ części adresowej instrukcji długiej. Jest to mikrorozkaz asynchroniczny: czas jego trwania uzależniony jest od czasu oczekiwania na przydział dostępu do szyny głównej oraz na przesłanie pobieranego słowa z pamięci operacyjnej.

Sygnał CBRN0=0 ustawiony w czasie mikrorozkazu MD2, potrzynywany jest aż do otrzymania z koordynatora kanałów sygnału przydziału dostępu do szyny głównej GRN0=0. Po pojawieniu się sygnału GRN0, układ współpracy z szyną główną ustawia na liniach MOPN kod operacji odczytu słowa (MOP=07) a na liniach ARN - zawartość rejestru adresowego.

Przebieg zegarowy procesora SXP jest blokowany aż do nadejścia strobu informacji z pamięci OUCN. Po otrzymaniu strobu OUCN sło-

wo znajdujące się na liniach informacyjnych wpisane zostaje do rejestru RQ.

Po mikrorozkazie MD3 następuje, zależnie od wykonywanej instrukcji, mikrorozkaz MD4, MD5, MD7, MJ2 lub MJ4:

- w przypadku instrukcji

3xx x3x
37x x22

następuje przejście do MD4 w celu zmodyfikowania pobranej części adresowej zawartością rejestru R(B),

- w przypadku instrukcji

37x x2x
37x x22

pobrana przez mikrorozkaz MD3 część adresowa jest adresem argumentu. Następnym mikrorozkazem jest MD5, relokujący ten adres przed pobraniem argumentu.

- w przypadku instrukcji

34x x1x
35x x1x

część adresowa instrukcji jest adresem początkowym pola pamięci z którego lub do którego ma być wykonane przesłanie blokowe. Po mikrorozkazie MD3 następuje przejście do mikrorozkazu MD7, w ramach którego adres ten jest relokowany przed rozpoczęciem przesłania blokowego.

- w przypadku instrukcji

37x x13
37x x14

pobrana do RQ część adresowa jest adresem skoku. Następnym mikrorozkazem jest MJ2 w którym wykonywane są dodatkowe operacje związane z tymi instrukcjami.

- w przypadku instrukcji

37x x15
37x x16
37x x17

następuje przejście do mikrorozkazu MJ4 w którym znajdujący się w RQ adres skoku umieszczony jest w rejestrze BR.

MD4

Wejście do MD4 może nastąpić tylko z mikrorozkazu MD3. W MD4 dokonywana jest modyfikacja części adresowej umieszczonej w RQ zawartością rejestru R(B).

Dla instrukcji

37x xx2

zmodyfikowana część adresowa jest adresem słowa pamięci z którego ma być pobrana nowa zawartość rejestru R(CA), lub słowa pamięci do którego aktualna zawartość rejestru R(CA) ma być przesłana, a następnym mikrorozkazem jest MD7. Dla pozostałych instrukcji zmodyfikowana część adresowa jest adresem argumentu.

MD5

Mikrorozkazy MD5 i MD6 tworzą fazę pobrania argumentu instrukcji z pamięci, bądź fazę wykonania dla instrukcji modyfikujących lub testujących słowo w pamięci. Wejście do MD5 następuje

- po mikrorozkazie MC5 w przypadku instrukcji

03x xxx

(^{20A}xxx) dla A=2,3,4,5,6,7
(^{36A}x10)

20x xxx

36x xx0

- po mikrorozkazie MD3 w przypadku instrukcji

34x x2x

35x x2x

36x x2x

37x x2E . dla E=3,4,5,6,7

- po mikrorozkazie MD4 w przypadku instrukcji

34x x3x

35x x3x

36x x3x

37x x3E dla E=3,4,5,6,7

Z opisów mikrorozkazów MC5,MD3,MD4 wynika, że w chwili przejścia do mikrorozkazu MD5 w rejestrze RQ znajduje się adres argumentu danej instrukcji.

Mikrorozkaz MD5 dokonuje relokacji tego adresu według trybu 2 (rozdz. 1.5), a otrzymany fizyczny adres argumentu wpisywany jest do rejestru adresowego AR. Układ kontroli adresu bada czy relokowany adres mieści się w obszarze ograniczonym wielkościami:

$$DT(N) = 256 \cdot R_{15:8}(N)$$

$$LT(N) = 256 \cdot R_{7:0}(N)$$

przy czym numer rejestru

$$N=12 \text{ gdy } B(SR_S, SR_D, SR_M) = 0, 3, 4$$

$$N=13 \text{ gdy } B(SR_S, SR_D, SR_M) = 1, 2, 5, 7$$

Adres nie jest relokowany gdy $B(SR_S, SR_D, SR_M) = 6$. Stwierdzenie przekroczenia adresu powoduje powstanie sygnału $ADE1 = WADE = 1$ i ustawienie $ADE := WADE$.

Ze względu na występujące w mikrorozkazie MD6 odwołanie do pamięci, ustawione zostaje w czasie mikrorozkazu MD5 żądanie dostępu do szyny głównej ($CBRN0 = 0$).

Po MD5 następuje mikrorozkaz MD6, pod warunkiem, że $ADE1 = 0$ (nie zostało stwierdzone przekroczenie adresu). Natomiast w obecności $ADE1 = 1$ następuje przejście do mikrorozkazu MB1 rozpoczynającego rejestrację przerwania ADE.

MD6

Przejście do mikrorozkazu MD6 może nastąpić tylko z mikrorozkazu MD5.

Mikrorozkaz MD6 pobiera z pamięci operacyjnej argument instrukcji. Jest to mikrorozkaz asynchroniczny - czas jego trwania uza-

leżniony jest od czasu oczekiwania na wykonanie żądanej operacji pamięci.

Po wejściu do mikrorozkazu MD6 układ współpracy z szyną główną podtrzymuje ustawiony w czasie MD5 sygnał żądania dostępu do szyny. Żądanie to wycofywane jest po otrzymaniu z koordynatora kanałów sygnału GRN0=0. Sygnał GRN0 powoduje wprowadzenie na linie adresowe (ARN) słowa z rejestru adresowego (ustawionego przez mikrorozkaz MD5), a na linie MOPN - zanegowanego słowa MOP określającego właściwą dla danej instrukcji operację pamięci, a mianowicie:

- 07 (odczytaj słowo) dla instrukcji

03x xxx

20x xxx

36x xx0

3Nx xDx dla N=4,5,7 oraz D=2,3

- 11 (odczytaj słowo) dla instrukcji

202 xxx

362 xx0

- 10 (zapisz zero) dla instrukcji

203 xxx

363 xx0

- 14 (zapisz +1) dla instrukcji

204 xxx

364 xx0

- 15 (zapisz -1) dla instrukcji

205 xxx

365 xx0

- 16 (dodaj +1) dla instrukcji

206 xxx

366 xx0

- 17 (dodaj -1) dla instrukcji

207 xxx

367 xx0

Po ustawieniu adresu i kodu operacji układ sterowania oczekuje na strob OUCN, przebieg zegarowy procesora SXP blokowany jest aż do otrzymania strobu. Po nadejściu sygnału OUCN wykonywane są dla poszczególnych instrukcji następujące czynności (poza operacją w pamięci):

- w przypadku rozkazów instrukcji:

03x xxx

20x xxx

36x xx0

37x xxx

słowo pobrane z pamięci, znajdujące się na liniach informacyjnych IFN, umieszczane jest w rejestrze RQ oraz w rejestrze BR;

- w przypadku instrukcji:

34x xxx

35x xxx

słowo pobrane z pamięci zapisywane jest tylko do rejestru RQ;

- w przypadku instrukcji:

202 xxx

362 xx0

w przerzutniku E rejestru stanu zapamiętywany jest stan linii \overline{MZN} ($E=\overline{MZN}=1$ gdy słowo z pamięci jest zerem), a w przerzutniku L - stan linii \overline{IFN}_{15} ($L=\overline{IFN}_{15}=1$ gdy liczba reprezentowana słowem z pamięci w kodzie U2 jest ujemna).

- w przypadku instrukcji:

206 xxx 366 xx0

207 xxx 367 xx0

w przerzutniku X zapamiętywany jest stan linii MZN.

Zapis do BR,RQ bądź SR kończy wykonanie mikrorozkazu MD6, a następnym mikrorozkazem będzie MC1,MC2,MD7,MJ2 lub MH1.

- W przypadku instrukcji

20x xxx

36x xx0

mikrorozkaz MD6 stanowi fazę wykonania. Następnym mikrorozkazem jest MC1 rozpoczynający pobranie kolejnej instrukcji.

- W przypadku instrukcji

03x xxx

37x xxx

argument pobrany przez mikrorozkaz MD6 do rejestrów BR i RQ jest adresem skoku. Jeżeli wykonana jest jedna z instrukcji:

37x xx5

37x xx6

37x xx7

po MD6 następuje przejście do mikrorozkazu MC2 rozpoczynającego pobranie instrukcji wskazanej adresem skoku. Natomiast dla instrukcji:

03x xxx

37x xx3

37x xx4

następnym mikrorozkazem jest MJ2.

- W przypadku instrukcji

$3\frac{4}{5}x x\frac{2}{3}x$

pobrany argument jest adresem początkowym pola pamięci z którego lub do którego dokonane ma być blokowe przesłanie wskazanej liczby słów (adresowanie pośrednie). Mikrorozkazem następującym po MD6 jest w tym przypadku MD7, dokonujący relokacji tego adresu przed rozpoczęciem przesłania.

- Dla instrukcji

$\overline{20}x xxx$

36x $\overline{xx0}$

następnym mikrorozkazem jest MH1. Mikrorozkaz ten wykorzystuje umieszczony przez mikrorozkaz MD6 w rejestrze BR argument instrukcji jako argument wskazanej treścią instrukcji operacji arytmetycznej lub logicznej.

MD7

Mikrorozkaz MD7 następuje po mikrorozkazie:

MC5 dla instrukcji	$\left. \begin{array}{l} 0^1x \text{ xxx} \\ 3^4x \text{ x0x} \end{array} \right\}$
MD3 dla instrukcji	3^4x x1x
MD6 dla instrukcji	3^4x x_3^2x
MD4 dla instrukcji	$37x \text{ xx2}$
ME1 dla instrukcji	(3^0x x_1^0x)
ME2 dla instrukcji	$33x \text{ x}_1^0x$
ME3 dla instrukcji	$31x \text{ x}_1^0x$
ME4 dla instrukcji	$\begin{array}{c} 0 \\ 3^1x \text{ x}_3^2x \\ 3 \end{array}$

Jak wynika z opisu tych mikrorozkazów, w chwili wejścia do MD7 rejestr RQ zawiera adres słowa pamięci, które ma być przesłane do rejestru R(CA), lub słowa pamięci do którego ma być zapisana aktualna zawartość R(CA) (blok słów przy przesłaniu blokowym).

Mikrorozkaz MD7 dokonuje relokacji adresu według trybu 2 (zob. mikrorozkaz MD5). Relokowany adres wpisywany jest do rejestru AR. Układ badania adresu sprawdza: relokowany adres, w razie stwierdzenia przekroczenia obszaru pamięci, ustawiany jest sygnał ADE1=1. Pod wpływem sygnału ADE1 następuje przejście do obsługi przerwania ADE. Rejestrację przerwania rozpoczyna mikrorozkaz MB2 jeśli wykonywana była jedna z instrukcji:

$3Nx \text{ xxx}$ dla $N=0,1,2,3$

a w przypadkach pozostałych - mikrorozkaz MB1.

Jeśli nie zostało wykryte przekroczenie adresu, po MD7 następuje mikrorozkaz MD8. Z uwagi na odwołanie od pamięci następują-

ce w mikrorozkazie MD8, układ współpracy z szyną główną ustawia w czasie mikrorozkazu MD7 sygnał CBRN0=0.

MD8

Sygnał żądania dostępu do szyny głównej podtrzymywany jest do chwili otrzymania sygnału GRN0=0 z koordynatora kanałów; po nadejściu tego sygnału, na linii adresowe ARN wprowadzana jest wartość rejestru adresowego, a na linii MOPN - negacją słowa MOP określającego wymaganą przez daną instrukcję operację pamięci, a mianowicie:

- 01 (zapisz prawy bajt) dla instrukcji

$$3_3^2x \ x_3^2x$$

gdy przerzutnik RQM1 (ustawiony mikrorozkazem ME4) jest w stanie 1, co oznacza, że bajtowy adres argumentu jest nieparzysty;

- 02 (zapisz lewy bajt) dla instrukcji

$$3_3^2x \ x_3^2x$$

gdy RQM1=0, tzn. bajtowy adres argumentu jest parzysty;

- 03 (zapisz słowo) dla instrukcji

$$02x \ xxx$$

$$3_3^2x \ x_1^0x$$

$$35x \ xxx$$

$$37x \ x32$$

- 05 (odczytaj prawy bajt) dla instrukcji

$$3_1^0x \ x_3^2x$$

gdy RQM1=1, tzn. bajtowy adres argumentu jest nieparzysty;

- 06 (odczytaj lewy bajt) dla instrukcji

$$3_1^0x \ x_3^2x$$

gdy RQM1=0, tzn. bajtowy adres argumentu jest parzysty;

- 07 (odczytaj słowo) dla instrukcji

$$01x \ xxx$$

$$3_1^0x \ x_1^0x$$

$$34x \ xxx$$

37x x12

37x x22

Po otrzymaniu strobu OUCN z pamięci następuje:

- wprowadzenie na linie informacyjne IFN szyny głównej zawartości rejestru R wskazanego przez aktualną zawartość licznika CA - w przypadku instrukcji:

02x xxx

32x xxx

33x xxx

35x xxx

37x x32

- wpis słowa z linii informacyjnych do rejestru R wskazanego przez aktualną zawartość licznika CA - w przypadku instrukcji:

01x xxx

30x xxx

31x xxx

34x xxx

37x x12

37x x22

Ponadto zwiększana jest o 1 zawartość rejestru adresowego AR, licznika CA, oraz zmniejszana o 1 zawartość licznika kroków \overline{CN} .

Operacje te są istotne tylko dla instrukcji:

34x xxx

35x xxx

dla których po mikrorozkazie MD8, jeśli $\overline{CN} \neq 0$ następuje mikrorozkaz MD9, w którym dokonywana jest kontrola nowego (zwiększonego o 1) adresu, po czym następuje powrót do mikrorozkazu MD8 w celu przesłania kolejnego słowa. Pętla ta jest obiegana aż do wyczerpania zawartości licznika kroków CN ($CNZ := (\overline{CN} = 0)$). Liczba kroków, określona przez bity $IR_{3:0}$ (słowa instrukcyjnego zapamiętywana jest w CN w czasie mikrorozkazu MC5.

Mikrorozkaz MD8 jest asynchroniczny - przebieg zegarowy SXP jest wstrzymywany aż do otrzymania strobu OUCN z pamięci.

Wejście do mikrorozkazu MD8 następuje:

- z mikrorozkazu MD7, w przypadku gdy nie nastąpiło przekroczenie adresu

- z mikroroz^{ku} MD2 w przypadku instrukcji

37x x12

Po wykonaniu mikrorozkazu MD2 rejestr adresowy zawiera relokowany adres słowa stanowiącego część adresową instrukcji zgodnie z treścią instrukcji, mikrorozkaz MD8 pobiera to słowo do rejestru R(CA).

- z mikrorozkazu MD9, wykonywanego dla instrukcji

34x xxx

35x xxx

jeśli przeprowadzona w tym mikrorozkazie kontrola adresu nie wykazała przekroczenia obszaru pamięci.

Mikrorozkaz MD8 kończy wykonanie instrukcji:

01x xxx

02x xxx

30x xxx

31x xxx

32x xxx

37x xx2

Następnym mikrorozkazem jest w tych przypadkach mikrorozkaz MC1.

Przejście do MC1 następuje również w przypadku instrukcji:

34x xxx

35x xxx

jeśli zawartość licznika \overline{CN} jest równa zeru. Gdy natomiast $\overline{CN} \neq 0$, następnym mikrorozkazem jest dla tych instrukcji MD9, jak to już wyjaśniono wyżej.

Dla instrukcji:

33x xxx

MD8 nie jest ostatnim mikrorozkazem fazy wykonania i następuje przejście do mikrorozkazu ME3, w celu wpisania do rejestru R(E)

nowej zawartości, wyznaczonej przez mikrorozkaz ME1.

MD9

Sytuacje, w których następuje przejście z MD8 do MD9 wymieniono w opisie mikrorozkazu MD8.

Mikrorozkaz MD9 przeznaczony jest na sprawdzenie nowego, zwiększonego w czasie MD8 adresu znajdującego się w rejestrze AR. Stwierdzenie przekroczenia adresu powoduje powstanie sygnału WADE=ADE2, wpisanie 1 do przerzutnika ADE rejestru zgłoszeń przerwań i przejście do mikrorozkazu MB1, przy czym sprawdzane jest tylko górne ograniczenie adresu

$$AR \gg 256 \cdot R_{7:0}(N)$$

(zob. mikrorozkaz MD7). Gdy nie wystąpi przekroczenie adresu następuje powrót do mikrorozkazu MD8.

Mikrorozkaz MD9 ustawia sygnał żądania dostępu do szyny głównej CBRN0=0.

ME1

Wejście do mikrorozkazu ME1 następuje z mikrorozkazu MC5, jeżeli pobrana została jedna z instrukcji:

$$\begin{array}{l} (10x \text{ xxx}) \cdot \overline{IR}_4 \\ 3Nx \text{ xxx} \end{array} \quad \text{dla } N=0,1,2,3$$

Jak wynika z opisu mikrorozkazu MC5, w chwili wejścia do ME1 rejestr RQ zawiera liczbę reprezentowaną w kodzie U2 przez 4 najmniej znaczące bity słowa instrukcyjnego $IR_{3:0}$.

Mikrorozkaz ME1 dodaje zawartość rejestru RQ do zawartości rejestru R(B), a wynik umieszczony jest w rejestrach RQ i BR.

Po mikrorozkazie ME1 następuje przejście do MJ3, ME2, ME3, ME4 lub MD7:

- Jeśli wykonywana jest instrukcja

$$(10x \text{ xxx}) \cdot \overline{IR}_4$$

po ME1 następuje mikrorozkaz MJ3, podstawiający wyznaczony przez mikrorozkaz ME1 argument do rejestru R(CA).

- Argument wyznaczony przez mikrorozkaz ME1 jest w przypadku instrukcji

$$3_2^0x \ x_1^0x$$

adresem miejsca pamięci do którego ma być przesłane słowo z rejestru R(CA), lub miejsce z którego ma być pobrane nowe słowo do tego rejestru. Następnym mikrorozkazem jest MD7 relokujący wyznaczony adres.

- Dla instrukcji

$$3_2^0x \ x_3^2x$$

wyznaczony argument jest adresem bajtowym. Jako następny wykonywany jest w tym przypadku mikrorozkaz ME4, przekształcający adres bajtowy w adres słowa pamięci zawierającego wskazany bajt.

- Instrukcje

$$31x \ xxx$$

wymagają podstawienia otrzymanego w mikrorozkazie ME1 argumentu do rejestru R(B). W tym celu po ME1 wykonywany jest mikrorozkaz ME3.

- W przypadku instrukcji

$$33x \ xxx$$

z ME1 następuje przejście do mikrorozkazu ME2.

ME2

Przejście do tego mikrorozkazu następuje po mikrorozkazie ME1. Przeznaczony jest on tylko dla instrukcji

$$33x \ xxx$$

Adresem słowa lub bajtu pamięci do którego ma być przesłana zawartość R(CA) jest w przypadku tych instrukcji dotychczasowa zawartość rejestru R(B). Mikrorozkaz ME2 umieszcza ten adres w rejes-

trze RQ. Natomiast w rejestrze BR pozostaje zapamiętany wynik sumowania przeprowadzonego w czasie mikrorozkazu ME1.

Po ME2 następuje przejście do mikrorozkazu MD7 (relokującego umieszczony w rejestrze RQ adres), jeśli $IR_4=0$, natomiast gdy $IR_4=1$ następnym mikrorozkazem jest ME4, wyznaczający adres słowa zawierającego wskazany bajt.

ME3

Zadaniem mikrorozkazu ME3 jest wpisanie do rejestru R(B) nowej zawartości, wyznaczonej mikrorozkazem ME1 i zapamiętanej w rejestrze BR. Wykonywany jest on bezpośrednio po ME1 w przypadku instrukcji

31x xxx

oraz po mikrorozkazie MD8, a więc po zapisaniu zawartości rejestru R(CA) do pamięci, jeśli wykonywana jest jedna z instrukcji:

33x xxx

W tym ostatnim przypadku mikrorozkaz ME3 kończy fazę wykonania instrukcji. Następnym mikrorozkazem jest MC1.

Dla instrukcji:

31x xxx

gdy $IR_4=0$ następnym mikrorozkazem jest MD7, a gdy $IR_4=1$ - mikrorozkaz ME4, wyznaczający adres słowa zawierającego wskazany bajt.

ME4

Mikrorozkaz ME4 wykorzystywany jest przy wykonywaniu instrukcji operujących adresem bajtowym. Mikrorozkaz wyznacza adres słowa zawierającego wskazany bajt. Dokonywane jest to przez przesunięcie o jedną pozycję w prawo zawartości rejestru RQ, do którego został wpisany wyznaczony przez mikrorozkaz ME1 adres bajtowy. Bit z pozycji RQ_0 wprowadzany jest do przerzutnika RQM1; stan tego przerzutnika określa więc który z dwóch bajtów słowa (lewy

gdy RQM1=0, prawy gdy RQM1=1) ma być przesłany.

Wejście do mikrorozkazu ME4 następuje po mikrorozkazie:

- ME1 przy instrukcjach: $3\frac{0}{2}x x\frac{2}{3}x$
- ME2 przy instrukcjach: $33x x\frac{2}{3}x$
- ME3 przy instrukcjach: $31x x\frac{2}{3}x$

Z ME4 następuje przejście do mikrorozkazu MD7, relokującego wyznaczony adres.

Mikrorozkazy MJ1, MJ2, MJ3, MJ4

Mikrorozkazy tej grupy wykorzystywane są podczas wykonywania niektórych instrukcji skoku. Mikrorozkaz MJ1 wyznacza programowy adres skoku gdy wykonywana jest instrukcja podająca adres skoku względem zawartości licznika instrukcji. Dotyczy to instrukcji:

- 04x xxx
- 05x xxx
- 06x xxx
- 07x xxx

Zadaniem mikrorozkazów MJ2 i MJ3 jest umieszczenie w rejestrze R(CA) adresu powrotu przed wykonaniem skoku ze śladem:

- 03x xxx
- 37x xx3

bądź modyfikacji zawartości rejestru R(CA) oraz zbadanie warunku skoku dla instrukcji

- 04x xxx
- 37x xx4

Mikrorozkaz MJ3 bierze również udział w wykonaniu następujących instrukcji

- 1Nx xxx dla N=3,4,5,6
- (2Nx xxx)
(36x xxE) . dla N=E=1,2,4,5,6
- 37x x00
- 37x x10

dokonując wpisu wyniku operacji arytmetycznej lub logicznej do rejestru R(CA), jak również w wykonaniu instrukcji:

00x 6xx

00x 7xx

10x x_1^0 x

11x xxx

umieszczając w rejestrze R(CA) nową zawartość, wyznaczoną w poprzednich mikrorozkazach.

Rola mikrorozkazu MJ4 polega na umieszczeniu adresu skoku w rejestrze BR.

MJ1

Przejsście do tego mikrorozkazu następuje po mikrorozkazie MC5 w przypadku instrukcji:

04x xxx

05x xxx

06x xxx

077 xxx

W chwili przejścia do MJ1 rejestr RQ zawiera adres względny skoku (względem zawartości licznika instrukcji).

W ramach mikrorozkazu MJ1 do zawartości rejestru RQ dodawana jest zwiększona o 1 zawartość licznika instrukcji wynik umieszczany jest w rejestrze RQ oraz w rejestrze BR.

Dla instrukcji:

04x xxx

następnym po MJ1 jest mikrorozkaz MJ2, a dla pozostałych następnym mikrorozkazem jest MC2 i rozpoczyna się pobranie instrukcji wskazanej przez wyznaczony w mikrorozkazie MJ1 adres programowy.

MJ2

Treść mikrorozkazu zależy od wykonywanej instrukcji.

-- Dla instrukcji

03x xxx

37x xx3

wykonanie mikrorozkazu polega na wpisaniu do rejestru BR zwiększo-

nej o 1 zawartości licznika instrukcji, czyli adresu powrotu,

- W przypadku instrukcji:

04x xxx

37x xx4

do rejestru BR wpisywana jest zwiększona o 1 zawartość rejestru R(CA). Jeśli otrzymana liczba jest zerem, zostaje wpisane 1 do przerzutnika NQ.

Mikrorozkaz MJ2 jest wykonywany po mikrorozkazie:

- MC5 w przypadku instrukcji: 37x x0₄³

- MJ1 w przypadku instrukcji: 04x xxx

- MD3 w przypadku instrukcji: 37x x1₄³

- MD6 w przypadku instrukcji: 03x xxx

37x x2₄³

37x x3₄³

Z mikrorozkazu MJ2 następuje przejście do MJ3 dla wszystkich instrukcji.

MJ3

Mikrorozkaz ten przesyła zawartość rejestru BR do rejestru R(CA). Jest on wykonywany:

- po mikrorozkazie MJ2 przy instrukcjach:

03x xxx

04x xxx

37x xx3

37x xx4

w celu zapamiętania w rejestrze R(CA) adresu powrotu bądź wpisania do rejestru R(CA) nowej, zwiększonej o 1 zawartości,

- po mikrorozkazie MH1 przy instrukcjach:

1Nx xxx dla N=3,4,5,6

(2Nx xxx) dla N,E=1,3,4,5,6
36x xxE

37x x₁⁰

w celu przepisania z rejestru BR do rejestru R(CA) wyniku wykona-

nej operacji arytmetycznej lub logicznej;

- po mikrorozkazie MC5 w przypadku instrukcji:

11x xxx

w celu umieszczenia argumentu (znajdującego się w BR w chwili opuszczenia MC5) w rejestrze R(CA);

- po mikrorozkazie ME1 w przypadku instrukcji:

10x x_1^0 x

w celu wpisania do rejestru R(CA) sumy wyznaczonej przez mikrorozkaz ME1;

- po mikrorozkazie MS2 lub MS3, gdy wykonywana jest instrukcja przesunięcia. Jeśli jest to instrukcja przesuwania słowa pojedynczej długości, tzn. jedna z instrukcji:

00x $\frac{6}{7}D$ x dla D=0,1,2

przejście do MJ3 następuje po ostatnim wykonaniu mikrorozkazu MS2 tzn. po wyczerpaniu zawartości licznika kroków. Natomiast w przypadku instrukcji łącznego przesuwania zawartości dwóch rejestrów:

00x $\frac{6}{7}3$ x

przejście do mikrorozkazu MJ3 następuje po wykonaniu mikrorozkazu MS3.

Następnym po MJ3 mikrorozkazem jest:

- MJ4 dla instrukcji:

03x xxx

37x xx3

oraz dla instrukcji:

04x xxx

37x xxx gdy NQ=0 (nie jest spełniony warunek skoku);

- MC1 w pozostałych przypadkach.

MJ4

Mikrorozkaz ten następuje po MJ3 w wymienionych już wyżej przypadkach, a ponadto po mikrorozkazie MD3 gdy wykonywana jest jedna z instrukcji: 37x x1E dla E = 5,6,7.

Mikrorozkaz MJ4 umieszcza w rejestrze BR adres skoku, wpisany w czasie poprzednich mikrorozkazów do rejestru RQ.

W chwili wejścia do mikrorozkazu MC2, następującego po MJ4, adres skoku znajduje się więc zarówno w rejestrze RQ jak i w rejestrze BR.

2.4.6. Mikrorozkazy podstawowych operacji arytmetyczno-logicznych:

MH1, MH2, MS1, MS2, MS3, MG1, MG2, MG3

MH1

W mikrorozkazie MH1 wykonywane są jednoargumentowe operacje przy czym argument umieszczony jest w rejestrze BR oraz dwuargumentowe operacje nad argumentami umieszczonymi w R(CA) oraz BR. Wynik umieszczany jest w BR bądź powoduje ustawienie właściwych pozycji rejestru stanu.

Zawartość rejestru BR w chwili wejścia do MH1 zależy od wykonywanej instrukcji.

- Gdy jest to jedna z instrukcji:

1Nx xxx dla N=2,3,4,5,6,7

36x x00

37x x00

37x x10

przejście do MH1 następuje po mikrorozkazie MC5. Rejestr BR zawiera w chwili przejścia:

a) w przypadku instrukcji

12x xxx

13x xxx

liczbę reprezentowaną w kodzie U2 przez pozycje $IR_{7:0}$ słowa instrukcyjnego

$$BR = -IR_7 \cdot 2^7 + \sum_{i=0}^6 IR_i \cdot 2^i$$

b) w przypadku instrukcji

14x $\overline{\text{xxx}}$

15x xxx

16x xxx

17x xxx

słowo złożone z zer na pozycjach 15:8 oraz z pozycji $\text{IR}_{7:0}$

$$\text{BR} = (0_{15:8}, \text{IR}_{7:0})$$

c) w przypadku instrukcji

36x $\text{x0}\overline{0}$

37x x00

37x x10

słowo z rejestru R(B)

$$\text{BR} = \text{R(B)}$$

- Gdy wykonywana jest jedna z instrukcji

$2\overline{0}$ x xxx

36x $\text{x}\overline{0}\overline{0}$

wejście do MH1 następuje z mikrorozkazu MD6. Rejestr BR zawiera pobrany przez mikrorozkaz MD6 argument instrukcji.

Mikrorozkaz MH1 kończy wykonanie instrukcji (sygnał IRCO):

12x xxx

22x xxx

36x xx2

następnym mikrorozkazem jest w tych przypadkach MC1.

Dla instrukcji

1Nx xxx dla $N=3,4,5,6$

2Nx xxx }
36x xxE } dla $N,E=1,3,4,5,6$

37x xx0

kolejnym po MH1 mikrorozkazem jest MJ3, przepisujący wynik operacji arytmetycznej lub logicznej do rejestru R(CA).

Instrukcje

17x xxx

27x xxx

36x xx7

wymagają kolejnego wykonania dwóch operacji logicznych (sygnał IRTS). Pierwszą z nich wykonywana jest w ramach mikrorozkazu MH1, po czym następuje przejście do mikrorozkazu MH2 w celu wykonania drugiej operacji logicznej.

MH2

Mikrorozkaz MH2 jest przeznaczony tylko dla instrukcji:

17x xxx

27x xxx

36x xx7

Mikrorozkaz MH2 poprzedzony jest zawsze mikrorozkazem MH1, a następuje po nim mikrorozkaz MC1.

MS1

Mikrorozkazy MS1, MS2, MS3 tworzą fazę wykonawczą instrukcji przesuwania słów w rejestrach:

w lewo 00A 6XE

w prawo 00A 7XE

Mikrorozkaz MS1 wykonywany jest gdy mają być przesunięte łącznie dwa słowa, zawarte w rejestrach R(CA) i R(CA+1), tzn. w przypadku instrukcji:

00x 63x

00x 73x

Wejście do MS1 następuje po mikrorozkazie MC5. Wykonywanie mikrorozkazu MS1 polega na wpisaniu zawartości rejestru R(CA+1) do rejestru RQ. Po MS1 wykonywany jest mikrorozkaz MS2.

MS2

Wejście do mikrorozkazu MS2 następuje z mikrorozkazu MC5 dla instrukcji przesuwania pojedynczego słowa:

00x 63E

00x 73E

bądź z mikrorozkazu MS1 dla instrukcji przesuwania pary słów:

00x 63E

00x 73E

W pierwszym przypadku przesuwane słowo znajduje się w rejestrze BR, a w drugim przypadku w rejestrach BR,RQ.

Liczba przesunięć określona jest ^{przez} pozycje $E=IR_{2:0}$ słowa instrukcyjnego i wynosi $E+1$. W mikrorozkazie MC5 do licznika kroków wpisywane jest uzupełnienie do 17₂ liczby E i w mikrorozkazie MS2 słowo w CN jest następnikowane. Mikrorozkaz MS2 wykonywany jest dotąd, dopóki nie wystąpi sygnał $CNZ=(CN=17)$, tzn. $E+1$ razy.

Po wystąpieniu CNZ z mikrorozkazu MS2 następuje przejście do mikrorozkazu:

~~MS3 dla instrukcji 00x $\begin{matrix} 6 \\ 7 \end{matrix}x$~~

w celu wpisania przesuniętego słowa do rejestru R(CA)

~~- MS3 dla instrukcji 00x $\begin{matrix} 6 \\ 7 \end{matrix}x$.~~

W mikrorozkazie MS2 ustawiane mogą być ponadto pozycje SR_C oraz SR_V rejestru stanu:

- dla instrukcji 00x 60x $SR_C:=BR_{15}$

- dla instrukcji 00x 71x $SR_C:=BR_0$

- dla instrukcji 00x 62x $SR_V:=SR_V \vee (BR_{15} \oplus BR_{14})$

MS3

Mikrorozkaz MS3 wykonywany jest dla instrukcji

00x $\begin{matrix} 6 \\ 7 \end{matrix}x$

Wpisuje on zawartość rejestru RQ do rejestru R(CA+1). Po MS3

wykonywany jest mikrorozkaz MJ3, przesyłający zawartość rejestru BR do rejestru R(CA).

Mikrorozkazy MG1, MG2, MG3

Mikrorozkazy te tworzą fazę wykonania instrukcji

37x xx1

przy czym dla instrukcji:

37x x31

wykonywane są po mikrorozkazie MC5 kolejno mikrorozkazy MG1, MG2, MG3, a dla instrukcji:

37x x31

po MC5 następują mikrorozkazy MG2, MG3 (MG1 jest pomijany).

Po mikrorozkazie MG3 następuje przejście do mikrorozkazu MC1.

Przed przejściem do mikrorozkazów tej grupy, w ramach mikrorozkazu MC5, słowo z rejestru R(B) przesyłane jest do rejestru BR.

Następnie mikrorozkaz MG1 umieszcza w rejestrze RQ słowo rejestru R(CA).

Dla instrukcji

37A B01

mikrorozkazy MG2 i MG3 powodują przepisanie słowa z rejestru RQ do rejestru R(B) oraz słowa z rejestru BR do rejestru R(CA). W rezultacie nastąpiła wymiana słów pomiędzy rejestrami R(CA) oraz R(B).

Dla instrukcji

37A B11

mikrorozkaz MG2 powoduje przepisanie mniej znaczącego bajtu z rejestru RQ do rejestru R(B) na pozycje 7:0, a mikrorozkaz MG3 powoduje przepisanie mniej znaczącego bajtu z rejestru BR do rejestru R(CA) na pozycje 7:0. W rezultacie nastąpiła wymiana mniej znaczących bajtów pomiędzy rejestrami $R_{7:0}(CA)$ oraz $R_{7:0}(B)$ przy czym

bardziej znaczące bajty nie uległy zmianie.

Dla instrukcji

37A B21

mikrorozkaz MG2 powoduje przepisanie mniej znaczącego bajtu z rejestru RQ do rejestru R(B) na pozycje 15:8, a mikrorozkaz MG3 powoduje przepisanie bardziej znaczącego bajtu z rejestru R(B) na pozycje 7:0. W rezultacie nastąpiła wymiana mniej znaczącego bajtu z rejestru $R_{7:0}(CA)$ oraz bardziej znaczącego bajtu z rejestru $R_{15:8}(B)$ przy czym pozostałe bajty nie uległy zmianie.

Dla instrukcji

37A B31

w mikrorozkazie MG2 zerowany jest mniej znaczący bajt rejestru R(B), a w mikrorozkazie MG3 zerowany jest bardziej znaczący bajt rejestru R(CA), a na mniej znaczący bajt wpisywany mniej znaczący bajt z rejestru BR. W rezultacie w przypadku gdy $CA \neq B$ mniej znaczący bajt z rejestru $R_{7:0}(B)$ umieszczony zostanie w rejestrze $R_{7:0}(CA)$, wyzerowane zostaną $R_{7:0}(B)$ oraz $R_{15:8}(CA)$, a $R_{15:8}(B)$ pozostanie niezmienny. W przypadku gdy $CA = B$ jedynym skutkiem instrukcji będzie wyzerowanie $R_{15:8}(CA)$, a $R_{7:0}(CA)$ nie ulegnie zmianie.

2.4.7. Mikrorozkazy mnożenia i dzielenia MP1,MP2,...,MP9

Instrukcja mnożenia

W przypadku instrukcji mnożenia:

MUL A B o kodzie 37A B20

słowo reprezentujące mnożnik umieszczone jest w R(A), a mnożna - w R(B). Słowo reprezentujące iloczyn (podwójnej długości) umieszczone będzie w rejestrach (R(B),R(A)) przy czym:

$$L(R(B),R(A)) = L(R(B)) \cdot L(R(A))$$

Mnożna, mnożnik i iloczyn interpretowane są w kodzie U2 jako liczby całkowite (funkcja L przyporządkowuje słowom binarnym liczby w kodzie U2). W przypadku gdy iloczyn nie może być przedstawiony słowem o pojedynczej długości (w $R(A)$) tzn. gdy w wyniku $R_i(B) \neq R_{15}(A)$ (dla $i=15, \dots, 0$) sygnalizowany jest nadmiar w operacji mnożenia.

Czynności przy wykonywaniu instrukcji mnożenia można opisać następującym mikroprogramem.

MC5: $BR:=0, RQ:=R(CA), RQM1:=0,$
 $TV:=0, \overline{CN}:=15_{10} \quad \rightarrow MP3 ;$

MP3: $(BR:=BR + R(B))/(\overline{RQ}_0 \cdot RQM1) ,$
 $(BR:=BR - R(B))/(\overline{RQ}_0 \cdot RQM1) ,$
 $(BR:=BR)/(\overline{RQ}_0 \oplus RQM1) ,$
 $TV:=VAD \quad \rightarrow MP2 ;$

MP2: $(BR, RQ) := SHR(BR_{15} \cdot \overline{TV}, (BR, RQ)), RQM1 := RQ_0,$
 $\overline{CN} := DC(\overline{CN}) \quad \rightarrow (MP3, MP8) (\overline{CN}=0) ;$

MP8: $R(CA) := RQ \quad \rightarrow MP9 ;$

MP9: $R(B) := BR, SR_V := SR_V \vee (BR \oplus RQ_{15} \neq 0) \quad \rightarrow MC1 ;$

Mnożenie przeprowadzone jest metodą Booth'a tzn. sterowanie odbywa się na podstawie dwóch kolejnych bitów mnożnika. Słowo reprezentujące mnożnik umieszczone jest wstępnie w rejestrze RQ (MC5) i w kolejnych krokach przesuwane jest w prawo (MP2) tak, że skrajny prawy bit (z pozycji RQ_0) umieszczany jest w rejestrze RQM1.

Mnożna przechowywana jest w rejestrze R(B) i nie ulega zmianie.

Kolejne iloczyny częściowe przechowywane są w rejestrach BR i RQ tak, że w kolejnych krokach iloczyn częściowy przesuwany jest o jedną pozycję w prawo i w końcu zajmuje obydwa rejestry BR oraz RQ (iloczyn końcowy).

W mikrorozkazie MC5 zerowana jest wstępnie bardziej znacząca część iloczynu częściowego (BR), mnożnik umieszczany jest w rejestrze RQ przy czym dodatkowa pozycja RQM1 jest zerowana. Ponadto zerowany jest przerzutnik TV zapamiętujący nadmiar w krokach pośrednich mnożenia (korygowany w czasie przesuwania w prawo iloczynu częściowego), a licznik kroków ustawiany jest w stanie początkowym (CN=0).

Para mikrorozkazów MP3, MP2 wykonywana jest 16 razy i relizuje w każdym kroku operację:

$$A_{i+1} := (A_i + k_i \cdot D) / 2 \quad \text{dla } i = \text{CN} = 0, 1, \dots, 15$$

gdzie:

$$D = L(R(B)) - \text{mnożna}$$

A_i - i-ty iloczyn częściowy reprezentowany w kodzie U2 słowem przechowywanym w BR oraz część rejestru RQ:

$$A_0 = L(BR_{15:0}) = 0$$

$$A_i = L(BR_{15:0}, RQ_{15:16-i}) = \\ = -BR_{15} \cdot 2^{15+i} + \sum_{j=1}^{14+i} BR_{j-1} \cdot 2^j + \sum_{j=0}^{i-1} RQ_{16-i+j} \cdot 2^j$$

(dla $i=1, \dots, 16$ przy czym A_{16} jest iloczynem końcowym)

$k_i \in \{-1, 0, +1\}$ jest tworzone na podstawie kolejnych cyfr mnożnika i określa rodzaj operacji:

$$k_i = -R_i(CA) + R_{i-1}(CA) = -RQ_0 + RQM1$$

Nadmiar w krokach pośrednich może powstać tylko gdy mnożna reprezentowana jest słowem:

$$R(B) = 10\dots0 \quad \text{tzn.} \quad D = -2^{15}$$

a mnożnik ma postać:

$R(CA) = \dots 10 \dots 0$ przy czym jedynka występuje na pozycji i-tej. W tym przypadku $k_i = -1$, w MP3 wykonywana będzie operacja:

$$BR := -R(B)$$

a zatem $VAD=1$, $BR_{15}=0$. Jednakże w mikrorozkazie MP2 nadmiar zosta-

nie skorygowany i iloczyn częściowy będzie prawidłowy:

$$(BR, RQ) = 10...0$$

Iloczyn końcowy reprezentowany jest słowem 32-bitowym umieszczonym w rejestrach BR, RQ:

$$A_{16} = L(BR_{15:0}, RQ_{15:0}) = -BR_{15} \cdot 2^{31} + \sum_{j=16}^{30} BR_{j-16} \cdot 2^j + \sum_{j=0}^{15} RQ_j \cdot 2^j$$

W mikrorozkazach MP8 i MP9 iloczyn umieszczony jest w rejestrach programowych R(CA) oraz R(B). Jeśli wynik mnożenia nie może być przedstawiony przy pomocy pojedynczego słowa tzn. gdy na wszystkich pozycjach w BR nie znajdują się same zera gdy $RQ_{15}=0$ bądź same jedynki gdy $RQ_{15}=1$ sygnalizowany jest nadmiar operacji mnożenia przez ustawienie pozycji V w rejestrze stanu SR. Mikrorozkaz MP9 kończy mikroprogram mnożenia.

W przypadku gdyby interpretować mnożną i mnożnik jako liczby ułamkowe z przecinkiem pomiędzy pozycją 15 i 14 iloczyn byłby liczbą ułamkową z przecinkiem pomiędzy pozycjami 14 i 13 w bardziej znaczącym słowie.

Instrukcja dzielenia

W przypadku instrukcji dzielenia:

DIV A B o kodzie 37A B30

słowo reprezentujące dzielną umieszczone jest w rejestrze R(A), a dzielnik w rejestrze R(B). Słowo reprezentujące iloraz umieszczone będzie w rejestrze R(A), a reszta w rejestrze R(B). Dzielną, dzielnik, iloraz i reszta interpretowane są jako liczby całkowite w kodzie U2.

Czynności przy wykonywaniu instrukcji dzielenia można opisać następującym mikroprogramem:

MC5: RQ:=R(CA), AS:=R₁₅(CA), $\overline{CN}:=15_{10} \rightarrow$ MP1 ;

MP1: $BR_{15:0} := AS, RQ := SHL(RQ, X), RQM1 := AS \oplus R_{15}(B)$
 $DS := R_{15}(B), ADZ = (R(B) = 0), SR_V := SR_V \vee ADZ \rightarrow (MP2, MC1)ADZ ;$
 MP2: $(BR, RQ) := SHL((BR, RQ), RQM1), \overline{CN} := DC(\overline{CN}) \rightarrow MP3 ;$
 MP3: $(AD = BR + R(B)) / (RQ_0 \oplus (\overline{CN} = 14_{10})),$
 $(AD = BR - R(B)) / (RQ_0 \oplus (\overline{CN} = 14_{10})),$
 $ADZ = (AD = 0), BR := AD, RQM1 := AD_{15} \oplus DS,$
 $NQ := ADZ \rightarrow (MP2, (MP4, MP7)ADZ) (\overline{CN} = 0)$
 MP4: $(AD = BR + R(B)) / \overline{RQM1},$
 $(AD = BR - R(B)) / RQM1,$
 $ADZ = (AD = 0), OKR = ADZ \vee (BR_{15} \oplus AS),$
 $BR := AD \rightarrow (MP5, (MP7, MP6)RQM1)OKR ;$
 MP5: $(BR := BR - R(B)) / \overline{RQM1},$
 $(BR := BR + R(B)) / RQM1,$
 $NQ := 1 \rightarrow MP7 ;$
 MP6: $RQ := RQ + 1 \rightarrow MP7 ;$
 MP7: $R(CA) := SHL(RQ, NQ), SR_V := SR_V \vee AS \cdot DS \cdot RQ_{14} \rightarrow MP9 ;$
 MP9: $R(B) := BR \rightarrow MC1 ;$

Dzielenie przeprowadzone jest metodą nierętytucyjną, z korekcją reszty (i ilorazu) taką, aby reszta miała znak zgodny ze znakiem dzielnej i była co do wartości bezwzględnej mniejsza od dzielnika.

Reszty częściowe przechowywane są w rejestrach BR i RQ tak, że reszta początkowa zajmuje 16 bitów rejestru BR i 15 bitów rejestru RQ, a reszta końcowa zajmuje tylko rejestr BR.

Iloczyny częściowe zajmują mniej znaczącą część rejestru RQ (zwalnianą w kolejnych krokach przez reszty częściowe), a iloczyn częściowy po korekcji zajmuje cały rejestr RQ.

W mikrorozkazie MC5 dzielna z rejestru R(CA) umieszczana jest w rejestrze RQ, a bit znakowy dzielnej w rejestrze AS (1-bitowym)

oraz ustawiany jest licznik kroków CN na wartość początkową (CN=0).

W mikrorozkazie MP1 tworzona jest reszta początkowa A_{15} , ustawiany jest w RQM1 znakowy bit ilorazu na podstawie znaku dzielnej (z rejestru AS) i znaku dzielnika $R_{15}(B)$, znak dzielnika zapamiętywany jest w rejestrze DS (1-bitowym) oraz sprawdzane jest czy dzielnik jest różny od zera. Jeśli dzielnik jest równy zeru (ADZ) dzielenie jest zakończone i następuje przejście do mikrorozkazu MC1, przy czym dzielna i dzielnik w R(CA) i R(B) nie uległy zmianie.

Początkowa reszta A_{15} tworzona jest z dzielnej tak, że:

$$A_{15} = L(BR_{15:0}, RQ_{15:1}) = L(R(CA)) \cdot 2^{-15} =$$

$$= -BR_{15} \cdot 2^{15} + \sum_{i=0}^{14} BR_i \cdot 2^i + \sum_{i=-15}^{-1} RQ_{16+i} \cdot 2^i$$

przy czym $BR_i = AS$ dla $i=0, \dots, 15$

Kolejne cyfry ilorazu q_{15}, \dots, q_1 ustawiane są na podstawie znaku reszty A_i oraz znaku dzielnika (z rejestru AS) i przechowywane są wstępnie w RQM1. Cyfry ilorazu q_i przyjmują wartości $\{-1, +1\}$ reprezentowane przez $q_i \in \{0, 1\}$ przy czym

$$q_{15} = AS \oplus DS ; \quad q_{15}^2 = r(q_{15}) = Sg(R(CA)) \cdot Sg(R(B))$$

gdzie $r(x) = 1 - 2 \cdot x$, $x \in \{0, 1\}$, $r(x) \in \{+1, -1\}$

Na podstawie A_{15} , q_{15} , $R(B)$ określana jest w mikrorozkazach MP2, MP3 kolejna reszta A_{14} , iloraz częściowy i cyfra ilorazu q_{14} w RQM1 przy czym operacje wykonywane w mikrorozkazach MP2, MP3 przy ich pierwszym wykonaniu można opisać następująco:

$$A_{14} = 2 \cdot A_{15} - q_{15}^2 \cdot D \quad \text{gdzie } D = L(R(B))$$

$$Q_{14} = q_{15}^2 ;$$

$$q_{14}^2 = r(Q_{14}) = Sg(A_{14}) \cdot Sg(D)$$

przy czym reszta częściowa A_{14} oraz iloraz częściowy Q_{14} reprezen-

towane są słowami:

$$A_{14} = L(BR_{15:0}, RQ_{15:2})$$

$$Q_{14} = L1(RQ_0)$$

gdzie $RQ_0 = q_{15}$

L1 określa liczbę w kodzie dwujedynkowym.

Para mikrorozkazów MP2, MP3 zostanie powtórzona jeszcze 14 razy dla kolejnych wartości $i = \overline{CN} = 14, 13, \dots, 1$, a wyznaczane reszty częściowe i ilorazy częściowe będą równe:

$$A_{i-1} = 2 \cdot A_i - q_i' \cdot D ;$$

$$Q_{i-1} = 2 \cdot Q_i + q_i' ;$$

$$q_{i-1}' = r(\overline{q}_{i-1}) = Sg(A_{i-1}) \cdot Sg(D) ;$$

i reprezentowane będą słowami binarnymi:

$$A_i = L(BR_{15:0}, RQ_{15:16-i}) \quad \text{dla } i=15, \dots, 1$$

$$A_0 = L(BR_{15:0}) = -BR_{15} \cdot 2^{15} + \sum_{j=0}^{14} BR_j \cdot 2^j$$

$$Q_i = L1(RQ_{14-i:0}) \quad \text{dla } i=14, \dots, 0$$

$$Q_0 = L1(RQ_{14:0}) = \sum_{j=0}^{14} q_j' \cdot 2^j = (1 - 2 \cdot q_{14}) \cdot 2^{14} + \sum_{j=0}^{13} (2 \cdot q_j - 1) \cdot 2^j =$$

$$= -q_{14} \cdot 2^{15} + \sum_{j=1}^{14} q_{j+1} \cdot 2^j + 1$$

Otrzymane po zakończeniu wykonywania mikrorozkazów MP2 i MP3 reszta końcowa A_0 w rejestrze BR oraz iloczyn końcowy Q_0 w rejestrze RQ (w kodzie dwujedynkowym) ulegają w następnych mikrorozkazach korekcji tak, aby:

- reszta miała znak zgodny z dzielną,
- reszta była mniejsza od dzielnika (co do wartości bezwzględnej),
- iloraz był w kodzie U2.

Dzielna $A = L(R(CA))$,

dzielnik $D = L(R(B))$,

reszta końcowa $A_0 = L(BR)$,

iloraz $Q_0 = L1(RQ)$

związane są zależnościami:

$$\frac{A}{D} = Q_0 + \frac{A_0}{D}$$

Jeśli $A_0 = 0$ to w mikrorozkazie MP3 wpisana zostanie jedynka do przerzutnika NQ. W tym przypadku po ostatnim wykonaniu MP3 nastąpi przejście do mikrorozkazu MP7 w którym dokonana zostanie konwersja ilorazu Q_0 z kodu dwujedynkowego do kodu U2 (przesunięcie słowa w lewo i wstawienie 1 na najmniej znaczącą pozycję).

Jeśli po ostatnim wykonaniu mikrorozkazu MP3 $A_0 \neq 0$ to następuje przejście do mikrorozkazu MP4 w którym sprawdzany jest warunek czy reszta jest mniejsza od dzielnika (co do wartości bezwzględnej) i czy znak reszty jest zgodny ze znakiem dzielnej poprzez wykonanie operacji:

$$A'_0 = A_0 - q'_0 \cdot D$$

gdzie $q'_0 = r(\overline{RQM1}) = 2 \cdot RQM1 - 1$.

Jeśli tak otrzymana reszta A'_0 nie jest "poprawna" tzn. nie jest równa zero bądź ma znak różny od znaku dzielnej powstaje sygnał OKR=0 oznaczający, że poprawna była reszta A_0 . Następuje przejście do mikrorozkazu MP5 w którym odtwarzana jest reszta A_0 :

$$A_0 = A'_0 + q'_0 \cdot D$$

i wykonywany jest mikrorozkaz MP7 w którym następuje omówiona poprzednio konwersja do kodu U2.

Jeśli wyznaczona w MP4 reszta A'_0 jest "poprawna" tzn. jest równa zero bądź ma znak zgodny ze znakiem dzielnej (sygnał OKR=1) iloraz korygowany będzie w dwóch mikrorozkazach MP6 i MP7 w sposób wynikający z zależności:

$$\frac{A}{D} = Q_0 + \frac{A_0}{D} = (Q_0 + q'_0) + \frac{(A_0 - q'_0 \cdot D)}{D} = Q'_0 + \frac{A'_0}{D}$$

gdzie:

$$q_0' = Q_0 + q_0'$$

$$A_0' = A_0 - q_0' \cdot D$$

$$q_0' = 2 \cdot RQM1 - 1$$

Jeśli teraz $RQM1=0$ czyli $q_0' = -1$ to z mikrorozkazu MP4 nastąpi przejście do mikrorozkazu MP7 i słowo reprezentujące iloraz będzie przesunięte w lewo z wpisaniem zera na pozycję najmniej znaczącą zera ($NQ=0$ z mikrorozkazu MP3). W przypadku natomiast gdy $RQM1=1$ czyli $q_0' = +1$ iloraz powinien być zwiększony o 2 co zrealizowano tak, że w mikrorozkazie MP6 następującym po MP4 dodawana jest jedynka, a w mikrorozkazie MP7 słowo reprezentujące iloraz przesuwane jest w lewo ze wstawieniem zera na pozycję najmniej znaczącą.

Dodatkowo w mikrorozkazie MP7 wykrywana jest jeszcze jedna sytuacja: gdy może powstać nadmiar przy dzieleniu liczb całkowitych (poza przypadkiem $D=0$ wykrywanym w mikrorozkazie MP1). Jeżeli dzielna $A = L(R(CA)) = -2^{15} = L(10\dots0)$ dzielnik $D = L(R(B)) = -1 = L(11\dots1)$ to iloraz powinien być równy 2^{15} . Taka liczba nie może być reprezentowana w kodzie U2 słowem 16-bitowym i sygnalizowany jest nadmiar. Jest to wykrywane w taki sposób, że w mikrorozkazie MP7 (przed przesunięciem) słowo reprezentujące iloraz ma w tym przypadku postać:

$$L(RQ) = L(010\dots0) = 2^{14}$$

a zatem $AS \cdot DS \cdot RQ_{14} = 1$.

LISTA INSTRUKCJI UMC 20									
MAAS	INE	HNA	B	D	E	FUNKCJA			
SVEA A D SYEB A D	0 0 A 0 0 A	0000 0100	0000 0100	0000 0100	0000 0100	$\alpha = S(\theta); S(\alpha: a+1) := (TM, SR_{10}, IC+1, R(1:17))$ $R(16) := 1 - IR_1; R(17) := IR_{19:0} = 0072000; IC := S(\alpha)$			
XWT	5	0 0 X	4	X	0	HAIT			
XR XIR XTR XITR	5	0 0 X	4	0	1	$T := IR_4; I := IR_5; \alpha = S(\theta)$ $(TM, SR_{10}, R(10:17)) := S(\alpha: a+1)$			
	1	0 0 X	4	X	2,7	INE			
XUT A SEL XUT A MUX XUQ A SEL XUQ A MUX	5	0 0 A	5	0	00x 01x 10x 11x	$CH := R(A); TEST := IR_2;$ $(X, Y) := CH_{1:0}; R(A) := CH;$			
SGZ A SCA A SXY A PUT A GET A		0 0 A	5	1	0	$C := R_0(A)$ $C := R_{15}(A)$ $X := R_{12}(A) \oplus R_{14}(A); Y := R_{11}(A)$ $SR_{15:0} := R_{15:0}(A) \vee SR_{10:0} := R_{10:0}(A)$ $R(A) := SR_{15:0}$			
	1	0 0 X	5	1	5,7	INE			
MZR A E MZR BE		0 0 A 0 7 7	5 bb	1 bb	5 cc	$Z_i := R_i(A) / IR_i, i = 3, 2, 1, 0;$ $Z_i := IR_{i+4} / IR_i, i = 3, 2, 1, 0;$			
SLL SLC A E+1 SLA SLD		0 0 A	6	0	E	$(C, R(A)) := SHL(R(A), \theta)$ $(C, R(A)) := SHL(C, R(A), C)$ $(C, R(A)) := SHL(C, R(A), \theta); V := V \oplus R_{15}(A)$ $(C, R(A), R(A+1)) := SHL(C, R(A), R(A+1), \theta)$			
SRL SRC A E+1 SRA SRD		0 0 A	7	0	E	$R(A) := SHR(\theta, R(A))$ $(R(A), C) := SHR(C, R(A), C)$ $R(A) := SHR(R_{15}(A), R(A))$ $(R(A), R(A+1)) := SHR(C, R(A), R(A+1))$			
# ARG L I T X U B U Q U B & U		0 3 X 0 4 7 X 3 7 X	L 1 1 B X X X	L 1 1 B X X X	L 1 1 3 2 3	ARG $S(L) \quad 0 \leq L \leq 377$ $IC+1+1 \quad -0200 \leq I \leq 0177$ $R(B)$ U $S(U)$ $S(U+R(B))$			
JPS# A ARG JCT# A ARG		0 3 A 3 7 A	X X	X X	X 3	$R(A) := IC+1; IC := ARG;$ $R(A) := R(A)+1; IC := ARG/(R(A)+\theta)$			
BYR BXR BCR BYR # ARG BYS BXS BCS BVS		0 5 3 7	1 2 3 X X X X	1 2 3 X X X X	1 2 3 X X X X	$IC := ARG /$ $\begin{matrix} Y \\ X \\ C \\ V \\ Y \\ X \\ C \\ Y \end{matrix}$			
BNR BFR BSR BTR # ARG BNS BFS BSS BTS		0 6 3 7	1 2 3 X X X X	1 2 3 X X X X	1 2 3 X X X X	$IC := ARG /$ $\begin{matrix} Z \\ 0 \\ Z \\ 1 \\ Z \\ 2 \\ Z \\ 3 \\ Z \\ 4 \end{matrix}$			
JUN JGT JNE # ARG JLT JGE JEQ JLE		0 7 3 7	1 2 3 X X X X	1 2 3 X X X X	1 2 3 X X X X	$IC := ARG /$ $\begin{matrix} L \\ E \\ L \\ L \\ E \\ L \\ V \\ E \end{matrix}$			
XRQ; XSQ XLR; XRL; XRR; XRU; XSU; XRI; XSI	3	3 7 7	0 1 2 3	rd rd rd rd	Y	$Q := IR_3;$ $(A, B) := (IR_2, IR_3);$ $U := IR_3;$ $I := IR_3;$			
RCF; SCF; RVE; SVF RDF; SDF		3 7 7	4 5 6	rd rd rd	7	$C := IR_3;$ $V := IR_3;$ $D := IR_3;$			
SHP; SHF; SPF NOP	H+1	3 7 7 3 7 7	7 7	3 3	7 7	$(M, P) := (IR_4, IR_6)$			

	INE	MNA	B	D	E				
LDR A B LDR A B&E		1 0 A 1 0 A	8 B	0 B	0 0	$R(A) := R(B) + E; -010 \leq E \leq 7$			
F## A B		1 0 A	B	0	0	$FPE; R(16) := \theta 2; R(17) := IR_1;$			
LDI CMI A I ADI		1 1 A 1 2 A 1 3 A	1 1 1	1 1 1	1 1 1	$R(A) := I$ $E := (R(A) > I); L := (R(A) < I);$ $R(A) := R(A) + I + P - C;$			
IUF SUF A L RUF TUF		1 4 A 1 5 A 1 6 A 1 7 A	L L L L	L L L L	L L L L	$R(A) := R(A) \oplus L$ $R(A) := R(A) \vee L$ $R(A) := R(A) \cdot L$ $X := ((R(A) \cdot L) = \theta); Y := ((R(A) \cdot L) \neq \theta);$			
# ARG L B R B X B U B BRU		2 X 3 6 X 3 6 X 3 6 X 3 6 X	L B B X B	L B B X B	L X 1 X 2 X 5 X	ARG $S(L)$ $R(B)$ $S(R(B))$ $S(U)$ $S(U+R(B))$			
		1 1 1 3 1 3	2 0 0 6 0 7 6 X	X X X	X X 0	INE			
SC SZ SP # ARG SN IN DC		2 3 0 4 3 6	X X X X	X X X X	X X 0	$E := (ARG = \theta); L := (ARG < \theta);$ $ARG := \theta;$ $ARG := +1;$ $ARG := -1;$ $ARG := ARG + 1; X := (ARG = \theta)$ $ARG := ARG - 1; X := (ARG = \theta)$			
TN CH AD SU # A ARG OR AN TS		2 1 A 3 6 A 2 2 A 3 6 A 2 3 A 3 6 A 2 4 A 3 6 A 2 5 A 3 6 A 2 6 A 3 6 A 2 7 A 3 6 A	X X X X X X X X X X X X X X	X X X X X X X X X X X X X X	X X 1 2 3 4 5 6 7	$R(A) := -ARG - (P \cdot \bar{C}); SET(V, C);$ $E := (R(A) = ARG); L := (R(A) < ARG);$ $R(A) := R(A) + ARG + (P \cdot \bar{C}); SET(V, C);$ $R(A) := R(A) - ARG - (P \cdot \bar{C}); SET(V, C);$ $R(A) := R(A) \vee ARG;$ $R(A) := R(A) \cdot ARG;$ $X := (\overline{R(A) \cdot ARG} = \theta); Y := (R(A) \cdot ARG \neq \theta);$			
EOR A B NEG A B MUL A B DIV A B		3 7 A	B	0	0	$R(A) := R(A) \oplus R(B);$ $R(A) := \overline{R(A)};$ $(R(B), R(A)) := R(A) \cdot R(B); V := V \oplus (R_{15}(A) \neq \theta)$ $da := R(A); dk := R(B); R(A) \cdot dk + R(B) = da; SET(V)$			
EXR A B EXB A B EXS A B MBC A B MBC A A				0 1 2 3	0 1 1 1	$R(A) := R(B)$ $R_{15:0}(A) := R_{15:0}(B)$ $R_{15:0}(A) := R_{15:0}(B)$ $R(A) := R_{15:0}(B); R_{15:0}(B) := \theta;$ $R_{15:0}(A) := \theta;$			
LDL A L STL A L		0 1 A 0 2 A	L L	L L	L L	$R(A) := S(L);$ $S(L) := R(A);$			
LDX A B&E LDS A B&E STX A B&E STS A B&E SBS		3 0 A 3 1 A 3 2 A 3 3 A	B B	0 0 0 0	0 0 0 0	$R(A) := \frac{S}{SB} (R(B) + E); -010 \leq E \leq 7$ $R(B) := R(B) + E; R(A) := \frac{S}{SB} (R(B));$ $\frac{S}{SB} (R(B) + E) := R(A);$ $\frac{S}{SB} (R(B)) := R(A); R(B) := R(B) + E;$			
X U B U Q U B & U		3 4 3 5	X X	B B	0 1 2 3	ARG $R(B)$ U $S(U)$ $S(U+R(B))$ $0 \leq E \leq 7$			
LD# A ARG LG# A ARG ST# A ARG SG# A ARG		3 4 A 3 4 A 3 5 A 3 5 A	X X X X	X X X X	0 0 0 0	$(R(A), \dots, R(A+E)) := (S(ARG), \dots, S(ARG+E))$ $(S(ARG), \dots, S(ARG+E)) := (R(A), \dots, R(A+E))$			
	1	3 7 X	X	0	2	INE			
LDN A U LDU A B&U STU A B&U		3 7 A 3 7 A 3 7 A	X B B	1 2 3	2 2 2	$R(A) := U;$ $R(A) := S(U+R(B));$ $S(U+R(B)) := R(A);$			

MIKROPROGRAM CENTRALNEGO PROCESORA (1)																			
MI	DI	SI	SI	SI	SI	SI	SI	SI	SI	MI	DI	SI	SI	SI	SI	SI	SI	SI	SI
MA1/INT, DP4: R(10) := 077 777 / (KP=0); AR := ARP / (KP=1,2); (BSE, SVC, ADE, FPE, RT) := 0, SET(INR); → (MC1, MA2, MA2, MA3, MA4, MA5, MC1, MC5) KP, MB2) INT;										MC1: C := RT (KPMG), BR := RQ := R(10) + C → (MC2); 3x 00 11 c 1 40 0 3 3									
MA2/GRC: (MRQ(07, AR), RK := IFN) / (KP=2); (MRQ(03, AR), IFN := R) / (KP=1); AR := IN(AR), ARP := IN(ARP) → (MA1);										MC2: (ARW = RQ, WADE = 0) / SR ₃ , (M, P) := 0 / TMP; (ARW = RQ + DT(2), WADE = RQ ₀₇ + ARW) / LT(2) / SR ₃ ; AR := ARP := ARW, ADE := WADE, MRQ(07, AR), SET(MR); → (MC3, MB2) INT;									
MA3: ARP := RK → (MA1);										MC3: R(10) := RQ → (MC4);									
MA4: R(RP) := RK → (MA1);										MC4/GRC: RQ := IFM, RK := IR := IFN, CA := (0, IFN ₀₇) → (MC5, MA1) (TMP-STEP);									
MA5: RK := R(RP) → (MA1);										MC5: (RQ := RQ, BSC := SVC := IR ₀) / (00X XXX) IR ₂ ; SR ₁ := IR ₄ / (00X 4X) SR ₅ ; SR ₂ := R ₀ (CA) / (00X 510) SR ₃ := R ₁₅ (CA) / (00X 514) (SR _x := R ₁₅ (CA) ⊕ R ₁₄ (CA), SR ₇ := R ₁₅ (CA)) / (00X 512) ((n=10, n=15) SR ₅ , SR _{n=0} := R _{n=0} (CA)) / (00X 513) R(CA) := SR ₁₅₌₀ / (00X 514) (Z _i := R _i (CA) / IR _i , i=3,2,1,0) / (00X 50X) IR ₄ BR := R(CA) / (00X 50X)									
MB1: C := BSC + SVC, R(10) := BR + C → (MB2);										RQ := RQ · (000 377) / (00X XXX) IR ₂ ; BR := RQ · (000 377) / (00X XXX) IR ₂ ; RQ := RQ √ (000 377) / (00X XXX) IR ₂ ; (Z _i := IR ₂₄ / IR _i , i=3,2,1,0) / (077 XXX) RQ := RQ · (000 047) / (00X XXX) IR ₂ ; RQ := RQ √ (000 047) / (00X XXX) IR ₂ ; RQ := RQ / (10X XXX) IR ₄ ; RQ := R(B) / (134 5X) XOX; BR := R(B) / (136 X) XOX; RQ := R(C) / (136 X) XOX; BR := R(B) / (137 X) XXO) IR ₄ ; (PV = MP = IR ₃ , BR := 0, RQ := R(CA) / (137 X) XXO) IR ₄ ; BR := RQ := R(B) / (137 X) XOX)									
MB2/GRC: CHR := 0, SET(CS) → (MB4, MB5) / CS, MB5) CS = 0;										RQ := RQ / (10X XXX) IR ₄ ; RQ := R(B) / (134 5X) XOX; BR := R(B) / (136 X) XOX; RQ := R(C) / (136 X) XOX; BR := R(B) / (137 X) XXO) IR ₄ ; (PV = MP = IR ₃ , BR := 0, RQ := R(CA) / (137 X) XXO) IR ₄ ; BR := RQ := R(B) / (137 X) XOX)									
MB3: RQ := RK → (MB5);										SR ₀ := IR ₃ / (137 7 0X7) SR ₅ ; SR ₁ := IR ₄ / (137 1X7) SR ₅ ; SR ₂ := IR ₃ / (137 2X7) SR ₅ ; SR ₃ := IR ₃ / (137 3X7) SR ₅ ; SR ₄ := IR ₃ / (137 4X7) SR ₅ ; SR ₅ := IR ₃ / (137 5X7) SR ₅ ; SR ₆ := IR ₃ / (137 6X7) SR ₅ ; SR ₇ := IR ₃ / (137 7X7) MP									
MB4/TREH: PGN, CHR _{n+1} := GR _{n+1} , RQ := IFN, SET(CS) → (MB5);										TV := 0, NQ := 0, RQH := 0, AS := R ₁₅ (CA), (CN := (0, IR ₂₌₀), CR := 47) (10X XXX + 37X XXX), RT := WRT, TMP := SETMP, INE := NIME									
MB5/GRC: MRQ(07, 0), AR := IFN, CA := 06 → (MB6);										→ (MA1) (00x4x0) SR ₅ , MB5 (00x4x1) SR ₅ , MCH (00x50x) SR ₅ , MB1/BSC, MB2/(MNEV FPE), (00X 510) IR ₂ , (3H5) X XOX, (03 X XXX), (00X 5X X) IR ₂ , (36H) X XOX) IR ₄ , (00P) X XXX), (0B2X XX X) JUV, MD1 / (36 X XX X) IR ₄ , MD5 / (05 X XXX), (37 X X 03) JUV, (37 X X 02) IR ₄ , (36H) X 10), (37 7 (01X 7) SR ₅ , (37 7 X 07), (36 X 1 15), (37 7 41X 7), (37 7 X X X X), ME1 / (38 5X XXX) IR ₄ , (37 7 7 5 7) SR ₅ , (37 7 X X X X), ME1 / (38 5X XXX), (37 7 7 3 7), (412) X X X X), MF4 / (10 X XXX) IR ₄ + FP, (37 7 7 3 7), (36 X X 0 0), MS4 / (00 X 67 5 X), (0 4 X XXX), (37 X X X 0) IR ₄ , MS2 / (00 X 67 3 X), (015 X XXX) JUV, MG1 / (37 X X 3 1), MP1 / (37 X X 3 0), MJ3 / (11 X XXX), MG2 / (37 X X 3 1), MP3 / (37 X X 2 0);									
MB6/GRC: MRQ(03, 07) RT, AR, IFN := TH / (CA=06) RT; IFN := SR / (CA=07) RT; CA := IN(CA), IFN := R(CA) / (CA=10) RT; TH := IFN / (CA=06) RT; SR := TH / (CA=07) RT; R(CA) := TH / (CA=10) RT; → (MB7, MC1) RT, MB5) CA SET;										MC1: CPB := R(CA) / (CN=0), CPB := R(CA+1) / (CN=1), CPB := R(B) / (CN=2), CPB := R(B+1) / (CN=3), CN := IN(CN) → (MF1, MF2) CNZ;									
MB7: R(16) := CS ₂₌₀ → (MB8);										MF2/FPS: R(CA) := CPB / CN FPE1, R(CA+1) := CPB / CN FPE1, RQ := CPB / FPE1, CN := IN(CN), FPE := FPE1 → (MC1, MT2) CN, MB) FPE1									
MB8: R(17) := RQ → (MB9);										CNLD									
MB9/GRC: MRQ(07, 1), RQ := BR := IFN, CLEAR(INB), SR _{n=0} := 0, SR _{n=1} := 4, TMP := 0 → (MC2);										SETMP									
MCH/TREH: CMM, TESTM, (TRPN, IFN := R(CA)) / GR ₀ , (R(CA) := IFN, SR _{n=0} := IFN ₀) / GR ₀ → (MC1);										BRCL									
MG1: RQ := R(CA) → (MG2);										QSB0 QSB1 QSB2 QSB3 QSB4 QSB5 QSB6 SETMP									
MG2: R(B) := RQ / (37x04), R ₇₌₀ (B) := RQ ₀ / (37x11), R ₀₌₁ (B) := RQ ₀ / (37x24), R ₇₌₀ (B) := 0 / (37x34) → (MG3);										QSB0 QSB1 QSB2 QSB3 QSB4 QSB5 QSB6 SETMP									
MG3: R(CA) := BR / x04, R ₇₌₀ (CA) := BR ₀ / x11, R ₀₌₁ (CA) := BR ₀ / x24, R(CA) := (0 ₀₌₀ , B ₇₌₀) / x34 → (MC1);										QSB0 QSB1 QSB2 QSB3 QSB4 QSB5 QSB6 SETMP									
MF1: CPB := R(CA) / (CN=0), CPB := R(CA+1) / (CN=1), CPB := R(B) / (CN=2), CPB := R(B+1) / (CN=3), CN := IN(CN) → (MF1, MF2) CNZ;										QSB0 QSB1 QSB2 QSB3 QSB4 QSB5 QSB6 SETMP									
MF2/FPS: R(CA) := CPB / CN FPE1, R(CA+1) := CPB / CN FPE1, RQ := CPB / FPE1, CN := IN(CN), FPE := FPE1 → (MC1, MT2) CN, MB) FPE1										QSB0 QSB1 QSB2 QSB3 QSB4 QSB5 QSB6 SETMP									

Tabela 1. Mikrooperacje R(N), N, CA, DS, AS

SXP	RWEL	R(N) - LEWY BAJT	RWER	R(N) - PRAWY BAJT
X	0	$NR_{15:8} := R_{15:8}(N)$	0	$NR_{7:0} := R_{7:0}(N)$
0	1	$R_{15:8}(N) := DQ_{15:8}$	1	$R_{7:0}(N) := DQ_{7:0}$
		$DS := NR_{15} / SXP(\uparrow) \cdot MP1$ $AS := NR_{15} / SXP(\uparrow) \cdot MC5$		
RPN 2:0		N = RPS 3:0		
0		$(SR_A, CA_{2:0})$		
1		$(SR_A, IN(CA_{2:0}))$		
2		$(SR_B, IR_{7:5})$		
3		$(SR_B, IN(IR_{7:5}))$		
4		$(10, 11, 12, 13) / RFC_{1:0}$		
5		$(14, 15, 16, 17) / RFC_{1:0}$		
6		CA 3:0		
7		RFP 3:0		

Tabela 2. Mikrooperacije rejestru BR

SXP	ERCI	ERS	REJESTR BR	
X	0	X	BR _{15:0}	:= 0
X	1	0		-
↑	1	1	BR _{15:0}	:= (BRR, BR _{15:1})
↑	1	2	BR _{15:0}	:= (BR _{14:0} , BRL)
↑	1	3	BR _{15:0}	:= DQ _{15:0}
INSTR		BRR	INSTR	BRL
00X	70X	0	00X 60X	0
00X	71X	SR ₂	00X 61X	SR ₂
00X	72X	BR ₁₅	00X 62X	0
00X	73X	SR ₂	00X 63X	RQ ₁₅
37X	X20	BR ₁₅ TV	37X X30	RQ ₁₅

Tabela 3. Mikrooperacije rejestru RQ

SXP	RQS	REJESTR RQ	
X	0		-
↑	1	RQ _{15:0}	:= (RQR, RQ _{15:1})
↑	2	RQ _{15:0}	:= (RQ _{14:0} , RQM1)
↑	3	RQ _{15:0}	:= DQ _{15:0}
SXP	RQMS	REJESTR RQM1	RQR
X	00000	-	
↑	10000	RQM1 := ASQBQ ₁₅	
↑	01000	RQM1 := RQ ₀	0 BR ₀
↑	00100	RQM1 := DSQAD ₁₅	1 0
↑	00010	RQM1 := RQ ₀	
X	00001	RQM1 := 0	

Tabela 4. Mikrooperacje multiplexerów ABQ, CQ, DQ

MULTIPLESER ABQ (BQ = AQ)				
BQS	BQL	AQ _{15:8} :=	BQF	AQ _{7:0} :=
0	0	NR _{15:8}	0	NR _{7:0}
1	0	BR _{15:8}	0	BR _{7:0}
2	0	RQ _{15:8}	0	RQ _{7:0}
3	0	RQ _{7:0}	0	BR _{15:8}
4	0	IFN _{15:8}	0	IFN _{7:0}
5	0	SR _{15:8}	0	SR _{7:0}
6	0	CPB _{15:8}	0	CPB _{7:0}
7	0	TM _{15:8}	0	TM _{7:0}
X	1	0	1	0

$BQS=B(BQC, BQB, BQA)$

MULTIPLESER CQ					
CQS	CQL	CQ _{15:8} :=	CQRL	CQ _{7:4} :=	CQ _{3:0} :=
0	0	BQ _{15:8}	0	BQ _{7:4}	BQ _{3:0}
1	0	AQ _{15:8}	0	AQ _{7:4}	AQ _{2:0}
2	0	REL _{15:8}	0	RQ _{7:4}	RQ _{3:0}
3	0	177	0	17	17
X	1	0	1	0	-

$CQS=B(CQB, CQA)$

MULTIPELKSER DQ	
DQ _{15:0} :=	$(\overline{CQ}_{15:0}, \overline{ADN}_{15:0})/DQS$

Tabela 5. Jednostka arytmetyczno-logiczna

ADS	BQE	BQS	CQE	CQS	AD _{15:8} :=
06	0	1	0	0	BR - R(N) - CIN
11	0	2	0	0	RQ + R(N) + CIN
11	0	1	0	0	BR + R(N) + CIN
14	0	1	0	1	R(N) - BR - CIN
11	3	X	0	0	R(N) + CIN
14	0	2	0	0	2 · RQ + CIN
17	0	1	0	0	ER + CIN
17	0	1	0	1	-BR - CIN
20	0	1	0	0	ER
20	0	2	2	0	(O _{15:8} , RQ _{7:0})
20	0	2	3	0	(O _{15:4} , RQ _{3:0})
20	0	6	3	1	(O _{15:4} , CPB _{3:0})
31	0	1	0	0	BR ⊕ R(N)
32	X	X	X	X	R(N)
33	0	1	0	0	ER ∨ R(N)
35	0	1	0	0	BR · R(N)
36	0	1	0	0	BR · R(N)
36	0	1	0	1	BR · R(N)

SXP	MC5	MP3	MP5	MP2	NQ
X	0	0	0	0	-
X	1	0	0	0	NQ := 0
1	0	1	0	0	NQ := ADZ
1	0	1	0	0	NQ := 1
1	0	0	0	1	NQ := ADZ

SXP	MC5	MP3	TV
X	0	0	-
1	1	0	TV := 0
1	0	1	TV := VAD

ADS = B(ADM, ADS3, ADS2, ADS1, ADS0)

BQE = B(BQL, BQR) CQE = B(CQL, CQL)

NADMIAR VAD

CQA	ADS0, ADS3	ADS1, ADS2	A ₁₅ B ₁₅ AD ₁₅	00	06	11	17	11	17	VM	VAD :=	5A14E
0	0	0	0	1	1							
0	0	1	1	1	1							
0	1	0	1	0	1							
0	0	0	0	0	0	0	1			0	CQA · ADS2	2Y3
1	0	1	0	0	0	0	0			1	$\overline{CQA} \cdot \overline{ADS2}$	2Y0
1	0	1	0	0	0	0	1	1		2	CQA · ADS3	1Y3
1	0	0	1	1	0	0	0	0		3	$\overline{CQA} \cdot \overline{ADS3}$	1Y0
0	1	1	0	1	0	1	0	0		4	$\overline{CQA} \cdot \overline{ADS2}$	2Y1
0	1	0	0	0	0	0	1	0		5	CQA · ADS2	2Y2
0	0	1	0	0	1	1	0	0		6	$\overline{CQA} \cdot \overline{ADS3}$	1Y1
0	0	0	0	0	0	0	0	0		7	-	-

VM = B(BQ₁₅, NR₁₅, ADN₁₅) = B(A₁₅, B₁₅, AD₁₅)

Tabela 6. Mikrooperacje IR, CA, CN.

SXP MC4	REJESTR IR
X 0	-
↑ 1	IR _{15:0} := BQ _{15:0}

SXP CALD MC4 CAEN	LICZNIK CA
↑ 0 0 X	CA := 06
↑ 0 1 X	CA _{3:0} := (0, BQ _{10:8})
X 1 X 0	-
↑ 1 0 1	CA := IN(CA)
CA17 := (CA = 17)	

SXP CNLD MC5 CNEN	LICZNIK CN
↑ 0 1 X	CN _{3:0} := (0, IR _{2:0})
X 1 0 0	-
↑ 1 0 1	CN := DC(CN)
↑ 1 0 1	CN := 17
CNZ := (CN = 0)	

Tabela 7. Mikrooperacje rejestru AR.

SXP ARLD AREN	REJESTR AR
↑ 0 X	AR _{15:0} := CQ _{15:0}
X 1 0	-
↑ 1 1	AR := IN(AR)

Tabela 8. Mikrooperacje licznika TM.

TWCK SR14 TMLD	LICZNIK TM
X X 0	TM _{15:0} := DQ _{15:0}
↑ 0 1	TM _{15:0} := DC(TM _{15:0})
X 1 1	-
TMZ := (TM = 0)	

SXP SRCR Q81 Q8A2 LRTS	Y SR ₀ :=	SXP SRCR Q830 Q831 Q832	Z _{3:0} SR _{7:4} :=
X 0 X XX	0	X 0 XXX	0
↑ 1 1 00	DQ ₀	↑ 1 100	DQ _{7:4}
↑ 1 0 10	BQ ₁₅	↑ 1 010	BQ _{3:0} /IR _{3:0}
↑ 1 0 01	ADZ	↑ 1 001	IR _{7:4} /IR _{3:0}
SXP SRCR Q81 Q8A2 MH2 XZR	X SR ₁ :=	SXP SRCR Q82 LRTS	E SR ₈ := L SR ₉ :=
X 0 X XXX	0	X 0 XXX	0
↑ 1 1 000	DQ ₁	↑ 1 100	DQ ₈ DQ ₉
↑ 1 0 100	BQ ₁₅ ⊕ BQ ₁₄	↑ 1 010	ADZ DQ ₁₅
↑ 1 0 010	ADZ	↑ 1 001	MZN BQ ₁₅
↑ 1 0 001	MZN		
SXP SRCR Q82 Q8A0 Q8A1 Q8B4 MPT ICSR SHLC	C SR ₂ :=	SXP SRCR Q82 Q8B6	D SR ₁₀ :=
X 0 X XXX XXX	0	X 0 XX	1
↑ 1 1 000 000	DQ ₂	↑ 1 10	DQ ₁₀
↑ 1 0 100 000	BQ ₀	↑ 1 01	IR ₃
↑ 1 0 010 000	BQ ₁₅		
↑ 1 0 001 000	IR ₃		
↑ 1 0 000 100	CAD		
↑ 1 0 000 010	BR ₀		
↑ 1 0 000 001	BR ₁₅		
SXP SRCR Q82 Q8A5 MPT IASR MP7 VMP	V SR ₃ :=	SXP SRCR Q83 Q8B0	Q SR ₁₁ :=
X 0 X XXX XXX	0	X 0 XX	0
↑ 1 1 000 000	DQ ₃	↑ 1 10	DQ ₁₁
↑ 1 0 100 000	IR ₃	↑ 1 01	IR ₃
↑ 1 0 010 000	SR ₃ ∨ VAD		
↑ 1 0 001 000	SR ₃ ∨ (BR ₁₅ ⊕ BR ₁₄)		
↑ 1 0 000 100	SR ₃ ∨ ADZ		
↑ 1 0 000 010	SR ₃ ∨ AS · DS · AD ₁₅		
↑ 1 0 000 001	SR ₃ ∨ ADZ		
SXP SRCR Q83 Q8B1 Q8B2		SXP SRCR Q83 Q8B1	B SR ₁₂ := A SR ₁₃ :=
X 0 X XXX XXX	0	X 0 XX	1
↑ 1 1 000 000	DQ ₃	↑ 1 10	DQ ₁₂ DQ ₁₃
↑ 1 0 100 000	IR ₃	↑ 1 01	IR ₃ IR ₄
↑ 1 0 010 000	SR ₃ ∨ VAD		
↑ 1 0 001 000	SR ₃ ∨ (BR ₁₅ ⊕ BR ₁₄)		
↑ 1 0 000 100	SR ₃ ∨ ADZ		
↑ 1 0 000 010	SR ₃ ∨ AS · DS · AD ₁₅		
↑ 1 0 000 001	SR ₃ ∨ ADZ		
SXP SRCR Q83 Q8B2		SXP SRCR Q83 Q8B2	S SR ₁₄ :=
X 0 X XXX XXX	0	X 0 X	1
↑ 1 1 000 000	DQ ₃	↑ 1 1	DQ ₁₄
↑ 1 0 100 000	IR ₃		
↑ 1 0 010 000	SR ₃ ∨ VAD		
↑ 1 0 001 000	SR ₃ ∨ (BR ₁₅ ⊕ BR ₁₄)		
↑ 1 0 000 100	SR ₃ ∨ ADZ		
↑ 1 0 000 010	SR ₃ ∨ AS · DS · AD ₁₅		
↑ 1 0 000 001	SR ₃ ∨ ADZ		
SXP SRCR Q83 Q8B2		SXP SRCR Q83 Q8B2	U SR ₁₅ :=
X 0 X XXX XXX	0	X 0 XX	0
↑ 1 1 000 000	DQ ₃	↑ 1 10	DQ ₁₅
↑ 1 0 100 000	IR ₃	↑ 1 01	IR ₃
↑ 1 0 010 000	SR ₃ ∨ VAD		
↑ 1 0 001 000	SR ₃ ∨ (BR ₁₅ ⊕ BR ₁₄)		
↑ 1 0 000 100	SR ₃ ∨ ADZ		
↑ 1 0 000 010	SR ₃ ∨ AS · DS · AD ₁₅		
↑ 1 0 000 001	SR ₃ ∨ ADZ		

SXP SRCR Q83 Q8B3	I SR ₁₆ :=	SXP SRCR Q87 MOC	P SR ₁₇ :=	M SR ₁₈ :=	SXP SRCR WRT	F SR ₁₉ :=
X OXX	0	X OXX	0	0	X OX	0
X 110	1	↑ 110	IR ₃	IR ₄	↑ 11	IR ₄
↑ 101	IR ₃	X 101	SR ₁₇ · TMP	SR ₁₈ · TMP		

SXP SRCR Q85 MOC Q8B7	TMP :=	SXP MA1 MOC WRT	RT :=
X 0 XXX	0	X 10C	0
↑ 1 110	1	↑ 010	0
↑ 1 101	SR ₁₈ · SR ₁₇	↑ 011	1
↑ 1 100	0		

Tabela 10. Układ zgłoszeń przerwania zewnętrznych.
Mikrooperacje SET(INR), CLEAR(INR)

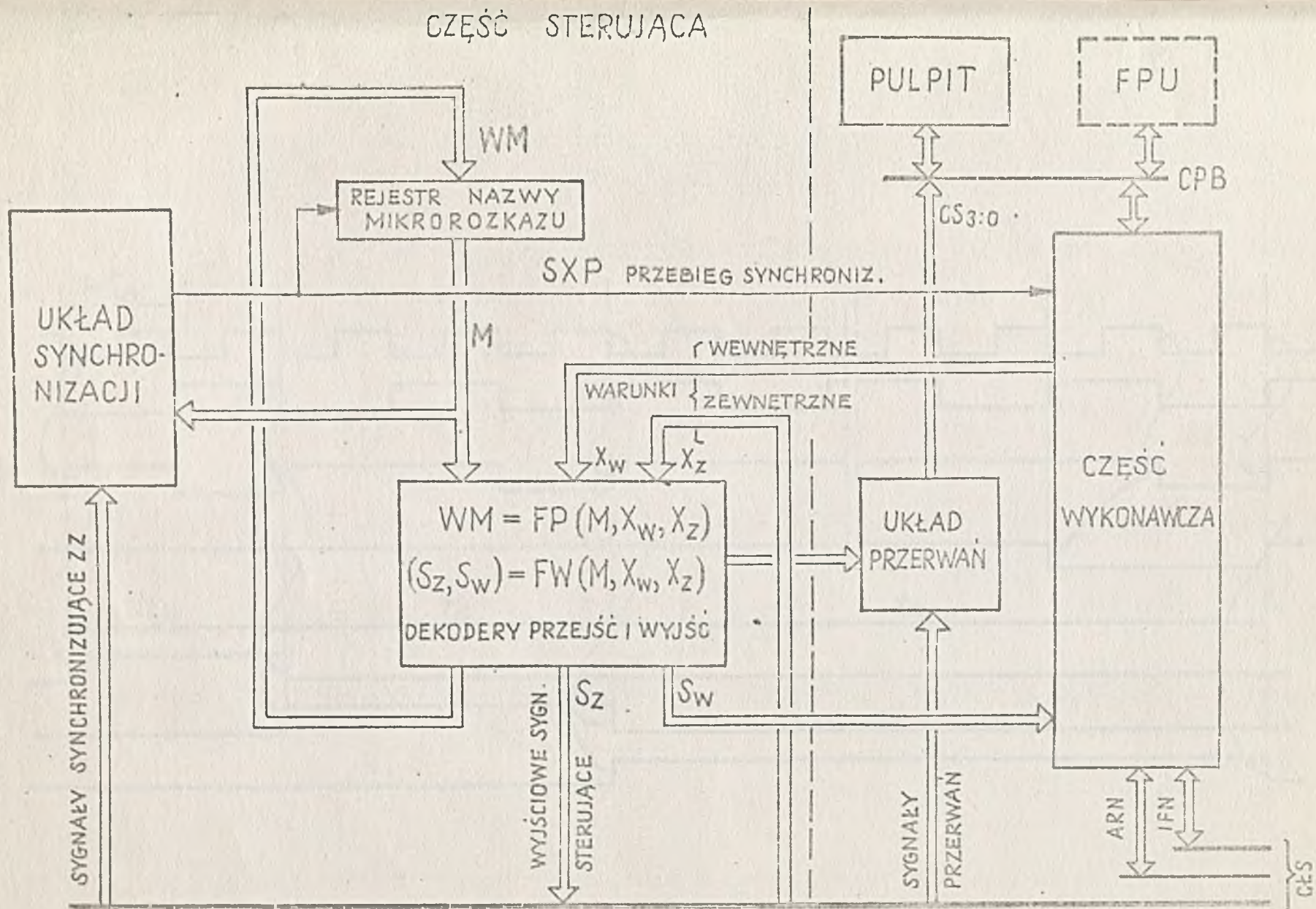
$$I1 = \overline{TMP} \cdot SR_{16} ; \quad I2 = I1 \cdot SR_{15}$$

CLRN	PFLZ	CS11	PFLW	RESN CS10	RESW	REQN CS06	REQW	TIMZ	CS05	TIMW
0	1 X	0	1 X	0	1 X	0	1 X	0	1 X	0
1	0 X	I1	0 X	I1	0 X	I1	0 X	I1	0 X	I2
1	1 †	0	1 †	0	1 †	0	1 †	0	1 †	0

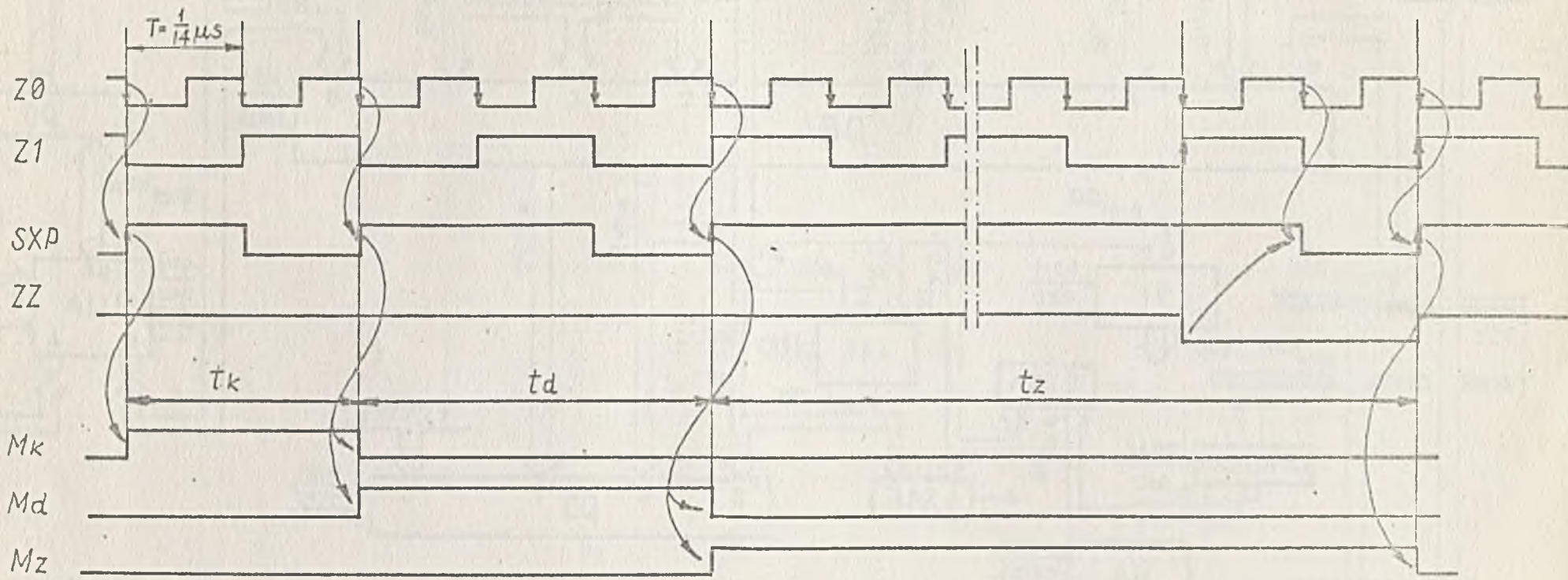
$\overline{CHIW} = I2 \cdot CHI ;$
 $(PFLZ, RESZ, ORQZ, TIMZ, CHIZ) :=$
 $(PFLW, RESW, REQW, TIMW, CHIW) / INCK(\dagger) ;$
 $INCK = MC2 \vee MA1 \cdot Z1 ;$
 $\overline{TRZ} = SR_T \cdot \overline{TMP} ;$

Tabela 11. Przerzutniki przerwania programowych

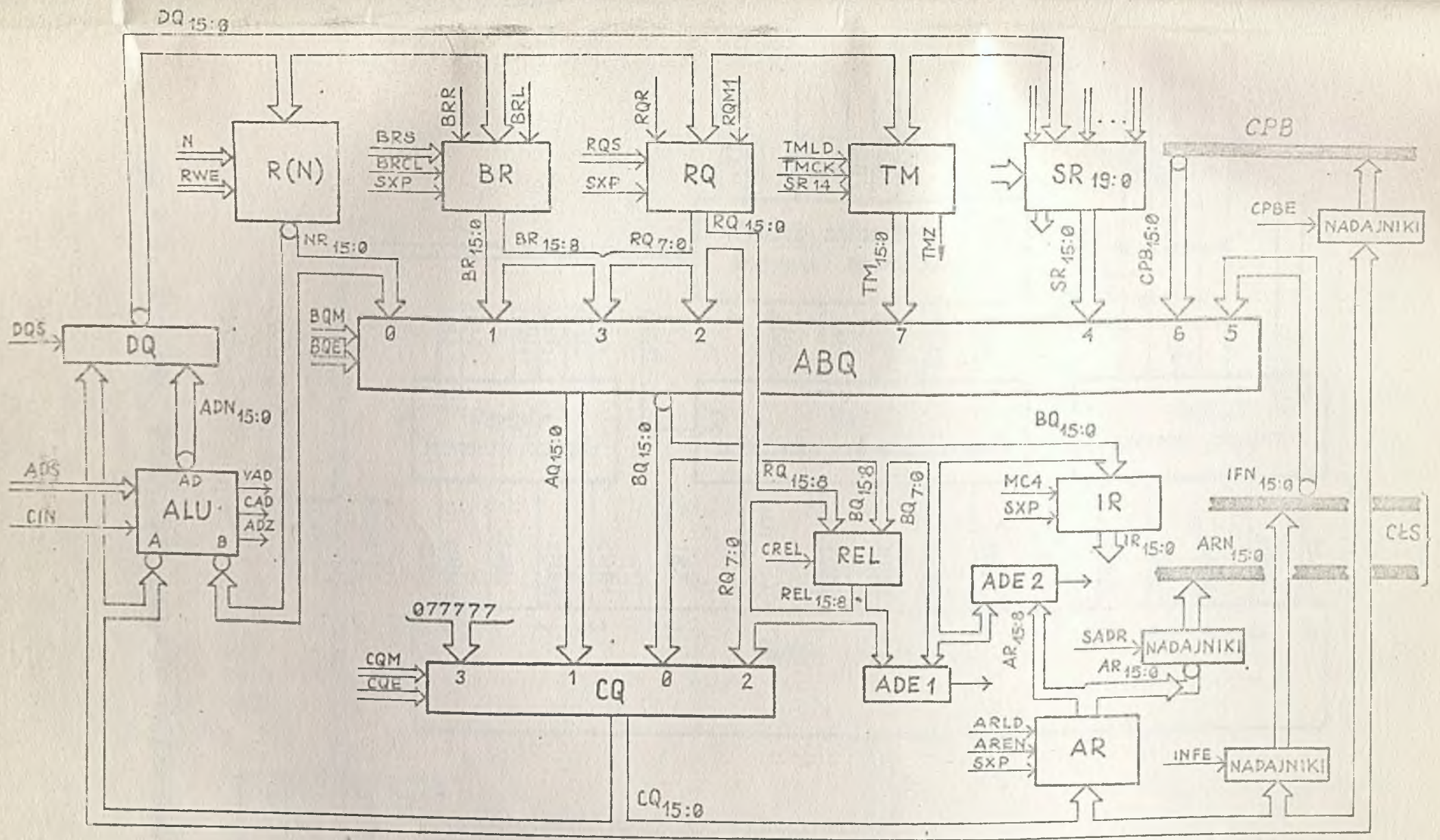
SXP	MA1 MC5	MD9	MRO5 MF2	BSC :=	SVC :=	INE :=	ADE :=	FPE :=
X	1 0	0 0	0 0	0	0	0	0	0
†	0 1	0 0	0 0	WBSC	WSVC	WINE	-	FPE
†	0 0	1 0	0 0	-	-	-	WADE	-
†	0 0	0 1	0 0	-	-	-	WADE	-
†	0 0	0 0	0 1	-	-	-	-	FPE1



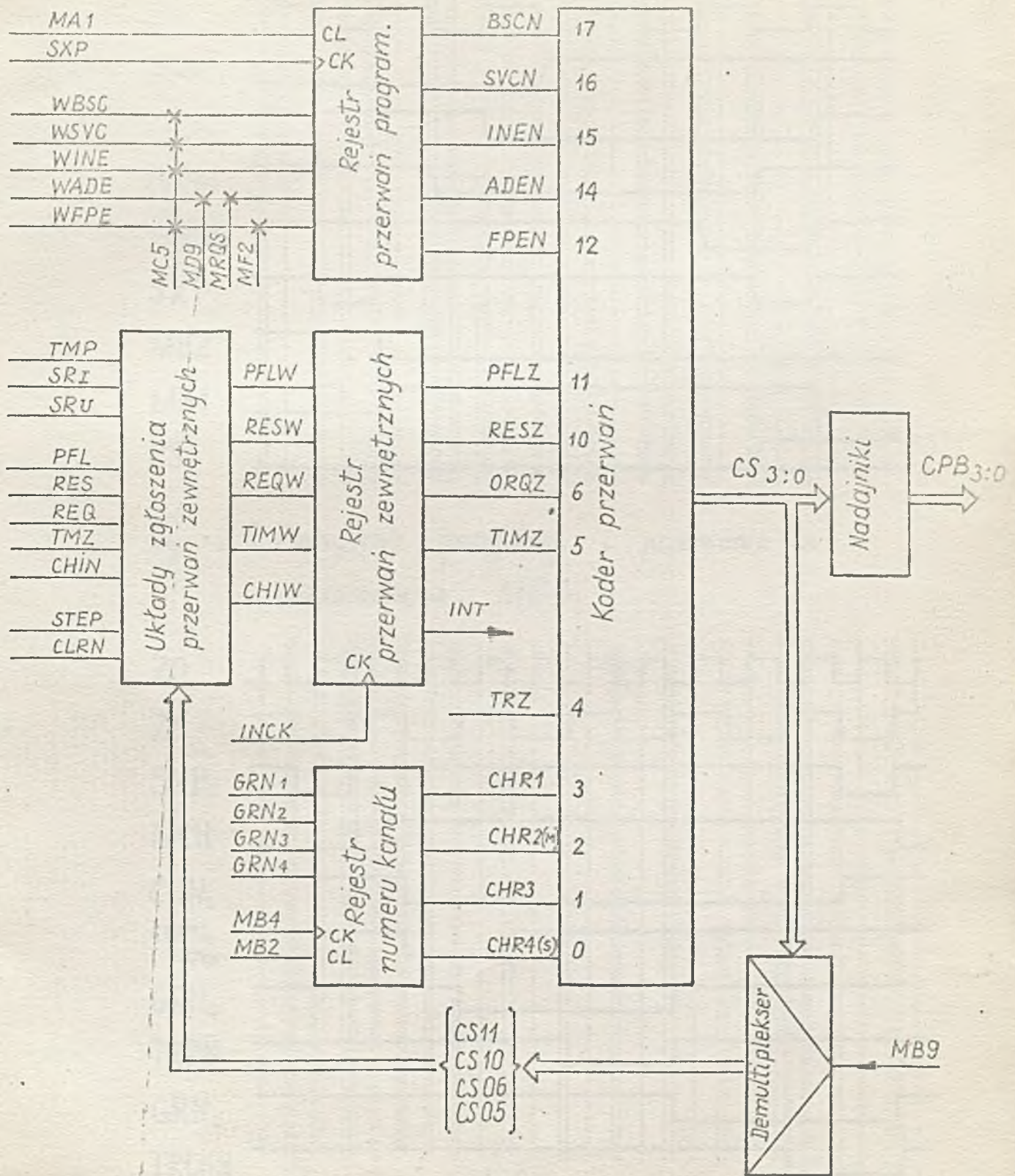
Rys. 2.1. Budowa centralnego procesora maszyny cyfrowej UMC-20



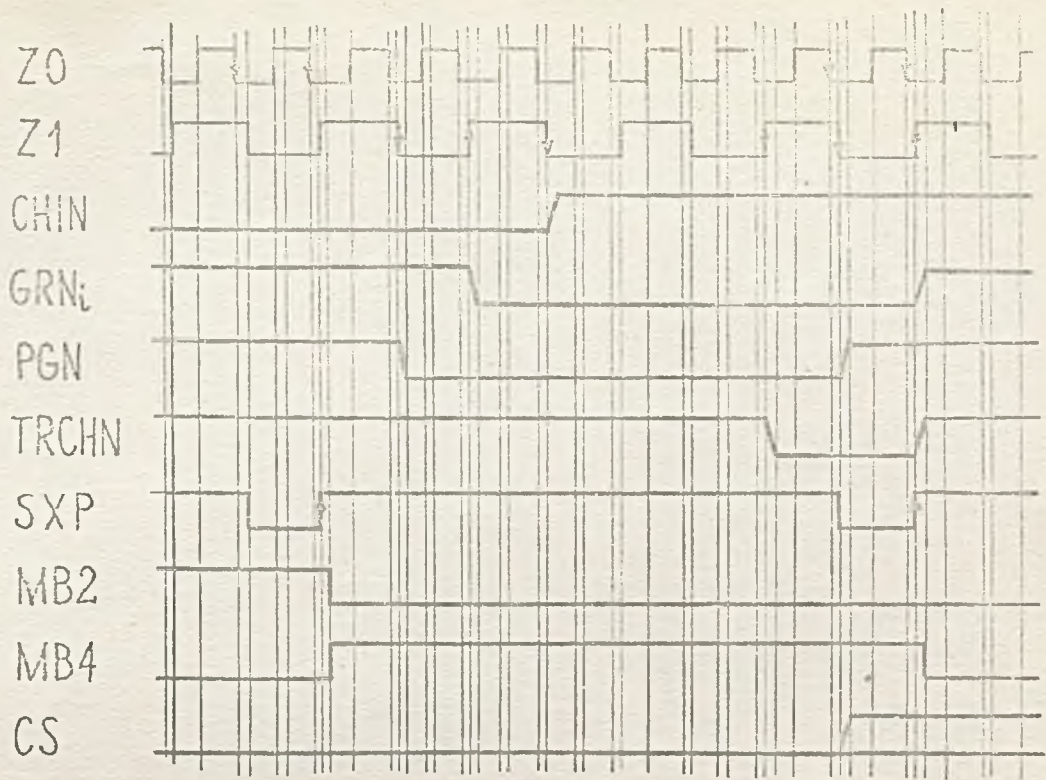
Rys.2.2 Synchronizacja mikrorozkazów centralnego procesora



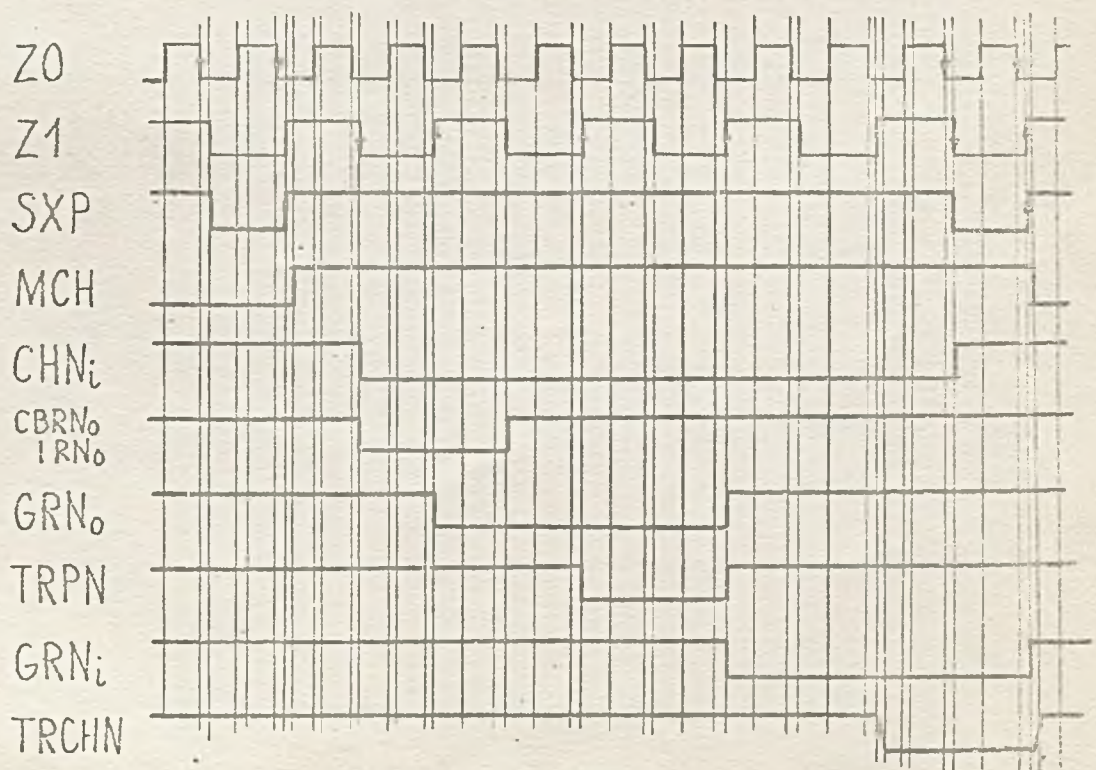
Rys.2.3. Część wykonawcza centralnego procesora [CPU] UMC-20



Rys. 2.4. SCHEMAT BLOKOWY UKŁADU PRZERWAŃ



Rys. 2.5. Przesyłanie specyfikacji i przerwania w mikrorozkazie MB4



Rys. 2.6. Zależności czasowe w mikrorozkazie MCH przesłania rozkazu do kanału

