

Miłosz LIZINIEWICZ

## PRZERWANIA W SYSTEMACH MIKROPROCESOROWYCH. PROGRAMOWALNY KONTROLER PRZERWAŃ 8259A I JEGO TRYBY PRACY

**Streszczenie.** W pracy przedstawiono zagadnienia związane z przerwaniami w systemach mikroprocesorowych. Ponieważ we współczesnych komputerach systemy przerwania opierają się na budowie programowalnego kontrolera przerwania 8259A (lub kaskadowego połączenia kilku takich kontrolerów), dlatego w opracowaniu omówiono: budowę, działanie i programowanie kontrolera 8259A. Badania, jakie przeprowadzono na testowanym kontrolerze 8259A, miały udowodnić, że inne tryby pracy niż standardowy tryb pełnego zagnieżdżenia (stosowany w większości komputerów klasy PC) w pewnych sytuacjach mogą okazać się bardziej efektywne.

## INTERRUPTS IN MICROPROCESSOR SYSTEMS. PROGRAMMABLE INTERRUPT CONTROLLER 8259A AND ITS MODES

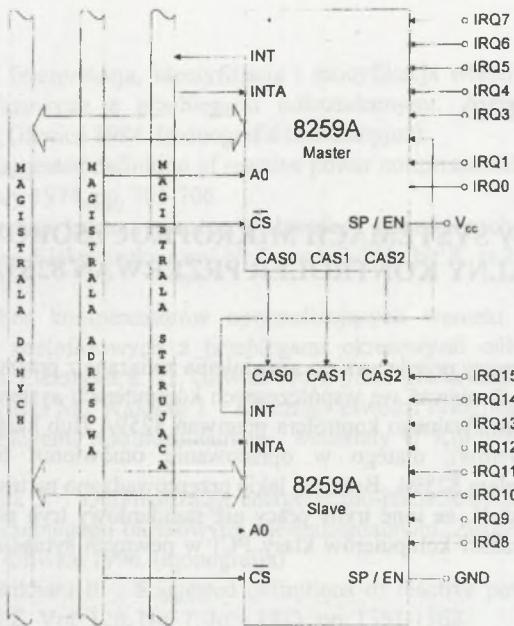
**Summary.** This paper deals with various aspects of interrupts in microprocessor systems. The main focus of this paper is construction, operation and programming of the Programmable Interrupt Controller 8259A (or a cascaded system of several such controllers) which is the main device responsible for systems of interrupts in the present personal computers.

The research, carried out on the 8259A controller, proved that the modes of operations other than the standard fully nested mode, which is used in most present personal computers, may be more efficient in some situations.

### 1. WSTĘP

We wszystkich systemach mikroprocesorowych bardzo istotną rolę pełnią przerwania. Umożliwiają one korzystanie z urządzeń zewnętrznych takich, jak: klawiatura, mysz, stacje dysków, drukarka. Właściwe wykorzystanie przerwania może wydatnie przyspieszyć działanie systemu. Wreszcie wywołanie przerwania w sytuacjach krytycznych często pozwala uniknąć poważnych skutków awarii.

Nad całością systemu przerwania nadzór sprawuje kontroler przerwania. W połowie lat 70. firma Intel wyprodukowała programowalny kontroler przerwania 8259, a w niedługim czasie po tym układ 8259A, będący unowocześnioną i rozbudowaną wersją układu 8259 [5].



Rys. 1. Schemat połączenia kaskadowego dwóch układów 8259A stosowany jako układ przerw w komputerach klasy PC [2]

Fig. 1. Cascaded system of two Programmable Interrupt Controllers 8259A which is applied in PC's [2]

Od tego czasu systemy przerw w większości systemów mikroprocesorowych oparte są na układach 8259A. We współczesnych komputerach klasy PC całość systemu przerw stanowią dwa układy 8259A połączone kaskadowo [2].

W komputerach kontroler przerw 8259A pracuje w trybie podstawowym - pełnego zagnieżdżenia. Jednak programowalny kontroler przerw (ang. *Programmable Interrupt Controller*) posiada kilka trybów pracy.

Celem poniższych rozważań jest przybliżenie budowy, działania i programowania kontrolera, omówienie najważniejszych trybów pracy oraz wyjaśnienie na przykładach, różnic w działaniu kontrolera pracującego w różnych trybach.

## 2. PROGRAMOWANIE 8259A

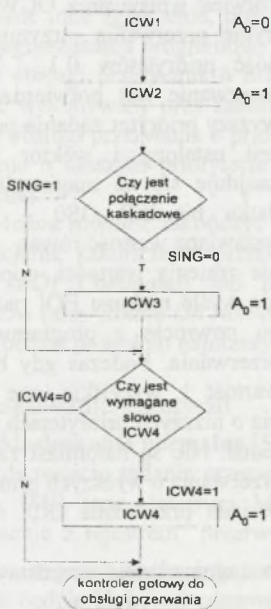
Programowalny kontroler przerw [1,2,5] (ang. *PIC - Programmable Interrupt Controller*) ma funkcję ogólnego menedżera w systemie przerw. Akceptuje on żądania z urządzeń zewnętrznych, określa, które z nadchodzących żądań mają najwyższy priorytet, bada czy nadchodzące przerwanie ma wyższy priorytet niż priorytet przerwania aktualnie obsługiwanego. Każde przerwanie odpowiadające określonej urządzeniu zewnętrznemu posiada swój własny program obsługi, zawierający specjalne funkcje. Programowalny kontroler przerw w wyniku wystąpienia przerwania przesyła do procesora informację o rodzaju przerwania, punkcie programu, w którym nastąpiło przerwanie z konkretnym urządzeniem zewnętrznym. Tę informację PIC dostarcza do procesora wraz z 3-bajtowym rozkazem

CALL [3,5]. W ten sposób wygląda współpraca kontrolera 8259A z mikroprocesorami 8080/85. W przypadku współpracy z procesorami klasy 8086 kontroler generuje wektor przerwania w postaci ośmiobitowej liczby, oznaczającej numer przerwania. Dla 8 przerwań jedynie 3 bity adresu skokowego lub numeru przerwania są zmieniane. Pozostałe bity mogą być ustawiane dowolnie metodą programową. Pozwala to na umieszczenie programów obsługi przerwań na dowolnej stronie przestrzeni adresowej. Podczas współpracy z mikroprocesorami 8086 kontroler generuje wektory przerwań w postaci 8 kolejnych numerów, oznaczających dowolną ósemkę pozycji w tablicy adresów skokowych (tablica ta zawiera do 256 pozycji)[1].

Kontroler 8259A wykorzystuje dwa adresy w przestrzeni adresowej WE/WY. Wejście adresowe  $A_0$  jest przylączone do linii  $A_0$  magistrali adresowej. Wykorzystywana jest wtedy para adresów: parzysty i kolejny nieparzysty. Komunikacja programowa z kontrolerem odbywa się za pomocą rozkazów IN i OUT, podobnie jak z każdym innym urządzeniem we/wy. Aktywacja kontrolera odbywa się za pomocą dekodera adresowego i sygnału  $\overline{CS}$ . Do kontrolera przesyłane są słowa sterujące i maska przerwań, a pobierane są zawartości rejestrów IRR, ISR, IMR, a także numer przerwania o najwyższym priorytecie.

Układ 8259A akceptuje dwa typy rozkazów generowanych przez CPU: słowa inicjalizujące i słowa sterujące wykonawcze.

### Słowa inicjalizujące - Initialization Command Words (ICW) [1,5]



Przed rozpoczęciem normalnej operacji (tzn. po włączeniu systemu) każdy kontroler 8259A jest inicjalizowany przez sekwencję 2 do 4 bajtów taktowanych sygnałem WR.

Słowa inicjalizujące ICW informują kontroler 8259A o:

- liczbie istniejących kontrolerów 8259A w systemie i o sposobie ich połączenia;
- adresie początkowym procedury obsługi lub generowanych numerach przerwań;
- odległości pomiędzy kolejnymi procedurami obsługi - 4 lub 8 bajtów;

Wyemitowanie ICW rozpoczyna sekwencję inicjalizacji 8259A. Raz rozpoczęta kolejność inicjalizacji musi być zakończona zanim 8259A będzie mógł obsługiwać żądane przerwania. Odnosi się to do każdego 8259A w systemie wielu kontrolerów 8259A. Procedura inicjalizująca rozpoczyna się każdorazowo, jeśli do kontrolera wysłany zostanie na adres parzysty bajt, w którym bit  $D_4=1$ .

Rys. 2. Inicjalizacji kontrolera przerwań 8259A [5]

Fig. 2. Initialization of Programmable Interrupt Controller 8259A [5]

### Słowa sterujące wykonawcze - Operation Command Words (OCW) [1,5]

Są to rozkazy stosowane do dynamicznej zmiany trybu pracy kontrolera 8259A.



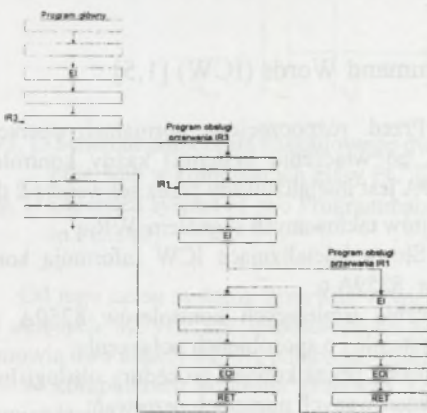
Rozróżniamy następujące tryby pracy kontrolera 8259A:

1. Tryb pełnego zagnieżdżenia - *fully nested mode*
2. Tryb z rotacją priorytetów (dynamiczna zmiana priorytetów)
3. Tryb maskowania
4. Tryb przeszukiwania - *polled mode*
5. Tryb buforowania

OCW mogą być wpisywane w dowolnym momencie po inicjalizacji. Po zaprogramowaniu 8259A przez ICW układ jest już gotów do obsługi przerwania na liniach wejściowych. Ponieważ w czasie wykonywania funkcji określonych przez te słowa zmienia się stan rejestru ISR, tak więc przed ich wysłaniem należy najpierw zablokować obsługę przerwania. Adres nieparzysty zarezerwowany został do obsługi rejestru masek - IMR przez OCW1. Natomiast pozostałe słowa wysyłane są na adresy parzyste. W celu ich rozróżnienia wykorzystywane są bity D<sub>4</sub> i D<sub>3</sub>.

### 3. OMÓWIENIE NIEKTÓRYCH TRYBÓW PRACY KONTROLERA

#### Tryb pełnego zagnieżdżenia (zagnieżdżenia normalnego) [1,5]



EOI - *End of Interrupt* - zakończenie przerwania

Rys. 3. Obsługa przerwania w trybie zagnieżdżenia normalnego

Fig. 3. Interrupt's maintenance in fully nested mode

Po sekwencji inicjalizującej najwyższy priorytet najniższy - IR7.

Obsługę przykładowych przerwania w trybie pełnego zagnieżdżenia przedstawiono na rys. 3.

#### Zakończenie przerwania [5]

Zakończenie przerwania jest niezbędne do prawidłowego funkcjonowania kontrolera przerwania. Odbywa się ono przez wysłanie do kontrolera słowa OCW2, w którym bit EOI ma wartość równą 1. Słowo to powinno kasować bit w rejestrze ISR, odpowiadający za trwające w danej chwili przerwania. Istnieją dwa rozkazy EOI:

Kontroler 8259A będzie pracował w tym trybie po wykonaniu sekwencji inicjalizującej bez jakiegokolwiek wpisywania OCW. W tym trybie przerwanie otrzymują kolejność priorytetów 0,1,... 7. Kiedy przerwanie jest potwierdzone, najwyższy priorytet żądania przerwania jest ustalony i wektor adresowy znajduje się na magistrali. W dodatku bit ISR (IS0 - IS7) ma wstawianą wartość równą 1. Bit ten nie zmienia wartości, dopóki CPU nie wyśle rozkazu EOI natychmiast po powrocie z programu obsługi przerwania. Podczas gdy bit IS ma wartość 1, wszystkie inne przerwania o niższych priorytetach są zabronione. Nie są natomiast zabronione przerwanie o wyższych priorytetach.

posiada przerwanie IR0, natomiast

1. Normalne zakończenie programu obsługi przerwania (ang. *Non Specific EOI Command*) powoduje wyzerowanie aktywnego bitu w ISR. Ponieważ w czasie wykonywania programu obsługi przerwania układ priorytetów nie został zmieniony, więc przez cały czas trwania obsługi przerwania odpowiadający mu bit ma najwyższy priorytet ze wszystkich wzbudzonych w ISR. W celu jego wyzerowania nie trzeba więc przesyłać informacji identyfikującej ten bit. Normalne zakończenie przerwania może być stosowane wtedy, gdy nie były używane słowa ustawiające nowe priorytety i gdy nie był stosowany tryb maskowania specjalnego.

2. Specjalne zakończenie programu obsługi (ang. *Specific EOI Command*) powoduje wyzerowanie wyznaczonego bitu w ISR. W trakcie wykonywania programu obsługi przerwania został zmieniony układ priorytetów przerwania, ponieważ teraz inne zgłoszenie posiada najwyższy priorytet. Kontroler nie posiada informacji o tym, jakie przerwanie jest obecnie obsługiwane. Tak więc słowo OCW2, kończące program obsługi przerwania, musi zawierać informację jednoznacznie określającą przeznaczony do wyzerowania bit ISR. Numer kasowanego bitu jest określony za pomocą bitów L2 – L0 (numer bitu w kodzie binarnym). Specjalne zakończenie programu może być stosowane w każdym trybie.

3. Tryb automatycznego zakończenia programu obsługi (ang. *Automatic EOI Mode*).

Tryb ten jest ustawiany za pomocą ICW4 z bitem AEOI=1. Pozwala on na automatyczne kasowanie bitu aktywnego w rejestrze ISR w czasie ostatniego opadającego zbocza sygnału INTA w sekwencji potwierdzania przerwania. Oznacza to, że przez cały czas trwania programu obsługi bit ISR jest wyzerowany, więc program obsługi mógłby być przerwany przez każde inne zgłoszenie, niekoniecznie o priorytecie wyższym. Narusza to podstawową zasadę zagnieżdżania programów, że tylko przerwanie o priorytecie wyższym może przerwać program obsługi przerwania o niższym priorytecie. Dlatego w tym trybie program obsługi musi być wykonywany przy zablokowanych przerwaniach w mikroprocesorze (rozkaz DI), a program obsługi przerwania o priorytecie wyższym może się rozpocząć, gdy program obsługi przerwania o niższym priorytecie usunie blokadę (rozkaz EI przed powrotem z programu przerwania).

4. Można również zakończyć przerwanie równocześnie z cykliczną zmianą priorytetów, np.: specjalne zakończenie przerwania z cykliczną zmianą priorytetów, (ang. *Rotate On Specific EOI Command*) lub normalne zakończenie przerwania z cykliczną zmianą priorytetów (ang. *Rotate On Non-Specific EOI Command*). Wtedy przerwanie wykonane jako ostatnie będzie posiadało najniższy priorytet (patrz: Tryb rotacji priorytetów).

## Maskowanie przerwania

### A) Maskowanie normalne [5]

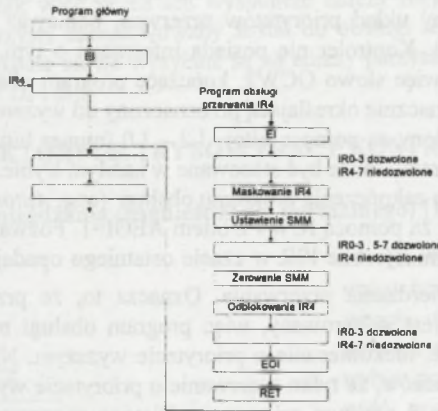
Każde wejście żądania przerwania IR może być indywidualnie maskowane przez rejestr masek - IMR (ang. Interrupt Mask Register) programowany za pomocą OCW1. IMR współpracuje z rejestrem przerwania w obsłudze - ISR (ang. In-Service Register). Należy zauważyć, że jeżeli przerwanie jest aktualnie obsługiwane (syg. INTA), maskowany poziom przerwania będzie blokował przerwanie o niższym priorytecie. Tryb ten nazywany jest trybem maskowania normalnego lub po prostu maskowaniem przerwania.

W celu udostępnienia niższych poziomów można wykonać dwie czynności:

- 1) wpisać rozkaz końca przerwania (End Of Interrupt - EOI) w OCW2 dla wyzerowania bitów ISR lub
- 2) ustawić "Special Mask Mode" używając OCW3.

**B) Maskowanie specjalne [1,2]**

Tryb specjalnego maskowania (ang. *Special Mask Mode*) jest trybem, w którym czasowo blokowane jest działanie bitów w rejestrze ISR. Wprowadzenie tego trybu i wyjście z niego odbywa się za pomocą słowa OCW3. Maskowaniu podlegają te bity rejestru ISR, które odpowiadają zgłoszeniom zamaskowanym w normalny sposób, tzn. za pomocą OCW1. Aby nie dopuścić do obsługi przerwania o niższych priorytetach, należy najpierw zamaskować w zwykły sposób przerwanie aktualnie będące w obsłudze, a następnie wprowadzić tryb specjalnego maskowania.



Rys. 4. Specjalny tryb maskowania

Fig. 4. Special mask mode

**Tryb rotacji priorytetów**

Istnieją dwa rodzaje rotacji priorytetów:

**A) Rotacja automatyczna (ang. *Auto Rotate*) [5]**

1) Rotacja normalna (ang. *Non-Specific Rotation*)

Wykonanie rozkazu *Rotate on Non-Specific EOI* spowoduje wyzerowanie bitu ISR o najwyższym priorytecie i przyznanie wejściu IR najniższego priorytetu. Tak więc w najgorszym razie urządzenie przyłączone do wejścia żądającego obsługi będzie czekało, aż zostaną obsłużone urządzenia z siedmiu innych wejść IR. W poniższym przypadku (patrz rysunek) ISR był ustawiony na linii 4 i 6. Wejście nr 4, posiadające wyższy priorytet, zostało obsłużone i otrzymało najniższy priorytet.

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0	1	0	1	0	0	0	0
7	6	5	4	3	2	1	0

Przed

↑  
najniższy priorytet

↑  
najwyższy priorytet



IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0	Po
0	1	0	0	0	0	0	0	
2	1	0	7	6	5	4	3	

Rotacja uzyskana za pomocą OCW2, gdzie  $R=1$ ,  $EOI=1$ ,  $SEOI=0$ .

#### 2) Rotacja z automatycznym zakończeniem przerwania

Użycie rozkazu *Rotate in Automatic EOI* wprowadza kontroler w tryb bardzo podobny w działaniu do trybu poprzedniego. Różnica polega na tym, że rotacja priorytetów odbywa się automatycznie zaraz po ostatnim impulsie  $\overline{INTA}$ . Wprowadzanie i wyjście z tego trybu odbywa się za pomocą odpowiednich słów OCW2 - *Rotate in Automatic EOI Mode, set* lub *Rotate in Automatic EOI Mode, clear*.

### B) Rotacja specjalna (ang. *Specific Rotate*) [5]

#### 1) Rotacja z ustawianiem priorytetów

Programista może zmienić priorytety przerwania poprzez wyznaczenie przerwania o najniższym priorytecie rozkazem *Set Priority Command*. Zmiana ta odbywa się niezależnie od tego, czy w danej chwili jest obsługiwane przerwanie, tzn. bez wywierania wpływu na ISR. Poprzez wyznaczenie priorytetu najniższego zostanie automatycznie wyznaczone przerwanie o priorytecie najwyższym. Przykładowo: jeżeli ustawimy IR5 jako priorytet najniższy, to automatycznie będzie to oznaczać, że najwyższy priorytet przerwania osiągnie przerwanie IR6.

#### 2) Rotacja ze specjalnym zakończeniem programu obsługi

Wykonanie rozkazu *Rotate on Specific EOI Command* wprowadza kontroler w tryb rotacji priorytetów ze specjalnym EOI. Jest to kombinacja trybu *Set Priority* - wystawienie najniższego priorytetu determinuje priorytet najwyższy; i trybu *Specific EOI* - wyznaczony priorytet jest zerowany w rejestrze ISR. Ten sam efekt można by uzyskać przez wysłanie dwóch słów sterujących: specjalnego zakończenia przerwania (ang. *Specific EOI Command*) i ustalenia najniższego priorytetu przerwania (ang. *Set Priority Command*).

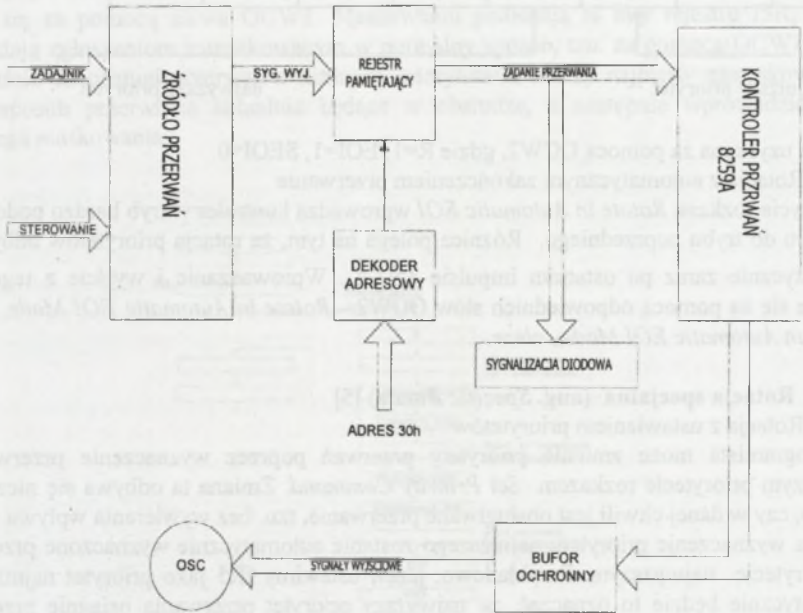
## 4. UKŁAD POMIAROWY

Urządzenie umożliwiające badanie kontrolera przerwania zrealizowano na podstawie schematu blokowego przedstawionego poniżej. Urządzenie zostało zbudowane jako standardowy pakiet systemu MSA-80, na który przeprowadzono testy.

Pakiet sterownika przerwania współpracuje z pakietem jednostki centralnej. W miejsce układu 8259A, znajdującego się na płycie jednostki mikroprocesorowej, dołączony jest wtyk emulacyjny, łączący ten pakiet z układem sterownika przerwania. Układ kontrolera przerwania jest umieszczony na pakiecie sterownika wraz z układami emulacyjnymi generowane przerwania.

Pakiet sterownika przerwania umożliwia obserwację obsługi przerwania na monitorze ekranowym w postaci zaprogramowanych napisów, jak również na wyświetlaczach LED umieszczonych na płycie czołowej panelu sterującego.

Możliwe jest obserwowanie niektórych przebiegów sygnałów sterujących na ekranie oscyloskopu (sygnały: INT,  $\overline{\text{INTA}}$ ,  $\overline{\text{WR}}$ ,  $\overline{\text{RD}}$ , jak również IRQ0 i IRQ1).



Rys. 5. Schemat blokowy modelu sterownika przerwán

Fig. 5. Block diagram showing measurement circuit of interrupt system

## 5. WYNIKI BADAŃ I WNIOSKI

W komputerze osobistym kontroler przerwán pracuje w podstawowym trybie (pełnego zagnieżdżenia). Nie zawsze jednak jest to najlepsze rozwiązanie dla systemów mikroprocesorowych, pracujących przy nadzorowaniu skomplikowanych i kosztownych procesów produkcyjnych.

Aby lepiej przedsięwziąć możliwości kontrolera przerwán zbudowany został układ pozwalający na badanie zachowań kontrolera w zależności od sposobu zaprogramowania i sposobu podawania przerwán. Do testów użyto kontrolera przerwán 8259A, zadajnik przerwán został zrealizowany na mikrokomputerze jednocukładowym AT89C51 firmy Atmel, całość zaś współpracowała z systemem MSA-80 wyprodukowanym przez firmę Unitra. Podczas procedury testowej programowano różne tryby pracy kontrolera i zadawano różne rodzaje przerwán. Wyniki w postaci tekstowej były wyświetlane na ekranie monitora systemu MSA-80.

- Nie zaobserwowano żadnych różnic w działaniu pomiędzy układem czułym na poziom a wyzwalanych zmianą poziomu sygnałów IRQ.
- Badano różne sposoby zakończenia przerwán za pomocą słowa OCW2, podczas gdy zadajnik przerwán działał kolejno w trzech trybach:
  - krążenie w przód (od przerwán 7 do 0),



- krażenie w tył (od przerwania 0 do 7),
- jednoczesne pojawienie się wszystkich sygnałów na liniach IRQ.

Na podstawie wizualnej obserwacji na ekranie nie zaobserwowano różnic pomiędzy tymi trzema trybami zadawania zadań obsługi przerwań. Wyniki zestawiono w poniższej tabeli w formie skróconej

I	II	III
JEDNOCZESNE POJAWIENIE SIĘ WSZYSTKICH IRQ	KRAŻENIE W PRZÓD (7-0)	KRAŻENIE W TYŁ (0-7)
ROZPOCZĘCIE 0	ROZPOCZĘCIE 7	ROZPOCZĘCIE 0
zakończenie 0	ROZPOCZĘCIE 6	zakończenie 0
ROZPOCZĘCIE 1	ROZPOCZĘCIE 5	ROZPOCZĘCIE 1
zakończenie 1	ROZPOCZĘCIE 4	zakończenie 1
ROZPOCZĘCIE 2	ROZPOCZĘCIE 3	ROZPOCZĘCIE 2
zakończenie 2	ROZPOCZĘCIE 2	zakończenie 2
ROZPOCZĘCIE 3	ROZPOCZĘCIE 1	ROZPOCZĘCIE 3
zakończenie 3	ROZPOCZĘCIE 0	zakończenie 3
ROZPOCZĘCIE 4	zakończenie 0	ROZPOCZĘCIE 4
zakończenie 4	zakończenie 1	zakończenie 4
ROZPOCZĘCIE 5	zakończenie 2	ROZPOCZĘCIE 5
zakończenie 5	zakończenie 3	zakończenie 5
ROZPOCZĘCIE 6	zakończenie 4	ROZPOCZĘCIE 6
zakończenie 6	zakończenie 5	zakończenie 6
ROZPOCZĘCIE 7	zakończenie 6	ROZPOCZĘCIE 7
zakończenie 7	zakończenie 7	zakończenie 7

Rodzaje słów OCW2, jakie zastosowano podczas tego badania:

- ⇒ Normalne zakończenie przerwania
- ⇒ Specjalne zakończenie przerwania
- ⇒ Normalne zakończenie przerwania z cykliczną zmianą priorytetów
- ⇒ Specjalne zakończenie przerwania z cykliczną zmianą priorytetów

Ponieważ po inicjalizacji zawsze najwyższy priorytet posiada przerwanie 0, a najniższy przerwanie 7, to przebieg obsługi przerwań dla zadajnika w trybie I i III jest oczywisty. Dla trybu II następowało zagnieżdżanie przerwań, od przerwania 7 (o najniższym priorytecie) do przerwania 0, a następnie zakańczanie przerwań w kolejności zależnej od wysokości priorytetu.

Taki sam przebieg miały przerwania kończone w trybie ze specjalnym zakończeniem przerwania (ang. *Specific EOI*) jak i z normalnym trybem zakończenia przerwań (ang. *Non-Specific EOI*). Wydawać się może, że tryb z rotacją priorytetów powinien zmieniać priorytety przerwań, a co za tym idzie i kolejność obsługiwanych przerwań, jednak zauważymy, że po skończonych ośmiu przerwaniach znajdujemy się powtórnie w miejscu wyjścia i przerwanie 7 ma znów najniższy priorytet, a przerwanie 0 ma priorytet najwyższy.

- Zastosowanie trybu maskowania wymaga wprowadzenia słowa OCW1 na adres parzysty. Słowo to umieszcza się po słowach ICW, tzn. już po zainicjalizowaniu kontrolera przerwań.

Podczas badania zastosowano następujące słowo:

	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
OCW1	1	0	1	0	1	0	1	0	1

gdzie 1 oznacza przerwanie zamaskowane.

Na ekranie monitora zaobserwowano następujące wyniki:

I	II	III
JEDNOCZESNE POJAWIENIE SIĘ WSZYSTKICH IRQ	KRAŻENIE W PRZOD (7-0)	KRAŻENIE W TYŁ (0-7)
ROZPOCZĘCIE 1 zakończenie 1	ROZPOCZĘCIE 7 zakończenie 7	ROZPOCZĘCIE 1 zakończenie 1
ROZPOCZĘCIE 3 zakończenie 3	ROZPOCZĘCIE 5 zakończenie 5	ROZPOCZĘCIE 3 zakończenie 3
ROZPOCZĘCIE 5 zakończenie 5	ROZPOCZĘCIE 3 zakończenie 3	ROZPOCZĘCIE 5 zakończenie 5
ROZPOCZĘCIE 7 zakończenie 7	ROZPOCZĘCIE 1 zakończenie 1	ROZPOCZĘCIE 7 zakończenie 7

I znowu wyniki procedury obsługi przerwania wydają się oczywiste dla I i III trybu pracy zadajnika. W trybie drugim nie obserwujemy już zagnieżdżenia przerwania. Jeżeli jednak weźmiemy pod uwagę, że żądanie przerywania trwa około 1 sekundy, zaś program obsługi przerywania około 2 sekund, czyli 2 razy dłużej, to stanie się oczywiste, że zamaskowanie co drugiego przerywania da czas potrzebny na zakończenie przerywania bez konieczności zagnieżdżenia.

- Testowanie obsługi przerwania przez podawanie zadanej sekwencji. Ponieważ, jak wcześniej wspomniano, układ niezależnie od sposobu uaktywniania się na żądanie obsługi przerywania, tzn. aktywności na poziom lub na zmianę poziomów żądania obsługi przerywania, działa tak samo, w punkcie tym badano jedynie zachowanie układu zależnie od zadanej sekwencji i sposobu zakończenia przerywania.

Rodzaje słów OCW2, jakie zastosowano podczas tego badania:

⇒ Normalne zakończenie przerywania

⇒ Normalne zakończenie przerywania z cykliczną zmianą priorytetów

Wyniki przedstawiono w poniższych tabelach.

I

Zadana sekwencja:

IRQ2

IRQ2

IRQ2 i IRQ3 i IRQ4

I	III
Normalne zakończenie przerywania	Normalne zakończenie przerywania z rotacją priorytetów
ROZPOCZĘCIE 2 zakończenie 2	ROZPOCZĘCIE 2 zakończenie 2
ROZPOCZĘCIE 2 zakończenie 2	ROZPOCZĘCIE 3 zakończenie 3
ROZPOCZĘCIE 3 zakończenie 3	ROZPOCZĘCIE 4 zakończenie 4
ROZPOCZĘCIE 4 zakończenie 4	ROZPOCZĘCIE 2 zakończenie 2

II

Zadana sekwencja:

IRQ2

IRQ2

IRQ2

IRQ2 i IRQ3 i IRQ4

I	III
Normalne zakończenie przerwania	Normalne zakończenie przerwania z rotacją priorytetów
ROZPOCZĘCIE 2 zakończenie 2	ROZPOCZĘCIE 2 zakończenie 2
ROZPOCZĘCIE 2 zakończenie 2	ROZPOCZĘCIE 2 ROZPOCZĘCIE 3
ROZPOCZĘCIE 3 zakończenie 3	zakończenie 3 ROZPOCZĘCIE 4
ROZPOCZĘCIE 4 zakończenie 4	zakończenie 4 zakończenie 2

Normalne zakończenie przerwania dla obu sekwencji generuje na ekranie identyczne rezultaty. Natomiast normalne zakończenie przerwania z rotacją priorytetów różni się dla obu sekwencji.

Różnica polega na fakcie, że przerwanie nr 2 podczas zadania dłuższej sekwencji zagnieżdża się jednokrotnie. W obydwu trybach zadajnika obserwujemy "zgubienie się" pewnej liczby przerw nr 2. Oba fakty pozwalają wysnuć wniosek, że co drugie przerwanie nr 2 zostaje niezauważone przez system.

Dzieje się tak, ponieważ program obsługi trwa dwukrotnie dłużej niż czas pojawienia się zadanego obsługi, dlatego też w przypadku dłuższej sekwencji brakuje czasu na zakończenie trzeciego przerwania nr 2, a ponieważ posiada ono już najniższy priorytet (po pierwszym wywołaniu), następuje jego zagnieżdżenie. Drugie przerwanie nr 2 pozostaje niezauważone w obu trybach zadajnika i dla obu sekwencji.

W rzeczywistych systemach taka sytuacja nie będzie jednak miała wpływu na poprawność działania urządzenia. Rozważmy poprawność pracy urządzenia kontrolującego poziom cieczy w mieszalniku. W przypadku niebezpieczeństwa przepełnienia nastąpi żądanie odcięcia dopływu w cieczy i odczekanie pewnego czasu na ustabilizowanie warunków przebiegu procesu. Mimo że czujnik wciąż będzie sygnalizował zbyt wysoki poziom cieczy, to jednak ciecz nie będzie już dopływać i po pewnym czasie ciecz opadnie do bezpiecznego poziomu.

- Podczas pracy kontrolera z automatycznym zakończeniem przerwania (*Automatic EOI*) dla układu aktywowanego poziomem istotne było odpowiednio wczesne kasowanie żądania obsługi przerwania. W przeciwnym przypadku obsługa przerwania będzie powtarzana aż do zapelnienia całej pamięci przez stos, na który są zapisywane zawartości rejestrów. To odpowiednio szybkie zakończenie przerwania powinno odbyć się jeszcze przed pierwszym, w programie przerwania, rozkazem EI, który odblokowuje system przerwania.

Natomiast podczas pracy w trybie automatycznego zakończenia przerwania dla układu aktywowanego zmianą poziomu nie było już istotne, czy żądanie przerwania zostanie usunięte na początku, czy na końcu podprogramu obsługi. Układ wejściowy jest w tym trybie tylko raz pobudzany sygnałem żądania przerwania.



- W trakcie badania modelu, gdy podawano zbyt duże ilości przerwania, obsługiwane były tylko przerwania o najwyższych priorytetach. Spowodowane to było zbyt długim czasem wykonywania programu ich obsługi.

Aby obsłużyć przerwanie z każdego poziomu priorytetu, należało skrócić czas obsługi pojedynczego przerwania. W układach rzeczywistych nie zawsze jest to możliwe, co może prowadzić do awaryjnych sytuacji. W celu uniknięcia tego należało zastosować rotację priorytetów wykorzystując odpowiednie słowo OCW2. Pozwoli to nawet zwiększyć liczbę obsługiwanych przerwania z każdego poziomu priorytetu.

## 6. UWAGI KOŃCOWE

W opracowaniu przedstawiono możliwości kontrolera przerwania w kontekście jego zastosowania w sprzęcie komputerowym i w przemyśle jako urządzenia pozwalającego zwiększyć bezpieczeństwo procesów przemysłowych. Obecnie wszystkie mikrokomputery jednocukładowe, wykorzystywane w zastosowaniach profesjonalnych, posiadają wbudowane kontrolery przerwania. Mikrokomputery te znajdują szerokie zastosowania głównie w aparaturze przemysłowej, ale również w sprzęcie elektronicznym powszechnego użytku.

## LITERATURA

1. Pogoda Z.: System przerwania. Seria „Pomoce dydaktyczne” - praca w przygotowaniu. Instytut Elektrotechniki Pol. Śl.
2. Metzger P.: Anatomia PC. Wydawnictwo HELION, 1996, wyd. II rozszerzone.
3. Liziniewicz M.: Opracowanie modelu prezentującego różne funkcje sterownika przerwania - praca magisterska, Pol. Śl. Wydz. AEiI, Gliwice 1997.
4. Pogoda Z.: Podstawy techniki mikroprocesorowej, cz.1. Seria „Pomoce dydaktyczne”. Instytut Elektroniki Pol. Śl., Gliwice 1994.
5. Microprocessor & Peripherals Handbook, vol. 1, Intel Corporation 1987.
6. Dokumentacja techniczna systemu MSA-80.

Wpłynęło do Redakcji dnia 8 października 1998 r.

Recenzent: Doc. dr inż. Zdzisław Pogoda

## Abstract

Interrupts, which are supervised by the interrupt controller, constitute a very important part of microprocessor systems. Controllers that are used most often in various microprocessor systems are based on the structure of Intel's Programmable Interrupt Controller 8259A (or a cascaded system of several such controllers).

The Programmable Interrupt Controller (PIC) functions as an overall manager in the Interrupt-Driven system environment. This controller was designed primary to be used in the real time, interrupt-driven microcomputer systems. The 8259A manages eight levels or requests and thanks to its built-in features can be connected to other 8259A units (up to 64 levels).

