

Technika Mikroprocesorowa I

Wykład 4

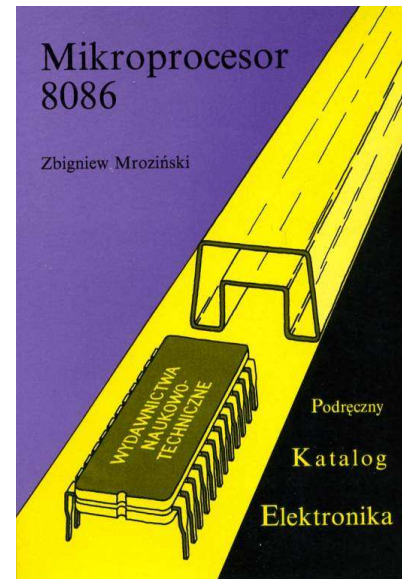
Mikroprocesor Intel 8088 (8086)

Literatura:

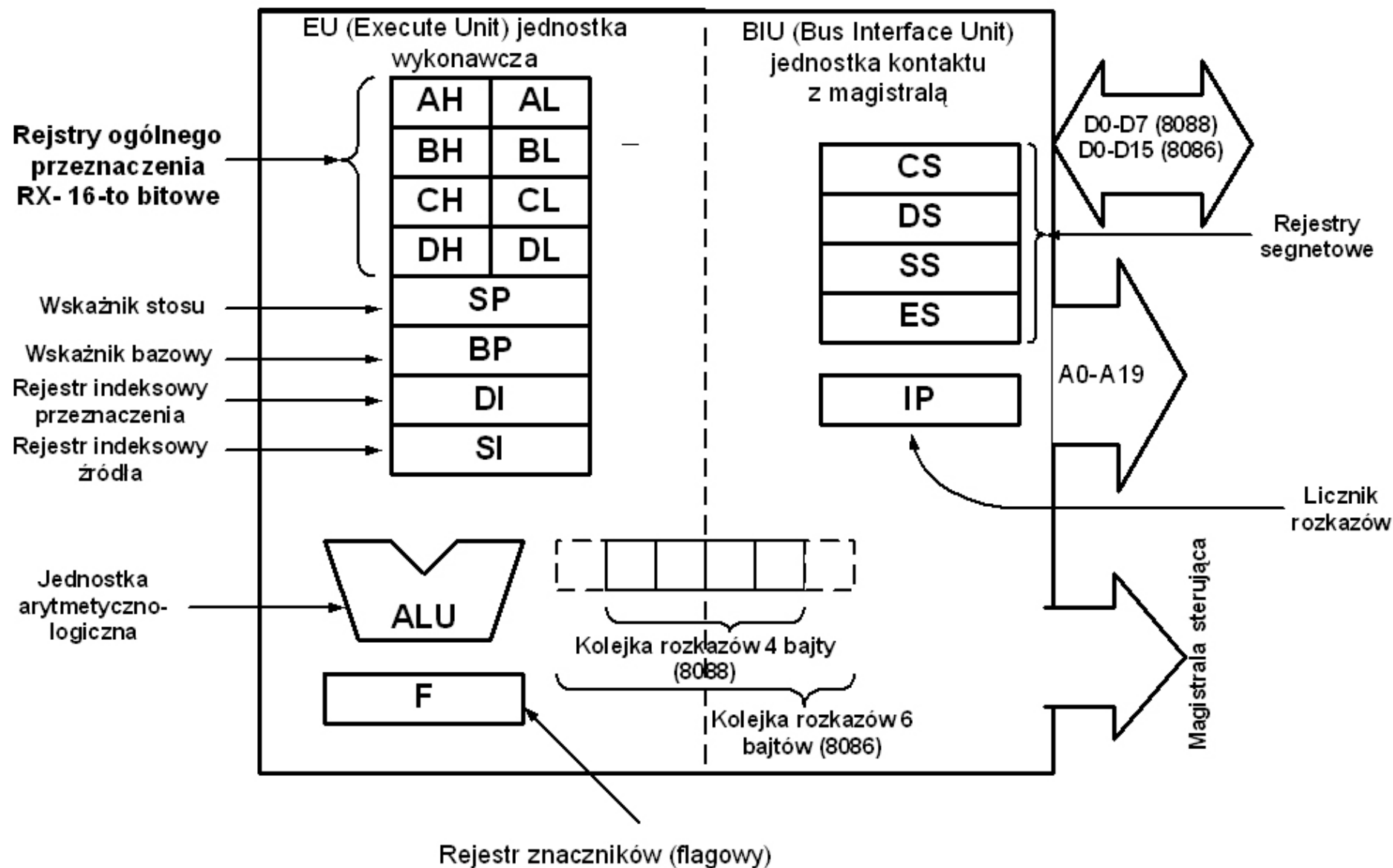
Mroziński Z.: Mikroprocesor 8086.
WNT, Warszawa 1992

iAPX 86,88 Users Manual Intel

80C86 Intersil 1997



[Źródło: www.swistak.pl]



Architektura wewnętrzna procesora

W przypadku obu tych układów architektura wewnętrzna jest 16-to bitowa.

**Zewnętrzna szyna danych jest dla 8088- 8-mio bitowa,
dla 8086- 16-to bitowa.**

Procesor został podzielony funkcjonalnie na dwie części:

-jednostkę kontaktu z magistralą- BIU, odpowiedzialną za pobieranie rozkazów i danych,

-jednostkę wykonawczą- EU, odpowiedzialną za wykonywanie rozkazów.

Dzięki temu obie te czynności są zrównoleglone. Powoduje to przyśpieszenie działania procesora.

Obie części są połączone **4-o (8088)** lub **8-mio bajtową (8086)** kolejką rozkazów typu **FIFO** (First Input, First Output)

Mikroprocesor może zaadresować do **1 M bajta pamięci programu i danych.**

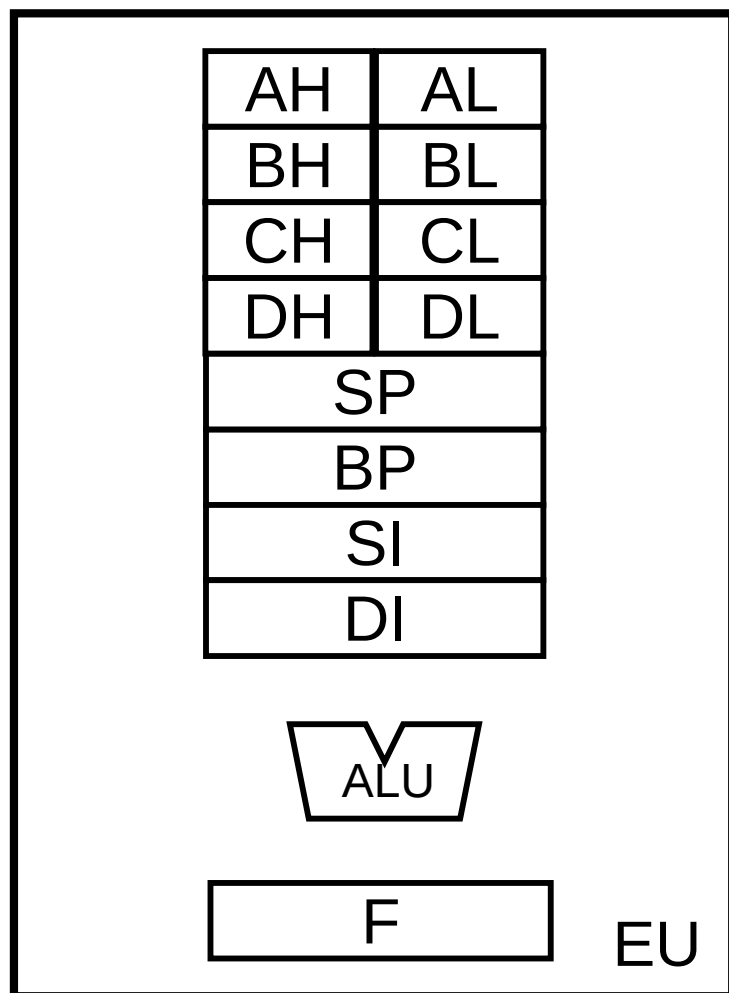
Przestrzeń adresowa **wejścia-wyjścia** wynosi **64 k bajty**

BIU:

Pobranie rozkażu N	Pobranie rozkażu N+1	Pobranie rozkażu N+2	
	Wykonanie rozkażu N	Wykonanie rozkażu N+1	Wykonanie rozkażu N+2

EU:

Współpraca jednostki kontaktu z magistralą i jednostki wykonawczej



AX

BX

CX

DX

Układ wykonawczy zawiera:

16-to bitową jednostkę arytmetyczno-logiczną,

Cztery 16-to bitowe rejestry użytkownika: **AX,BX,CX,DX** lub osiem rejestrów 8-mio bitowych: **AL,AH,BL,BH,CL,CH,DL,DH**.

Cztery 16-to bitowe rejestry adresacji:

-**SP**- wskaźnik stosu

-**BP**- wskaźnik bazy

-**SI**- rejestr indeksowy źródła danych

-**DI**- rejestr indeksowy przeznaczenia danych

Rejestr znaczników- **F**

Opis rejestrów użytkownika (dedykowane funkcje)

AX– akumulator- niektóre instrukcje mogą wykorzystywać tylko akumulator (np.. mnożenie, dzielenie)

BX– rejestr bazowy. Zawiera przemieszczenie względem początku segmentu danych

CX – rejestr licznikowy używany do zliczania liczby wykonań grupy instrukcji (np.: licznik pętli, przesłań blokowych)

DX – Używany jako rozszerzenie akumulatora przy 32-u bitowych Operacjach arytmetycznych

F-16-to bitowy rejestr znaczników

Pozostałe rejestry w jednostce wykonawczej

SP- wskaźnik stosu

BP- rejestr bazowy, który zawiera przemieszczenie
względem początku segmentu stosu

SI- rejestr indeksowy źródła. Zawiera adres źródła danych w
operacjach łańcuchowych

DI- rejestr indeksowy przeznaczenia. Zawiera adres
przeznaczenia danych w operacjach łańcuchowych

Wszystkie wymienione powyżej rejestry są 16-to bitowe.

Rejestr znaczników F

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	OF	DF	IF	TF	SF	ZF	0	AF	1	PF	1	CF

Można wydzielić dwa typy znaczników:

- znaczniki stanu wewnętrznego,
- znaczniki sterujące.

Do grupy **znaczników stanu wewnętrznego** zaliczamy:

C- znacznik przeniesienia lub pożyczki

P- znacznik parzystości (dla parzystej ilości jedynek w wyniku operacji jest ustawiany na 1)

A- przeniesienie połówkowe

Z- znacznik zera (wynik operacji=0)

S- znacznik znaku (ujemna liczba w kodzie U2)

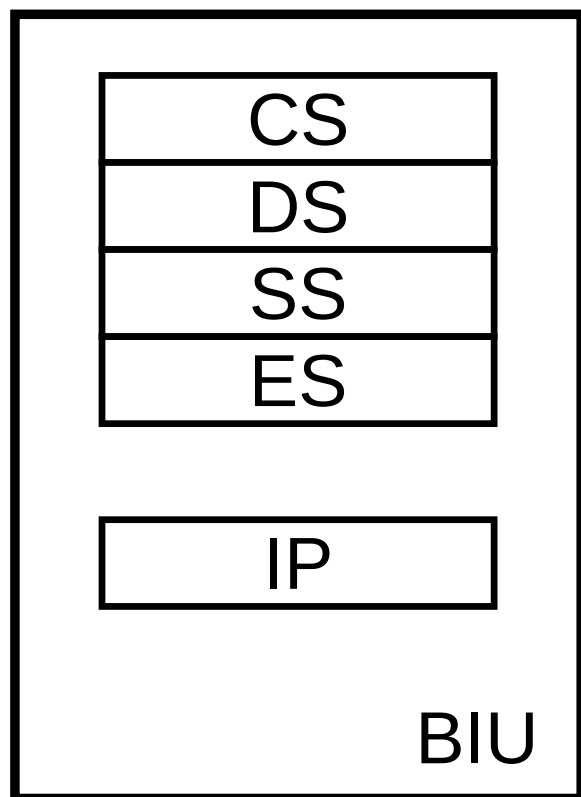
O- Przepełnienie w kodzie U2.

Znaczniki sterujące:

T- włączanie pracy krokowej (gdy $T=1$ praca krokowa włączona, po każdym rozkazie następuje wejście do procedury obsługi pracy krokowej)

I- maska przerwań maskowalnych (gdy $I=1$ przerwania odmaskowane)

D- znacznik kierunku przy operacjach blokowych ($D=1$ powoduje, że przy przesłaniu blokowym następuje zwiększanie adresu SI i DI, dla $D=0$ - zmniejszanie).



Układ kontaktu z magistralą BIU zawiera:

IP- 16-to bitowy rejestr licznika rozkazów,

Cztery rejestry segmentowe:

CS- rejestr segmentowy programu

DS- rejestr segmentowy danych

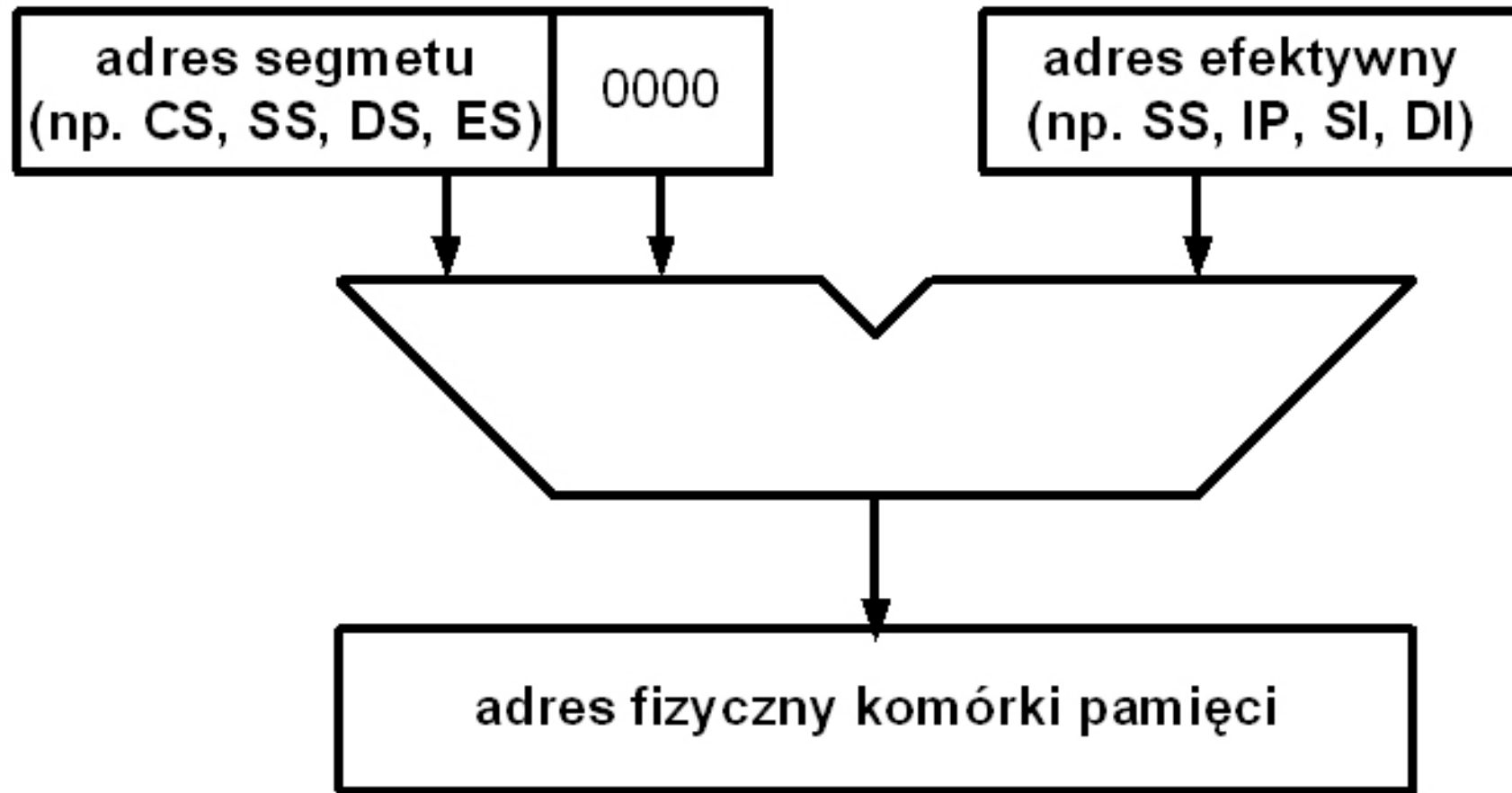
SS- rejestr segmentowy stosu

ES- rejestr segmentowy dodatkowy

Wszystkie rejestry wewnętrzne mikroprocesora są 16-to bitowe, podczas gdy magistrala adresowa mikroprocesora zawiera 20 linii adresowych A0-A19 czyli umożliwia adresację do 1MBajta pamięci.

Tak więc do utworzenia adresu fizycznego konieczne jest stosowanie dwóch rejestrów. Adres fizyczny jest tworzony przez sumowanie przesuniętego o cztery bity w lewo i dopełnionego zerami rejestru segmentowego i odpowiedniego rejestru zawierającego przesunięcie.

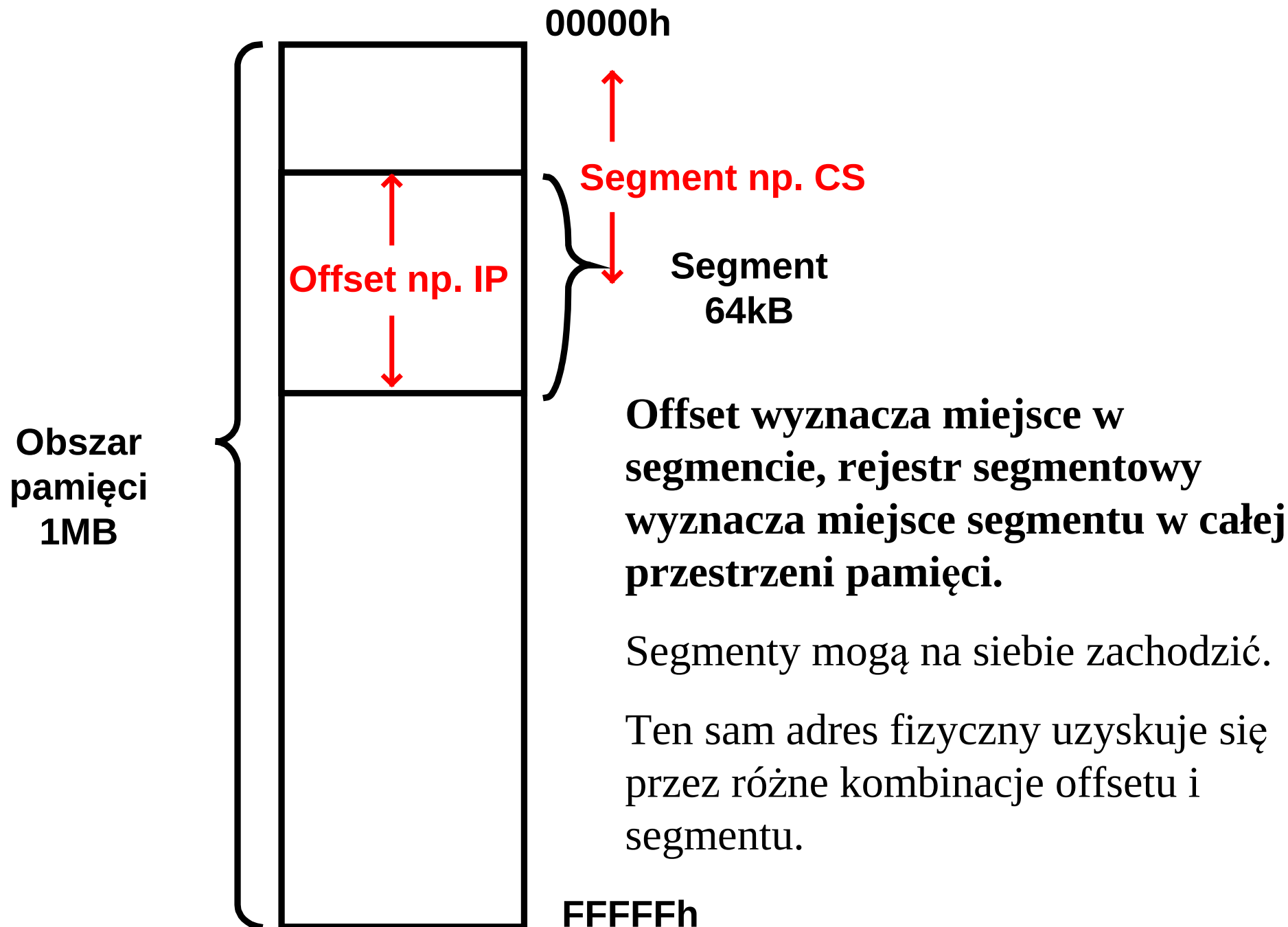
Np.: dla programu jest to rejestr segmentowy kodu **CS** i licznik rozkazów **IP**



Schemat powstawania adresu fizycznego na magistrali adresowej

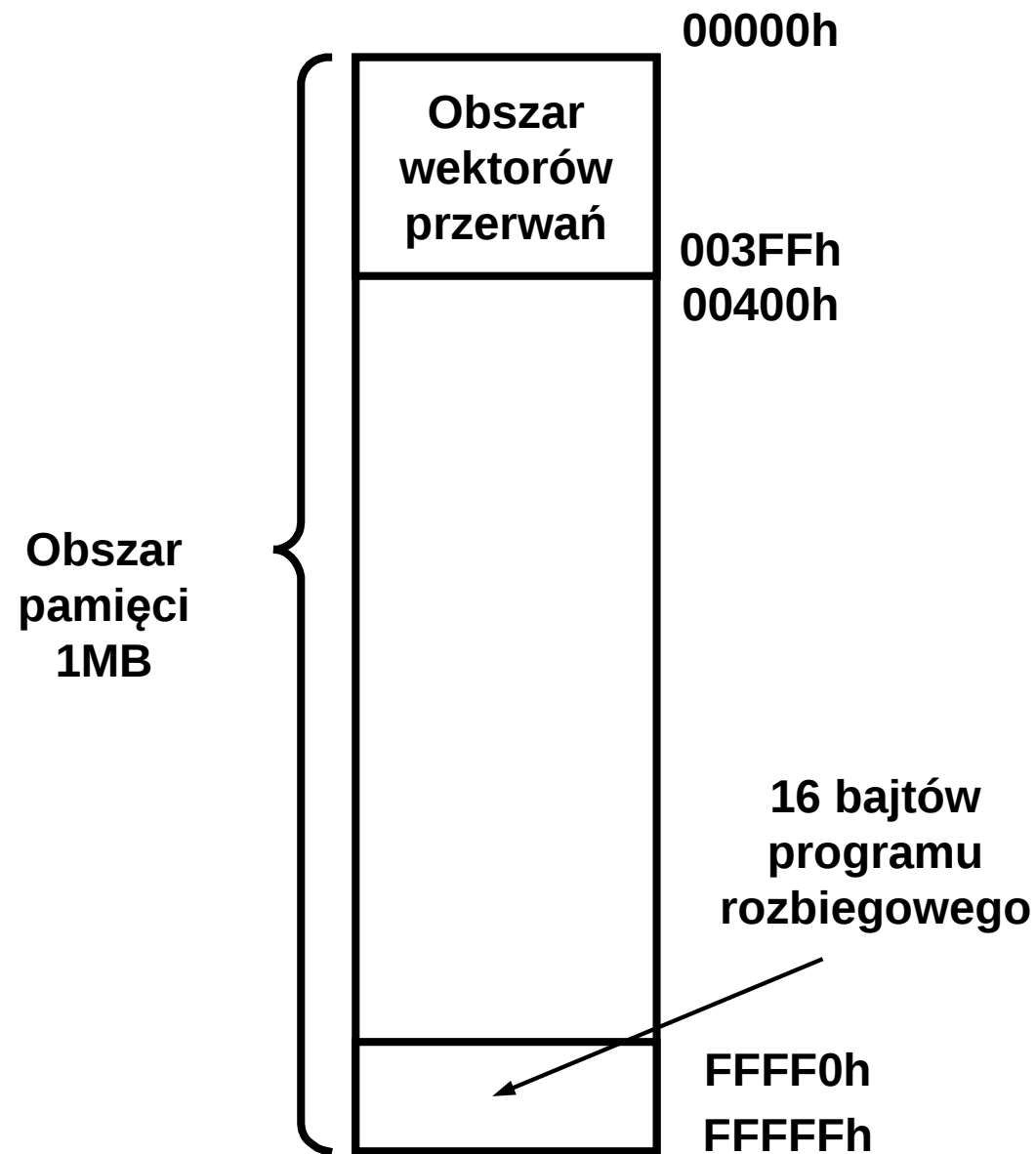
Powoduje to tzw. segmentację pamięci. Rozróżniamy cztery segmenty:

- Segment programu,
- Segment danych,
- Segment stosu
- Dodatkowy segment danych.



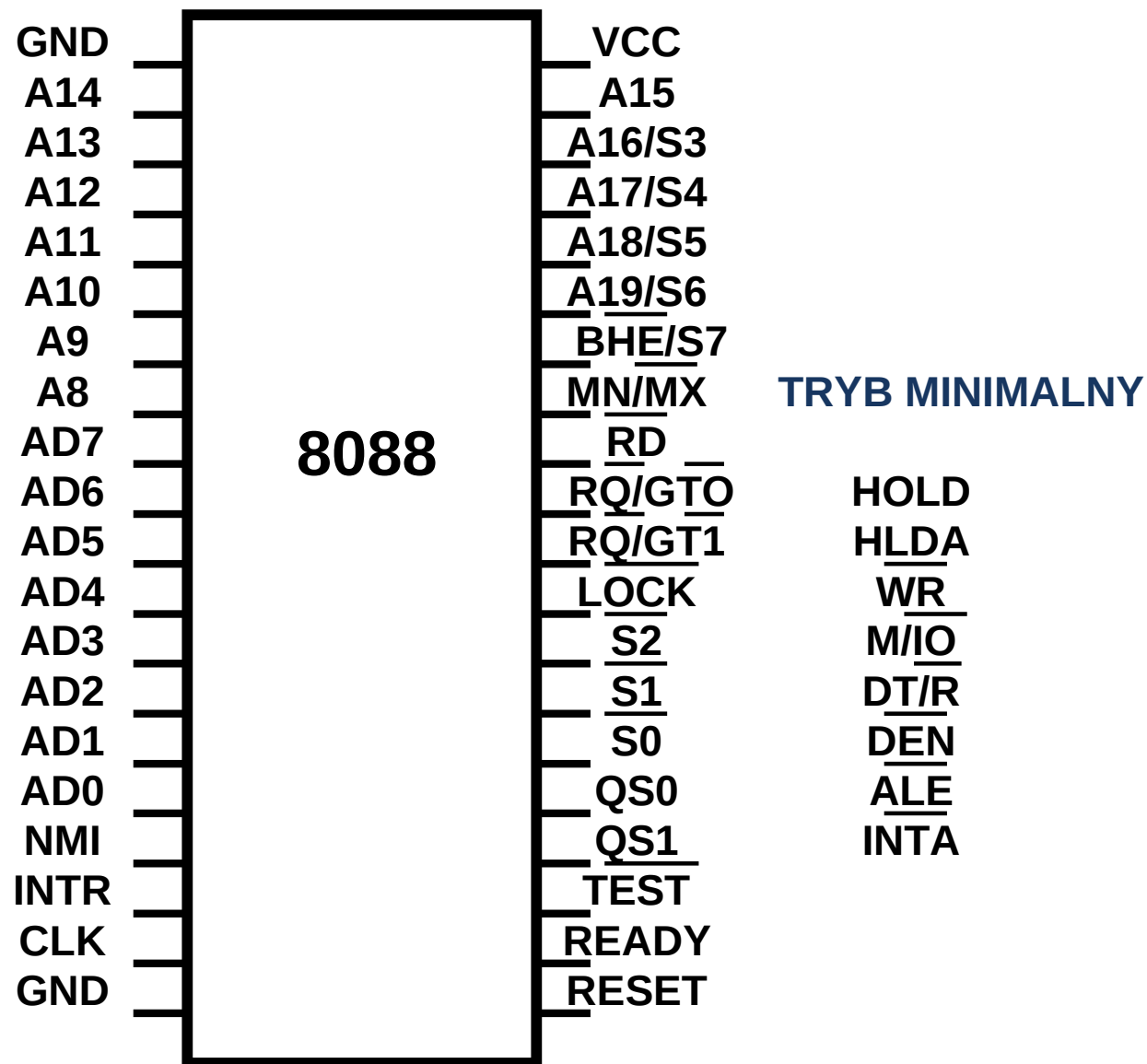
Typ operacji	Segment domyślny	Inne możliwe segmenty	Offset
Pobranie instrukcji	CS	-	PC
Operacja na stosie	SS	-	SP
Adres źródła dla łańcucha	DS	CS,SS,ES	SI
Adres przeznaczenia dla łańcucha	ES	-	DI
BP jako rejestr bazowy	SS	CS,SS,ES	DOWOLNY
BX jako rejestr bazowy	DS	CS,SS,ES	DOWOLNY
SI, DI jako indeks	DS	CS,SS,ES	DOWOLNY
Inna zmienna w pamięci	DS	CS,SS,ES	DOWOLNY

Rejestry do generowania 20-bitowego adresu fizycznego



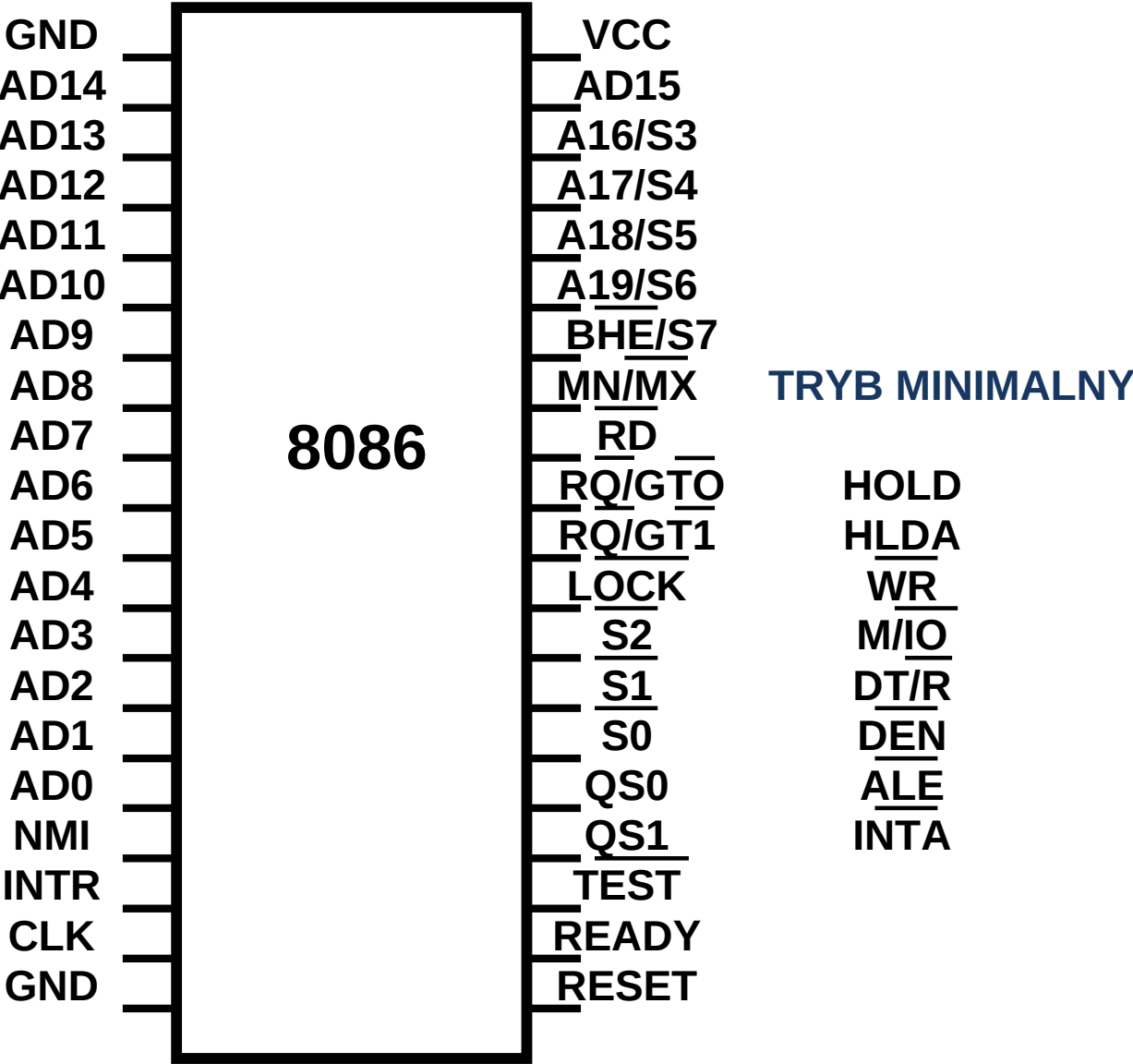
Organizacja przestrzeni pamięci

TRYB MAKSYMALNY



Sygnały mikroprocesora 8088

TRYB MAKSYMALNY



Sygnały mikroprocesora 8086

Procesor może pracować w dwóch trybach pracy:

- minimalnym bez dodatkowego wspomagania sprzętowego,
- maksymalnym, gdzie niezbędny jest dodatkowy sterownik magistrali do generacji sygnałów sterujących na podstawie linii statusowych.

Sygnały mikroprocesora w trybie minimalnym:

Sygnały wspólne dla trybu maksymalnego i minimalnego:

MN/MX (tryb Minimum/ tryb Maximum)- stan wejścia wyznacza tryb pracy mikroprocesora.

MN/MX=1 tryb minimalny, MN/MX=0 tryb maksymalny

CLK- sygnał zegarowy wyznaczający takt pracy mikroprocesora

READY- sygnał potwierdzenia gotowości pamięci lub urządzenia wejścia/wyjścia do zakończenia transmisji danych

RESET- zerowanie mikroprocesora, powoduje wyzerowanie rejestrów DS, SS, ES, rejestru znaczników (wyłączenie przerwań) i wpisanie do rejestru CS wartości FFFFh

INTR- sygnał żądania przerwania. Wejście wyzwalane poziomem, Gdy sygnał na wejściu jest aktywny, a system przerwania odblokowany (ustawiona flaga IF), mikroprocesor rozpoczyna cykl potwierdzenia przerwania INTA.

NMI- wejście przerwania niemaskowalnego, wyzwalane zboczem narastającym. Zmiana stanu sygnału z poziomu niskiego na wysoki powoduje zainicjowanie obsługi żądania przerwania.

TEST- stan tego wejścia jest związany z instrukcją WAIT. W przypadku wykonywania instrukcji WAIT, kiedy sygnał wejściowy TEST jest w stanie wysokim czeka aż nastąpi zmiana stanu tego sygnału na niski, po czym wykonuje następną instrukcję po WAIT. Używany do synchronizacji pracy z koprocesorem.

AD0 - AD15- 16-bitowa, multipleksowana magistrala danych i adresów, w przypadku 8088 multipleksowane są tylko linie AD0-AD15.

A19/S6, A18/S5, A17/S4, A16/S3- w trakcie pierwszego taktu cyklu dostępu do pamięci to cztery najbardziej znaczące linie adresu, w przypadku cyklu dostępu do urządzenia wejścia/wyjścia sygnały te wchodzą w stan niski. W pozostałych taktach cyklu, linie te zawierają 4 bity słowa statusu mikroprocesora:

-**S6** ma zawsze wartość zero

-**S5** zawiera stan flagi blokady systemu przerwań mikroprocesora INTE

-**S4 i S3**- kodują informacje o aktualnie wykorzystywanym segmencie pamięci.

S4	S3	segment
0	0	ES
0	1	SS
1	0	CS
1	1	DS

BHE/S7- (Bus High Enable) podczas trwania pierwszego taktu cyklu maszynowego, sygnał ten informuje czy ma zostać użyty bardziej znaczący bajt magistrali danych. W pozostałych taktach cyklu końcówka pełni funkcję wyjścia S7 które pozostaje aktywne w stanie niskim. Sygnał przechodzi w stan wysokiej impedancji w trakcie stanu zawieszenia mikroprocesora.

RD-sygnał strobujący dla odczytu danych z pamięci lub urządzenia wejścia/wyjścia. Aktywny w stanie niskim.

Wyprowadzenia, których funkcja zmienia się w zależności od wyboru trybu pracy mikroprocesora

Znaczenie końcówek dla trybu minimalnego:

WR- sygnał strobujący dla zapisu do pamięci lub urządzenia wejścia/wyjścia. Aktywny stanem niskim

M/IO – linia statusowa. Określa ona czy wykonywany cykl dotyczy pamięci czy urządzenia wejścia/wyjścia

M/IO=1- kontakt z pamięcią

M/IO=0- kontakt z urządzeniem we-wy

ALE- (Address Latch Enable)- opadające zbocze w trakcie trwania pierwszego taktu cyklu magistrali zatrzymuje zmultiplexowany z danymi adres komórki pamięci lub urządzenia wejścia/wyjścia w zewnętrznych rejestrach.

INTA- sygnał potwierdzenia przyjęcia przerwania. Wykorzystywany jako sygnał do wyprowadzeniu wektora przerwań z urządzenia zgłaszającego na magistralę danych

HOLD- sygnał żądania zwolnienia dostępu do magistrali przez urządzenie we-wy (aktywny stanem wysokim)

HLDA- sygnał potwierdzenia zwolnienia magistrali i przejścia mikroprocesora w stan zawieszenia (aktywny stanem wysokim)

DEN- sygnał sterujący buforami magistrali danych, otwierający je na czas przesyłania danych przez multipleksowaną magistralę danych i adresów

DT/~R- sygnał określający kierunek przesyłania danych, do sterowania zewnętrznymi buforów

DT/R=1- transmisja od mikroprocesora do pamięci lub urządzeń we-wy

DT/R=0- transmisja od pamięci lub urządzeń we-wy do mikroprocesora

Znaczenie końcówek dla trybu maksymalnego

LOCK - sygnał informujący zewnętrzne urządzenie mogące żądać dostępu do magistrali (za pomocą stanu niskiego), że żądanie zwolnienia magistrali zostało zablokowane do czasu wykonania instrukcji z prefiksem LOCK.

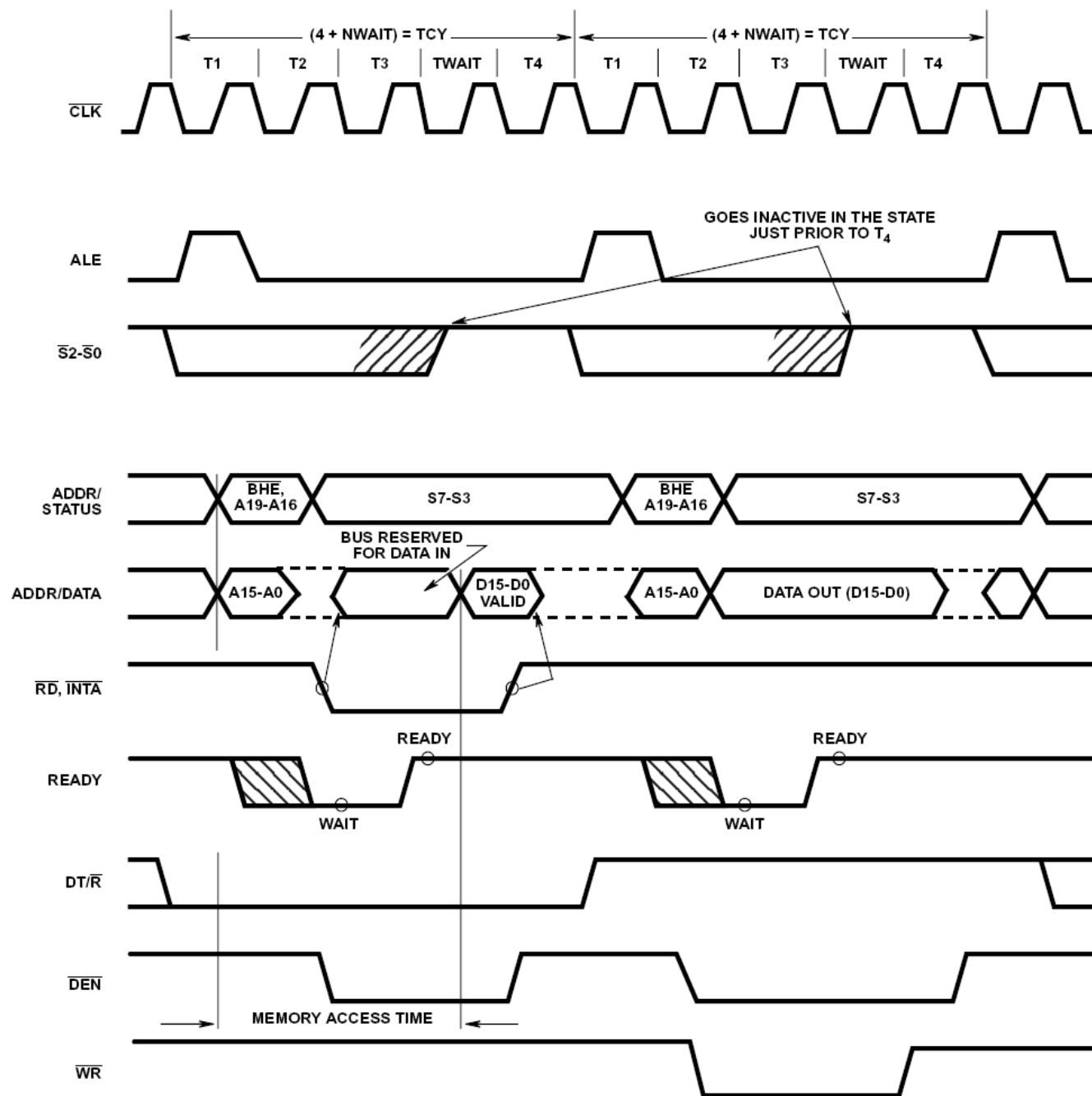
S0, S1, S2 - sygnały te określają rodzaj cyklu maszynowego, wykonywanego przez mikroprocesor.

S2	S1	S0	Cykl magistrali
0	0	0	Przyjęcie przerwania
0	0	1	Odczyt urządzenia we-wy
0	1	0	Zapis urządzenia we-wy
0	1	1	Zatrzymanie HALT
1	0	0	Pobranie rozkazu
1	0	1	Odczyt pamięci
1	1	0	Zapis pamięci
1	1	1	Brak cyklu

QS0, QS1- sygnały określające stan kolejki rozkazów mikroprocesora

QS1	QS0	Stan kolejki
0	0	Brak operacji
0	1	Pobranie pierwszego bajtu rozkazu
1	0	Kolejka pusta
1	1	Pobranie następnego bajtu rozkazu

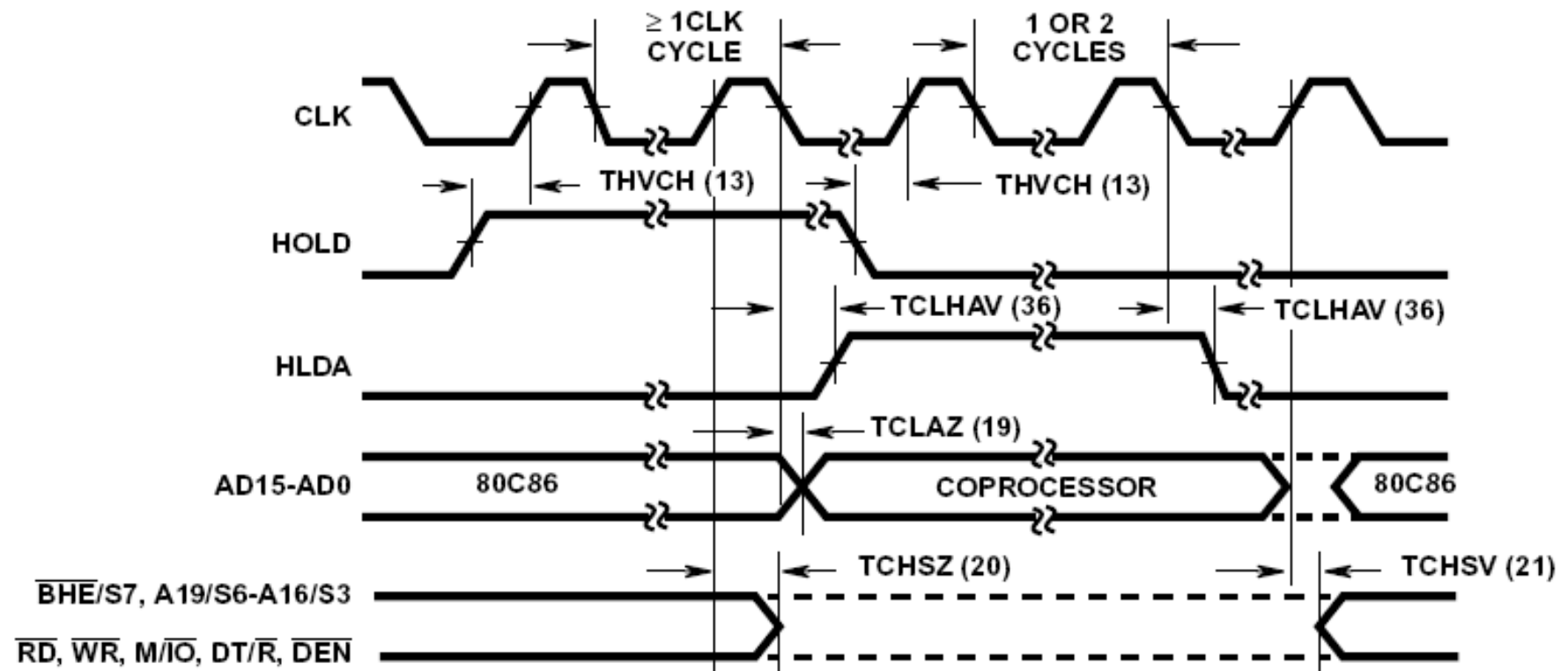
RQ/GT0 oraz RQ/GT1- sygnały wykorzystywane są do wymuszenia zwolnienia magistrali przez procesor,. Sygnał RQ/GT0 ma wyższy priorytet od sygnału RQ/GT1. Pojedynczy impuls ujemny na linii RQ/GTi sygnalizuje żądanie zwolnienia magistrali, jeśli rozkaz nie był poprzedzony prefiksem LOCK i został dokończony bieżący cykl magistrali, procesor oddaje magistralę i potwierdza to ujemnym impulsem na linii RQ/GTi. Koniec przejęcia magistrali jest sygnalizowany przez urządzenie pojedynczym impulsem ujemnym na tej samej linii ($i=0,1$)



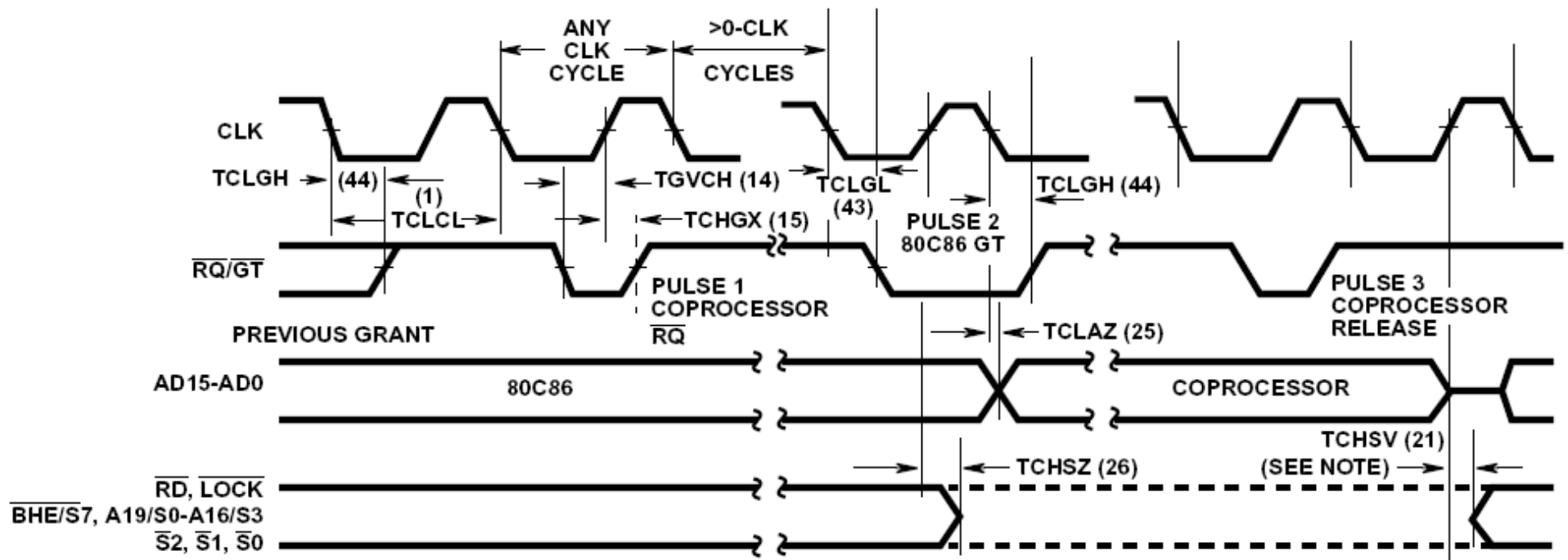
Przebiegi czasowe
na magistrali

[Źródło: <http://www.dsi.unifi.it/~nesi/8086.pdf>]

FIGURE 3. BASIC SYSTEM TIMING

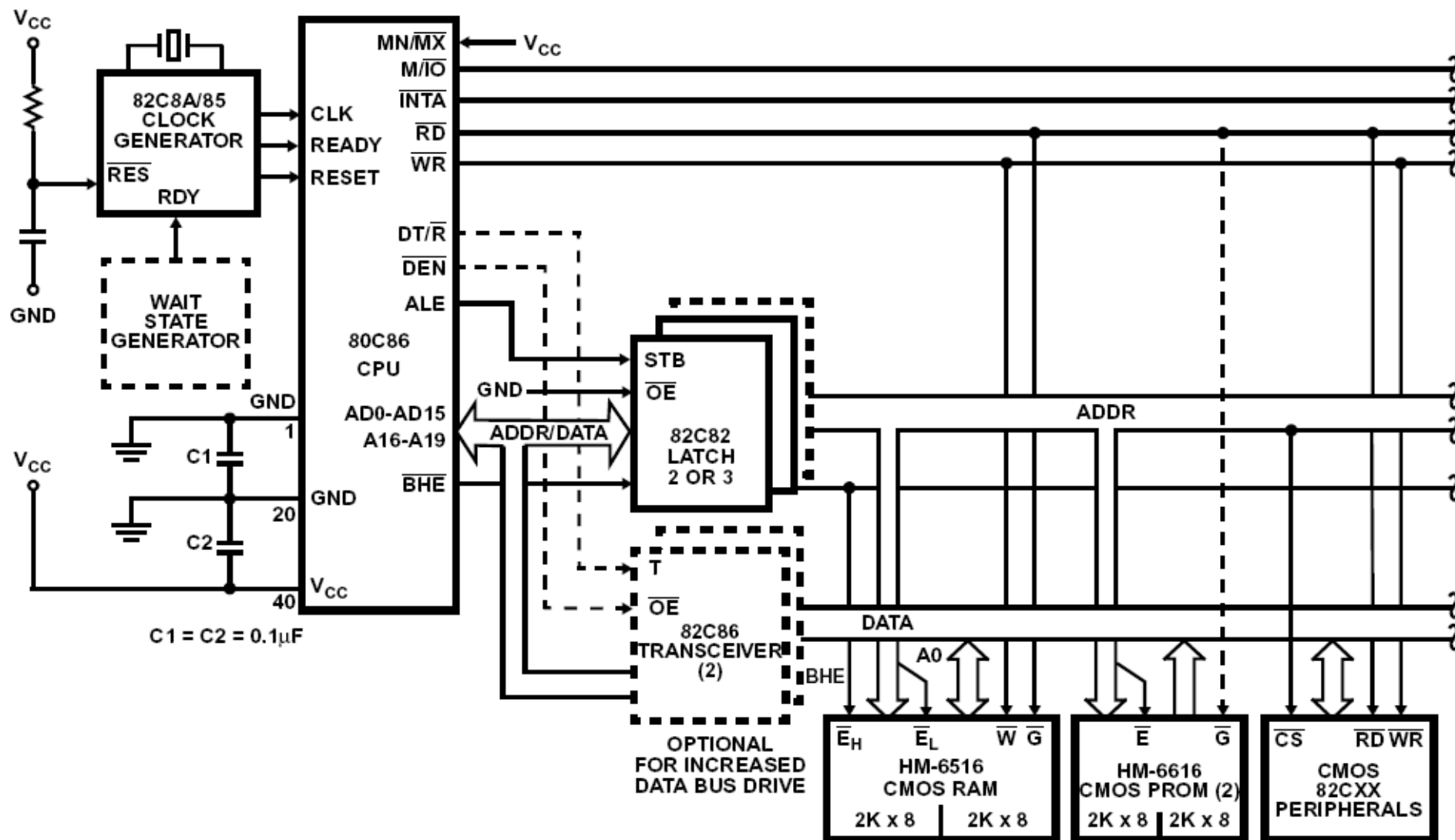


Cykl przejęcia magistrali w trybie minimalnym



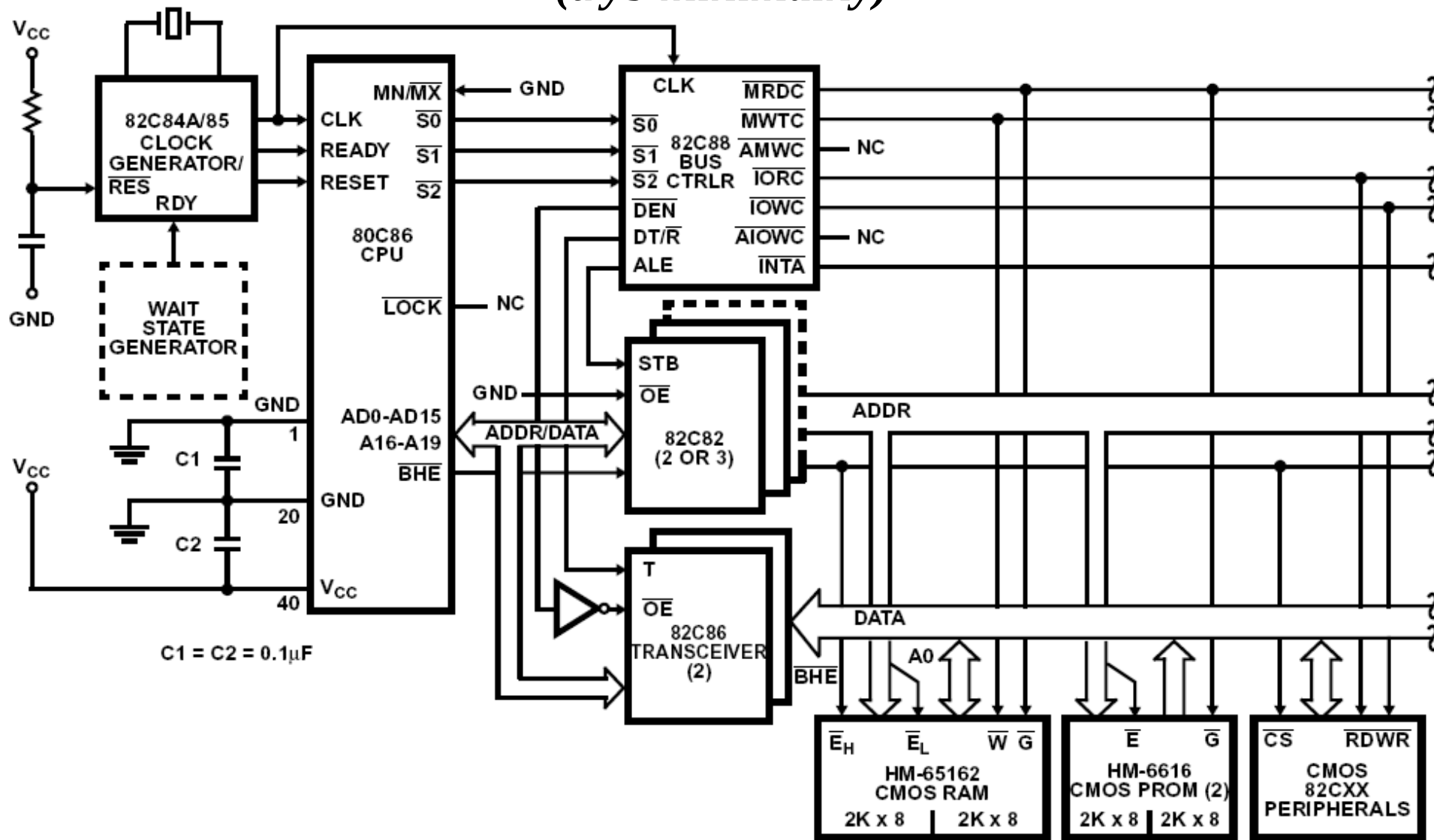
Cykl przejęcia magistrali w trybie maksymalnym

Współpraca mikroprocesora 8086 z pamięcią i urządzeniami we-wy (tryb minimalny)



[Źródło: <http://www.dsi.unifi.it/~nesi/8086.pdf>]

Współpraca mikroprocesora 8086 z pamięcią i urządzeniami we-wy (tryb minimalny)



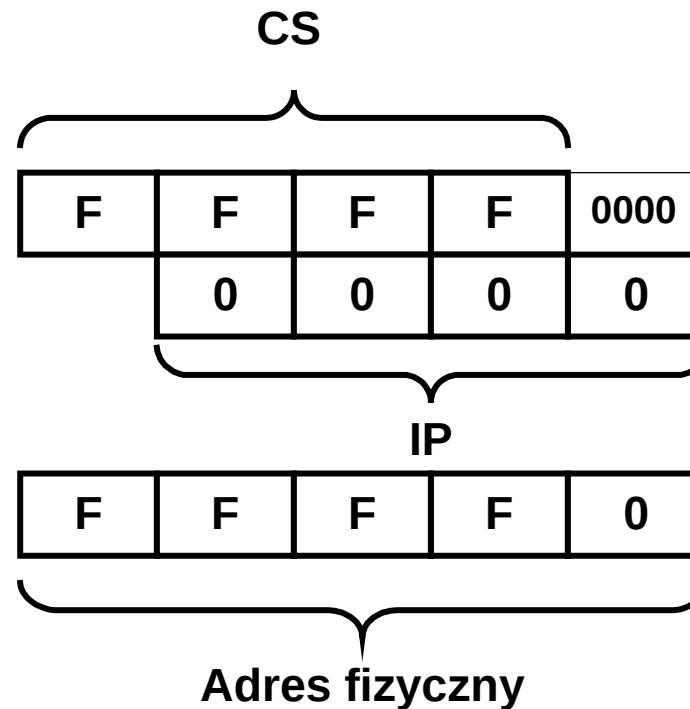
[Źródło: <http://www.dsi.unifi.it/~nesi/8086.pdf>]

Stan mikroprocesora po sygnale RESET

Sygnal RESET jest aktywny stanem wysokim, powoduje wyzerowanie mikroprocesora.

Licznik rozkazów IP jest zerowany **IP=0000h**

Rejestr segmentowy rozkazów **CS=FFFFh**



Czyli adres fizyczny na magistrali adresowej po sygnale RESET wynosi-
FFFF0H

Po sygnale RESET bit maski przerwań **I** w rejestrze znaczników **F** jest zerowany (przerwania maskowalne- zamaskowane).

Przerwania w mikroprocesorze 8086

Mikroprocesor posiada dwa sygnały do zgłaszania przerwań:

NMI- przerwanie niemaskowalne aktywne narastającym zboczem,

INTR- przerwanie maskowalne, aktywne wysokim stanem logicznym.

System przerwań procesora jest systemem wektorowym.

Jeśli zgłoszone przerwanie maskowalne jest odmaskowane ($I=1$), procesor kończy bieżący rozkaz i przechodzi do trybu akceptacji przerwania.

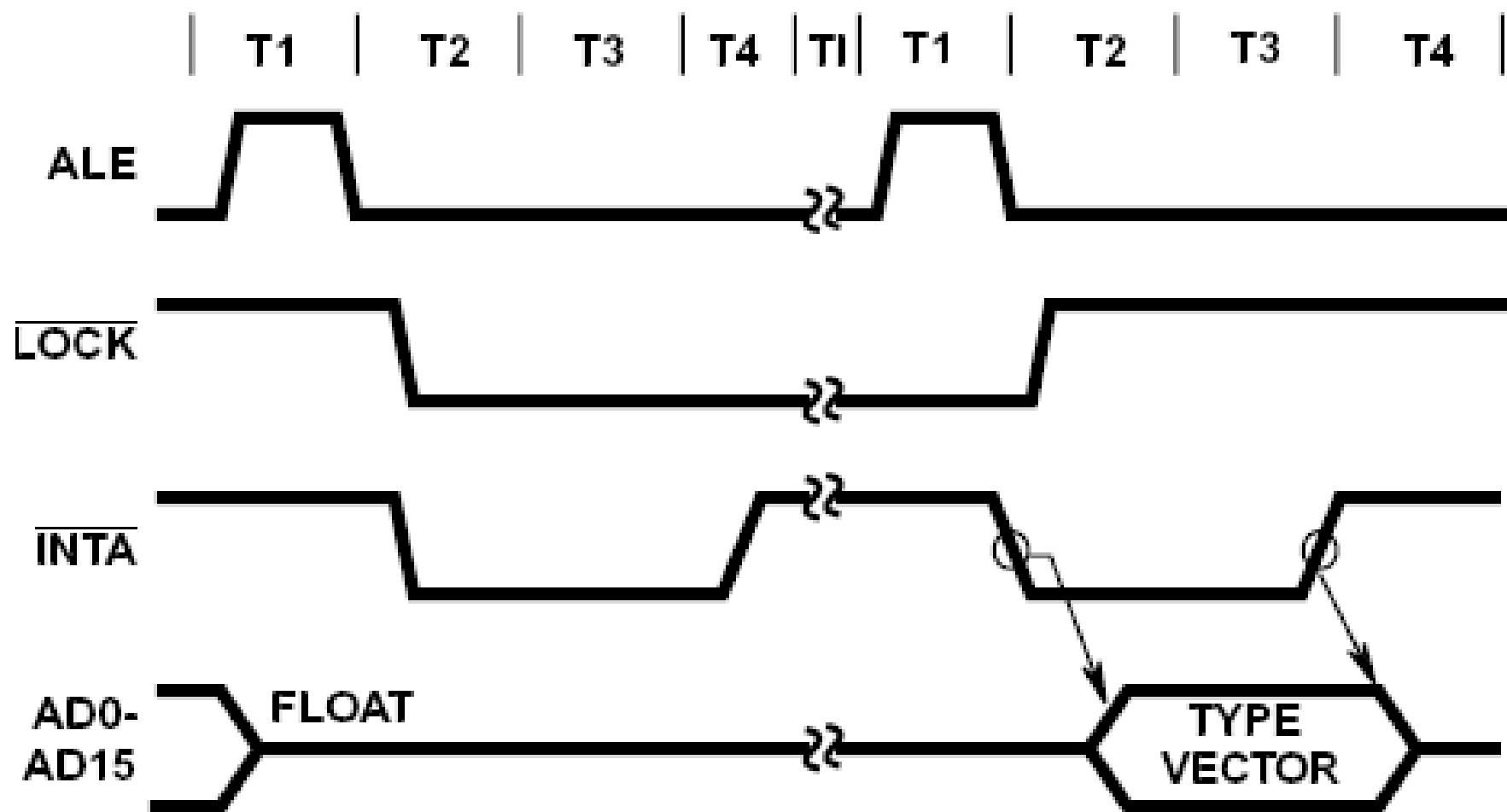


FIGURE 5. INTERRUPT ACKNOWLEDGE SEQUENCE

Cykl przyjęcia przerwania maskowalnego

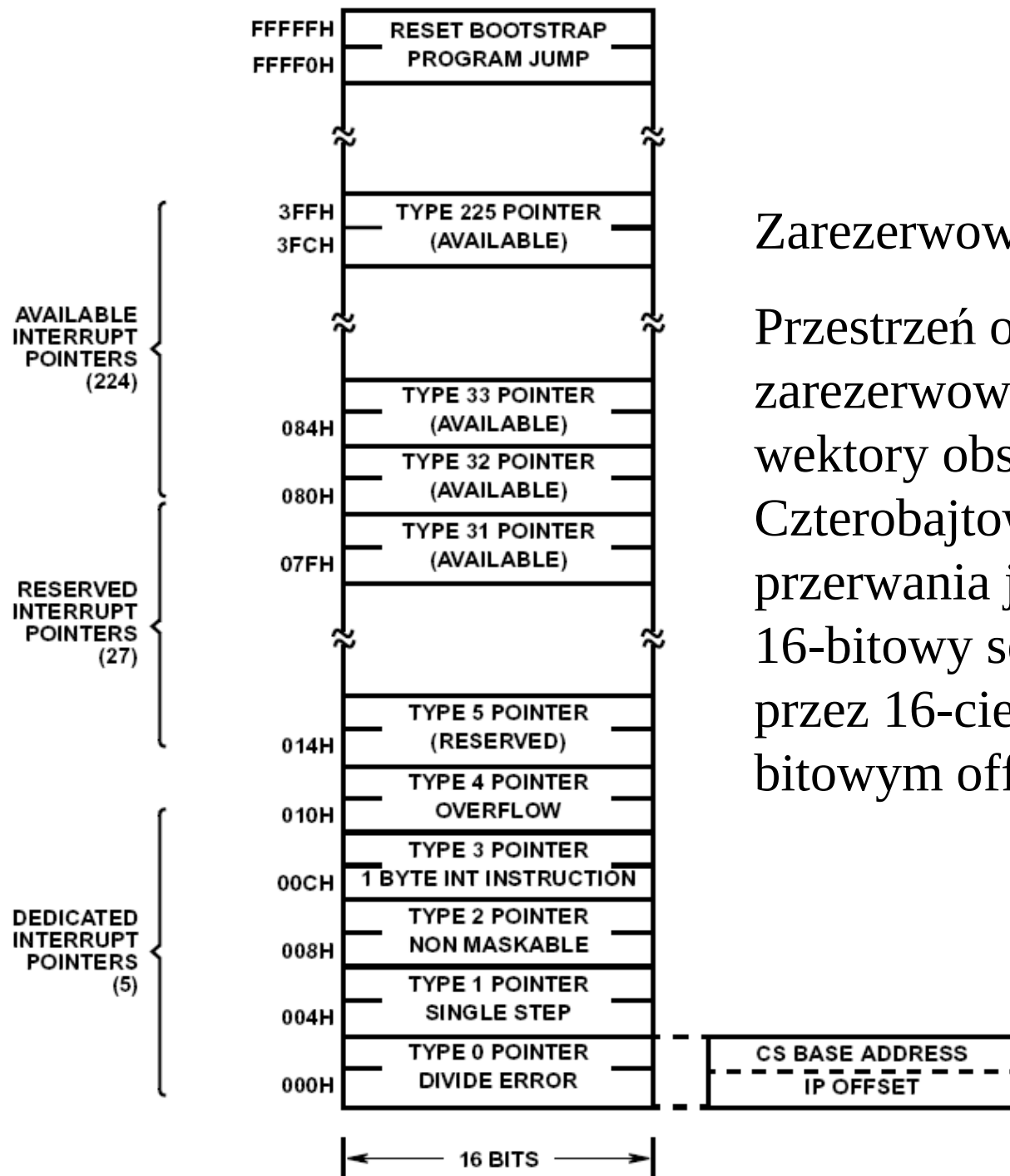
[Źródło:<http://www.dsi.unifi.it/~nesi/8086.pdf>]

Cykl akceptacji przerwania jest podwójny:

Składa się z dwóch INTA:

-pierwszy służy do **rozstrzygnięcia priorytetu przerw** przez zewnętrzny kontroler priorytetu,

-drugi jest sygnałem do **wystawienia** przez urządzenie na magistrali danych **8-mio bitowego wektora obsługi przerwania: 00-FFh**.

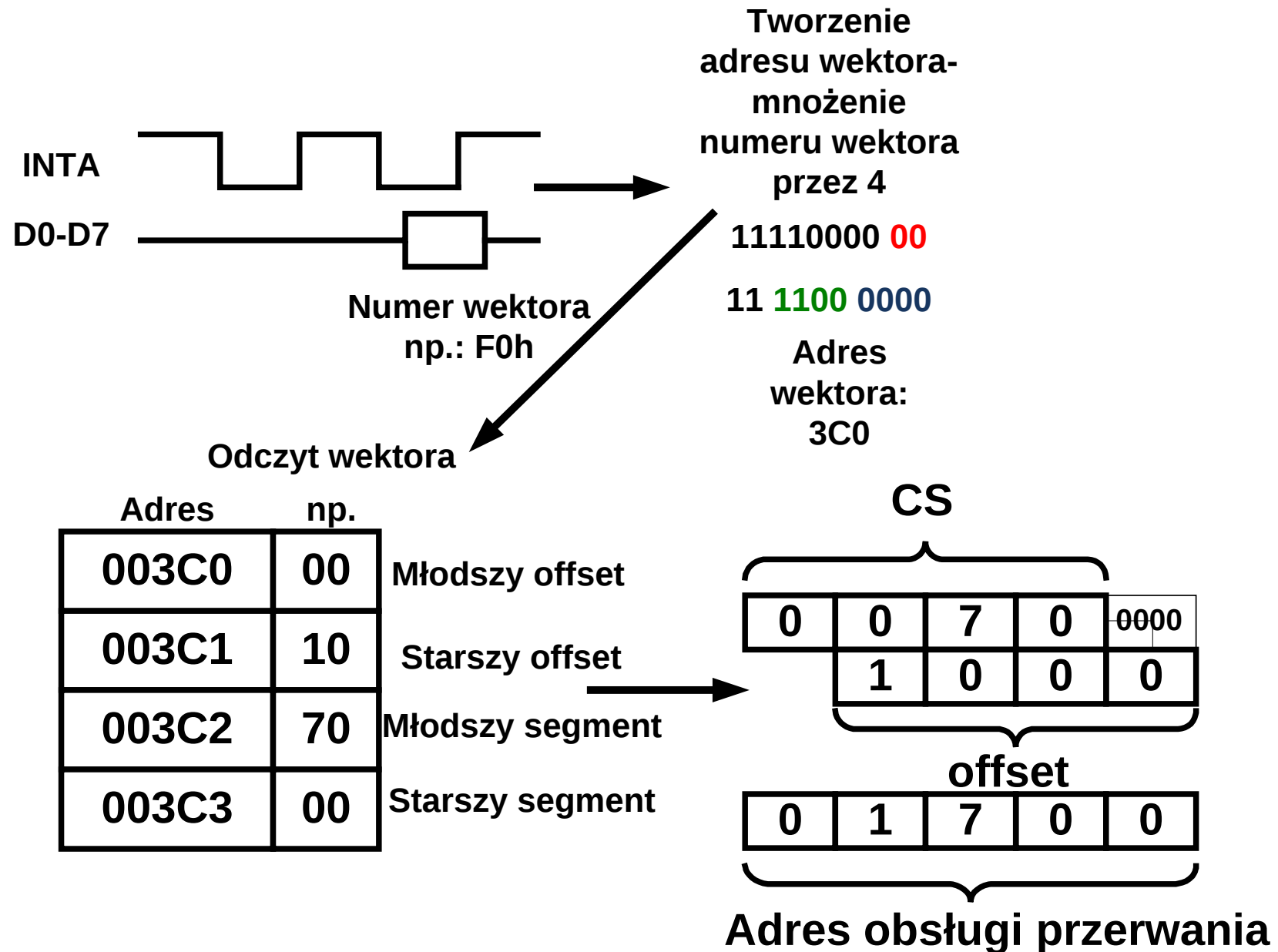


Zarezerwowane obszary pamięci

Przestrzeń od 00000h do 003FFh jest zarezerwowany na czterobajtowe wektory obsługi przerwania.

Czterobajtowe gdyż adres obsługi przerwania jest tworzony wg zasady 16-bitowy segment, wymnożony przez 16-cie zsumowany z 16-to bitowym offsetem.

Schemat tworzenia adresu procedury obsługi przerwania.



Niektóre wektory przerwań są dedykowane określonym źródłom przerwania:

Wektor 0- dzielenie przez zero

Wektor 1- praca krokowa T

Wektor 2- NMI

Wektor 3- jednobajtowa instrukcja przerwania programowego

Wektor 4- przepełnienie

Wektor 5-30: zarezerwowane

Wektory 31-255- dostępne dla przerw zewnętrznych.

Najwyższy priorytet ma przerwanie o wektorze 0, najniższy 255.

Obsługa przerwania

W trakcie obsługi przerwania mikroprocesor składa na stosie 6-bajtów:

2 bajty rejestru znaczników F

2 bajty rejestru segmentowego CS

Przerwa- następuje pobranie pierwszego bajtu kodu rozkazu z procedury obsługi przerwania

2 bajty licznika rozkazów IP

Tryby adresacji mikroprocesora 8086

Tryb natychmiastowy- argument umieszczono bezpośrednio za kodem rozkazu np.: `MOV AX,1233`.

Tryb rejestrowy- argumenty znajdują się w rejestrach mikroprocesora np.: `MOV AX,1233`.

Tryb bezpośredni- adres argumentu znajduje się za kodem rozkazu np.: `MOV AX,[1234]`

Tryb pośredni- adres argumentu znajduje się w jednym z rejestrów mikroprocesora np.: `MOV AX,[CX]`

Tryb bazowy- jest rodzajem adresowania pośredniego, adres argumentu jest sumą zawartości rejestru bazowego BX lub BP i 8-bitowego lub 16-bitowego przesunięcia np.: `Np. MOV AX, [BP+1234]`.

Tryb indeksowy- jest rodzajem adresowania pośredniego, adres efektywny jest sumą zawartości rejestru indeksowego SI lub DI i przesunięcia. np.: MOV AX, [SI+1234].

Tryb bazowo-indeksowy- adres efektywny jest sumą zawartości jednego z rejestrów bazowych, jednego z rejestrów indeksowych i przesunięcia. Np. MOV AX, [SI+BP+2345].

Rozkazy przesłań blokowych, operujące na ciągach słów posługują się rejestrami indeksowymi. Rejestry SI i DI zawierają adresy efektywne pierwszego słowa odpowiednio w bloku źródłowym i wynikowym. Po każdej transmisji wartości rejestrów indeksowych są automatycznie inkrementowane lub dekrementowane, jest to zależne od stanu bitu DF w rejestrze F.

Rozkazy WE/WY posługują się trybem adresacji natychmiastowej i pośredniej.

Struktura rozkazów mikroprocesora 8086

Rozkazy mikroprocesora 8086 składają się w zależności od rodzaju rozkazu z od jednego do sześciu bajtów.

7	6	5	4	3	2	1	0
Kod operacji						D	W
MOD		REG			R/M		

Pierwszy bajt rozkazu zawiera:

- sześciobitowy kod operacji,
- bit D- określający kierunek transmisji: *0* - wynik operacji przesyłany z rejestru do pamięci, *1* - z pamięci do rejestru,
- bit W- określający długość operandu rozkazu: *0* - operacje bajtowe, *1* - operacje na 16-bitowym słowie.

7	6	5	4	3	2	1	0
Kod operacji						D	W
MOD		REG			R/M		

W rozkazach wielobajtowych drugi bajt rozkazu określa sposób adresowania argumentów. Zawiera on trzy grupy bitów:

- MOD**- określający tryb adresowania,
- REG**- określający rejestr w którym znajduje się argument,
- R/M**- określający sposób wyznaczania miejsca argumentu.

**Określanie rejestru w którym znajduje się argument poprzez
zawartość pól REG i W**

REG	W=0	W=1
000	AL.	AX
001	CL	BX
010	DL	CX
011	BL	DX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

Wyznaczanie adresu argumentu znajdującego się w pamięci przy pomocy zawartości pól MOD i R/M

R/M	MOD=0	MOD=1	MOD=2
000	BX+SI	BX+SI+p8	BX+SI+p16
001	BX+DI	BX+DI+p8	BX+DI+p16
010	BP+SI	BP+SI+p8	BP+SI+p16
011	BP+DI	BP+DI+p8	BP+DI+p16
100	SI	SI+p8	SI+p16
101	DI	DI+p8	DI+p16
110	BP	BP+p8	BP+p16
111	BX	BX+p8	BX+p16

Struktura rozkazu 6-cio bajtowego

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOD		REG			R/M			Kod operacji						D	W
przesunięcie															
dana															

Rodzaje rozkazów w mikroprocesorze 8086

- przesłań,
- arytmetyczno-logiczne,
- skoków, obsługi pętli, wywołań i powrotów z podprogramu,
- rozkazy dotyczące rejestrów segmentowych,
- wykonujące operacje na ciągach słów
- wejścia/wyjścia
- inne np. Prefixy rozkazów

Rozkazy prefiksujące

Są to rozkazy jednobajtowe, których zastosowanie zmienia sposób wykonania następującego po nich rozkazu. Nie powodują żadnego ustawienia znaczników. Do rozkazów takich zaliczamy:

SEG- zmienia rejestr segmentowy przypisany danemu rozkazowi przy tworzeniu adresu fizycznego. Np.: w rozkazie **MOV AX,[BX]** do tworzenia adresu fizycznego służy rejestr segmentowy DS., w przypadku ciągu instrukcji:

SEG CS

MOV AX,[BX] adres fizyczny będzie tworzony przy udziale rejestru CS.

LOCK- zabezpiecza cały następujący po nim rozkaz przed przejęciem magistrali przez inne urządzenie. W normalnych warunkach przejęcie magistrali jest możliwe po zakończeniu każdego cyklu magistrali tworzącego cykl rozkazowy. Dodatkowo ochrona rozkazu jest sygnalizowana na linii wyjściowej LOCK np.:

LOCK

MOV SP,#1000h

REP- przedrostek, który umieszczony przed rozkazem np. przesłania blokowego zapewnia jego powtarzanie aż do wyzerowania się licznika CX np.:

REP

MOVSB

Rozkazy przesłań

Pomiędzy rejestrem i pamięcią

MOV- przesłanie bajtu lub słowa

PUSH- przesłanie słowa na stos

POP- odczyt słowa ze stosu

XCHG- zamiana miejscami bajtu lub słowa

XLAT- pobranie bajtu z tablicy

Ładowanie adresów

LEA- ładowanie adresu efektywnego

LDS- ładuj adres logiczny do wskazanego rejestru oraz DS.

LES- ładuj adres logiczny do wskazanego rejestru i ES

Do i z układu we-wy

IN- odczyt urządzenia we-wy

OUT- zapis urządzenia we-wy

Operacje na rejestrze F

LAHF- przesłanie wskazanego bajtu rejestru F do AH

SAHF- przesłanie AH do rejestru znaczników

PUSHF- złożenie na stosie rejestru F

POPF- ściągnięcie ze stosu rejestru F

Rozkazy arytmetyczne

ADD- dodawanie bajtów lub słów dwubajtowych

ADC- dodawanie bajtów lub słów dwubajtowych z uwzględnieniem C

INC- zwiększ bajt lub słowo o 1

AAA- korekcja ASCII po dodawaniu

DAA- korekcja dziesiętna dla dodawania spakowanej liczby w kodzie BCD

SUB- dodawanie bajtów lub słów dwubajtowych

SBB- dodawanie bajtów lub słów dwubajtowych z uwzględnieniem C

DEC- zmniejsz bajt lub słowo o 1

AAS- korekcja ASCII po odejmowaniu

DAS- korekcja dziesiętna dla odejmowania spakowanej liczby w kodzie BCD

NEG- zaneguj bajt lub słowo

CMP- porównaj bajty lub słowa

MUL- mnożenie dwóch bajtów lub słów bez znaku

IMUL- mnożenie dwóch bajtów lub słów ze znakiem

AAM- korekcja dziesiętna ASCII po mnożeniu

DIV- dzielenie dwóch bajtów lub słów bez znaku

IDIV- dzielenie dwóch bajtów lub słów ze znakiem

AAD- korekcja dziesiętna ASCII po dzieleniu

CBW- zamiana bajtu na słowo

CWD- zamiana słowa na długie słowo (32-bity)

Rozkazy logiczne

NOT- negacja bajtu lub słowa

AND- iloczyn logiczny bajtów lub słów

OR- suma logiczna bajtów lub słów

XOR- operacja exclusive-or bajtów lub słów

TEST- porównanie logiczne bajtów lub słów

Rozkazy przesunięć i obrotów

SHL- przesunięcie logiczne bajtu lub słowa w lewo

SAL- przesunięcie arytmetyczne bajtu lub słowa w lewo

SHR- przesunięcie logiczne bajtu lub słowa w prawo

SAR- przesunięcie arytmetyczne bajtu lub słowa w prawo

ROL- obrót w lewo bajtu lub słowa

ROR- obrót w prawo bajtu lub słowa

RCL- obrót w lewo bajtu lub słowa z przeniesieniem C

RCR- obrót w prawo bajtu lub słowa z przeniesieniem C

Rozkazy blokowe

MOVS- prześlij bajt lub słowo z jednego bloku danych do drugiego

LODS- prześlij do akumulatora wybrany element bloku danych

STOS- zapis zawartości akumulatora do bloku danych

SCAS- porównanie elementu bloku danych z zawartością akumulatora

CMPS- porównanie elementów dwóch bloków danych

W operacjach blokowych-

adres źródła określają rejestry: **DS** i **SI**

adres przeznaczenia określają rejestry **ES** i **DI**

licznikiem operacji jest rejestr **CX**

Rozkazy powtórzeń (przedrostki)

REP- powtarzaj operację po prefiksie do wyzerowania CX

REPZ/REPE- powtarzaj operację po prefiksie do wyzerowania CX lub do ustawienia bitu Z ($Z=1$) w rejestrze F, dla instrukcji SCAS, CMPS zakończenie następuje gdy $Z=0$.

REPNZ/REPNE-powtarzaj operację po prefiksie do wyzerowania CX lub do skasowania bitu Z ($Z=0$) w rejestrze F, dla instrukcji SCAS, CMPS zakończenie następuje gdy $Z=0$.

Rozkazy skoków

Bezwarunkowe

JMP- skok bezpośredni, bezwarunkowy

CALL- wywołanie podprogramu bezwarunkowe

RET- powrót z podprogramu

Skoki warunkowe

JC,/JNC- skok od bitu przeniesienia C ($C=1/C=0$)

JO,/JNO- skok od bitu przepełnienia O ($O=1/O=0$)

JP,/JNP- skok od bitu parzystości P ($P=1/P=0$)

JS,/JNS- skok od bitu znaku S ($S=1/S=0$)

JZ,/JNZ- skok od bitu zerowości Z ($Z=1/Z=0$)

Skoki warunkowe złożone

JE/JNE- skok gdy równe/nierówne ($Z=1/Z=0$)

JB/JNB- skok gdy poniżej/nie poniżej ($C=1/C=0$)

JG/JNG- skok jeśli większy/nie większy $(S \oplus O) \vee (Z=0) / (S \oplus O) \vee (Z=1)$

JGE/JNGE- skok jeśli większy lub równy/nie większy lub równy
 $(S \oplus O=0)/(S \oplus O=1)$

JL/JNL- skok jeśli mniejszy/nie mniejszy $(S \oplus O = 1)/(S \oplus O = 0)$

JLE/JNLE- skok jeśli mniejszy lub równy/nie mniejszy lub równy
 $(S \oplus O) \vee (Z = 1) / (S \oplus O) \vee (Z = 0)$

JA/JNA- skok jeśli powyżej/nie powyżej $(C \vee Z = 0)/(C \vee Z = 1)$

JAЕ/JNAЕ- skok jeśli powyżej lub równo/nie powyżej lub równo
 $(C = 0)/(C = 1)$

JBE/JNBE- skok jeśli poniżej lub równo/nie poniżej lub równo
 $(C \vee Z = 1)/(C \vee Z = 0)$

Rozkazy przerwań programowych

INT- skok do procedury obsługi przerwania

INTO- skok do procedury obsługi przerwania gdy wystąpi przepełnienie
OF

Rozkazy sterujące

STC- ustaw bit przeniesienia

CLC- zeruj bit przeniesienia

CMC- neguj bit przeniesienia

STD- ustaw bit kierunku transferu danych w operacjach blokowych

CLD- kasuj bit kierunku transferu danych w operacjach blokowych

STI- ustaw bit maski przerwań (odmaskuj przerwania)

CLI- kasuj bit maski przerwań (zamaskuj przerwania)

Rozkazy synchronizujące

WAIT- oczekiwanie na aktywny stan na linii **TEST**

HALT- oczekiwanie na przerwanie

ESC- deklaracja wyłączenia operacji przez procesor przełączenie na procesor zewnętrzny np. koprocesor

LOCK- brak dostępu do magistrali

NOP- operacja pusta

Pętle programowe

Badają stan licznika CX

LOOP- powtórz sekwencję rozkazów jeśli $CX \neq 0$

LOOPE/LOOPZ- powtórz sekwencję rozkazów jeśli $CX \neq 0$ lub jeśli występuje równość argumentów

LOOPNE/LOOPNZ- powtórz sekwencję rozkazów jeśli $CX \neq 0$ lub jeśli nie występuje równość argumentów

JCXZ- skok względny jeśli $CX=0$

Praca krokowa programowa mikroprocesora 8086

Włączenie przez ustawienie bitu **T** w rejestrze znaczników **F**

Do rejestru **F** nie można wpisać danej, modyfikacji **F** można dokonać przez operację wymiany przez stos. Realizuje to sekwencja rozkazów

PUSH F

POP AX

Modyfikacja ustawiająca bit **T** np. operacja **OR AX#0100H**

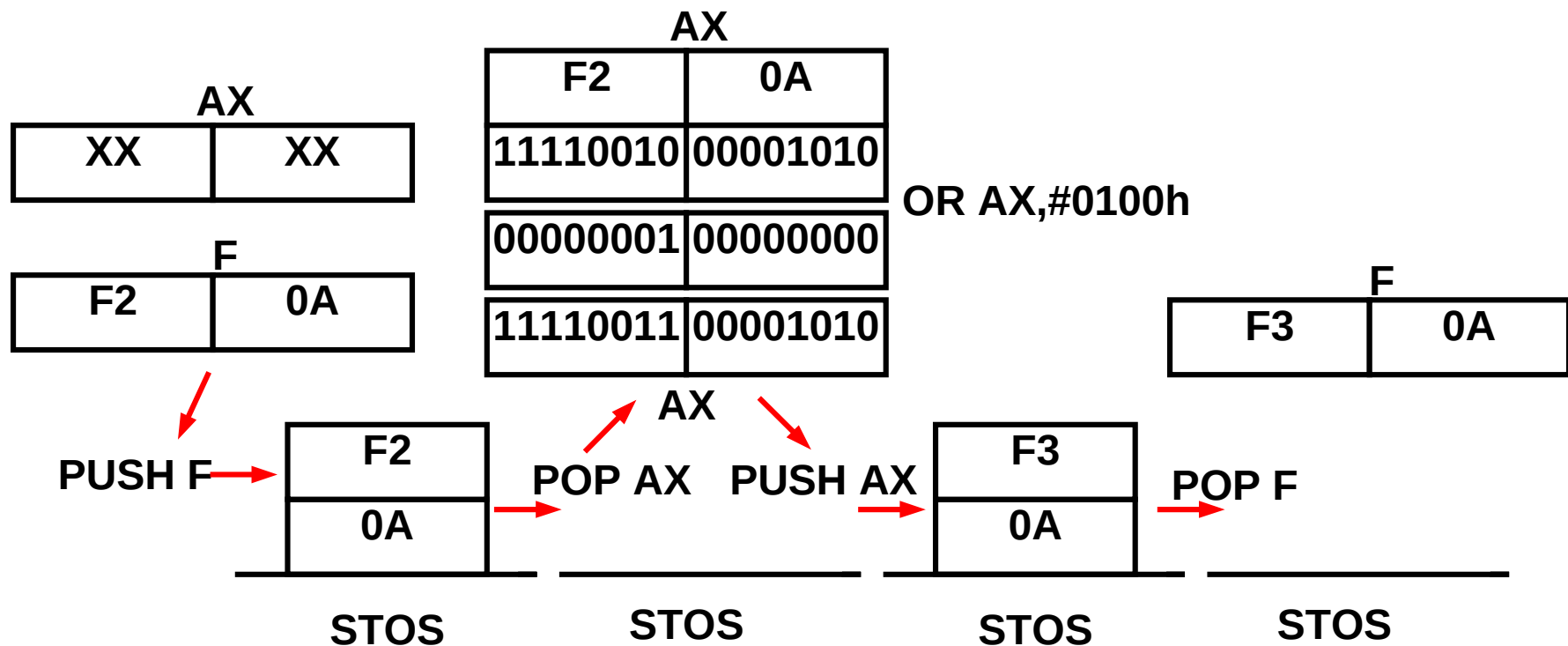
PUSH AX

POP F

Działanie przedstawionej sekwencji rozkazów ilustruje poniższy rysunek

1	1	1	1	0	0	1	0	0	0	0	0	1	0	1	0
X	X	X	X	OF	DF	IF	TF	SF	ZF	0	AF	1	PF	1	CF

Stan rejestru F sprzed modyfikacji: F2 0A



Obsługa przerwania

W trakcie obsługi przerwania mikroprocesor składa na stosie 6-bajtów:

2 bajty rejestru znaczników F

2 bajty rejestru segmentowego CS

Przerwa- następuje pobranie pierwszego bajtu kodu rozkazu z procedury obsługi przerwania

2 bajty licznika rozkazów IP

Na czas obsługi przerwania pracy krokowej, przerwania maskowalne są zamaskowane. Maska przerw jest odtwarzana dopiero po powrocie z obsługi.

Po powrocie z procedury obsługi przerwania jest wykonywany jeden rozkaz i następuje ponowne przejście do procedury obsługi pracy krokowej.