Laboratorium Podstaw Informatyki

Laboratorium nr 4 – Przerwania

1 Wstęp

Istnieją czasem takie sytuacje, w których należy przerwać dotychczasową pracę i wykonać coś, co w danej chwili jest ważniejsze. Po skończeniu, można dalej kontynuować dotychczasowe zajęcie – w momencie, w którym zostało ono przerwane.

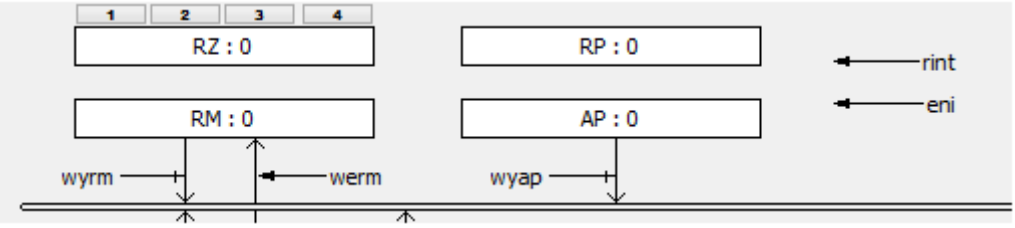
Podobna sytuacja występuje i w świecie komputerów. Wówczas to wykonujący się aktualnie program może zostać przerwany na skutek wystąpienia pewnego zjawiska, nazywanego *zgłoszeniem przerwania*. Zgłaszający przerwanie (może to być człowiek naciskający przycisk, ale też np. inny układ elektroniczny) żąda, aby bieżące zadanie zostało przerwane, i co więcej – jego zgłoszenie zostało obsłużone jak najszybciej.

Fragment programu odpowiedzialny za obsługę przerwania nosi nazwę *procedury obsługi przerwania*. Zazwyczaj, program podczas swojej regularnej pracy „nie dociera" do takiej procedury, a zostaje ona wywołana na skutek wystąpienia przerwania. Ważne jest, aby po skończeniu takiej procedury, powrócić w miejsce, w którym przerwania nastąpiło i być w stanie kontynuować dalszą pracę. Oznacza to konieczność zachowania bieżącego stanu procesora.

Czasami występują sytuacje, w których nie można przerwać aktualnie wykonywanego programu, chociażby podczas konfiguracji układu przerwań. Co więcej, poszczególne przerwania (np. związane z różnymi przyciskami) mogą się różnić od siebie stopniem ważności – priorytetem. Wówczas należy w pierwszej kolejności obsłużyć przerwanie ważniejsze, a dopiero później – te mniej ważne. Do aktywacji i dezaktywacji poszczególnych przerwań służy tzw. *maska*. Przerwanie może zostać *zamaskowane*, oznacza to, że jego zgłoszenie jest jakby niewidoczne dla procesora.

2 Układ przerwań maszyny W

Maszyna W umożliwia obsługę przerwań za pomocą *układu przerwań*. Został on przedstawiony na rysunku poniżej (Rysunek 1).



Rysunek 1. Układ przerwań maszyny W

Składa się on z 4 podstawowych rejestrów:

1. **Rejestru zgłoszeń (RZ)** – przechowującego informacje o zgłoszeniach, pochodzące z przycisków umieszczonych ponad nim (oznaczonych numerami od 1 do 4).
2. **Rejestru maski (RM)** – przechowującego bieżący stan maski.
3. **Rejestru przerwań (RP)** – przechowującego przerwanie aktualnie przyjęte do obsługi.
4. **Rejestru adresu procedury obsługi przerwania (AP)** – przechowującego adres w pamięci, pod którym znajduje się procedura obsługi przerwania.

Trzy pierwsze rejestry (RZ, RM, RP) są 4 bitowe – służą one do przechowywania informacji o przerwaniach, których w maszynie W dostępnych jest właśnie 4. Każdy bit takiego rejestru odpowiada poszczególnemu przerwaniu. Rejestr AP ma rozmiar równy szerokości magistrali adresowej – jego zadaniem jest właśnie przechowywanie adresu.

Z punktu widzenia procesora, bezpośrednio dostępne dla niego są tylko 2 rejestry: RM i AP, bowiem istnieją sygnały sterujące umożliwiające ich odczyt (RM, AP) i zapis (tylko RM). Są to odpowiednio: *wyrm, werm, wyap*. Rejestr AP jest tylko do odczytu, bowiem znajduje się w nim adres procedury obsługi przerwania, który jest tam wpisywany przez układ przerwań, w momencie wystąpienie i przyjęcia zgłoszenia przerwania. Oba te rejestry podłączone są do magistrali adresowej.

Oprócz wspomnianych sygnałów sterujących związanych z wyżej wymienionymi rejestrami, układ przerwań maszyny W udostępnia jeszcze 2:

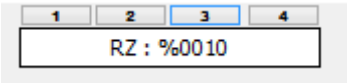
1. **eni** – (ang. *enable interrupt*) – aktywujący układ przerwań i nakazujący mu ustawienie odpowiednich wartości rejestrów: RP i AP,
2. **rint** – (ang. *reset interrupt*) – resetujący układ przerwań – zerujący rejestry: RP i AP oraz odpowiedni bit rejestru RZ[1].

3 Proces przyjmowania przerwania

Proces od naciśnięcia przycisku, a więc od momentu zgłoszenia przerwania, do jego faktycznej obsługi, czyli przejścia programu do procedury obsługi przerwania, jest dość złożony i składa się z kilku faz. Zostały one przedstawione poniżej. Na wszystkich rysunkach, wartości rejestrów 4 bitowych zostały przedstawione w postaci binarnej – znacznie ułatwiającej ich interpretację.

3.1 Zgłoszenie przerwania

Zgłoszenie przerwania w maszynie W następuje na skutek naciśnięcia przycisku (1 – 4). Wówczas to, następuje ustawienie odpowiedniego bitu w rejestrze RZ (Rysunek 2).



Rysunek 2. Zgłoszenie przerwania nr 3

Możliwe jest zgłoszenie kilku przerwań na raz, przykładowo dla przycisków 1, 2 i 4, rejestr zgłoszeń będzie miał postać: *%1101*. Stan tego rejestru odpowiada zgłoszonym przerwaniom, od momentu ostatniego resetu układu przerwań.

3.2 Maskowanie przerwań

Do rejestru maski można wpisywać różne wartości – jego bity odpowiadają poszczególnym przerwaniom. Jeżeli dany bit jest ustawiony (ma wartość 1), to dane przerwanie uznaje się za *zamaskowane*, a więc jest nieaktywne.

3.3 Przyjmowanie przerwania

Układ przerwań dokona przyjęcia przerwania, w momencie uaktywnienia się sygnału *eni*, oczywiście w sytuacji w której jakieś przerwanie zostało zgłoszone. W danej chwili czasu tylko jedno przerwanie może być obsługiwane na raz – odpowiadający mu bit zostaje wówczas ustawiony w rejestrze przerwań (RP). Łatwo można zauważyć, że łączna liczba jedynek w tym rejestrze to maksymalnie 1.

[1] Dokładny opis przyjmowania przerwania znajduje się w następnej części.

Na przyjęcie danego przerwania ma wpływ kilka czynników:

1. zgłoszenie danego przerwania,
2. brak ustawionej maski dla danego przerwania,
3. brak zgłoszenia przerwania o wyższym priorytecie.

O ile w przypadku dwóch pierwszych punktów, komentarz jest zbyteczny, to ostatni punkt wymaga chwili wyjaśnienia. Poszczególne przerwania mają priorytety – są one zgodne z ich numerami. Najwyższy priorytet ma zatem przerwanie nr 1, a najniższy – przerwanie nr 4. Oznacza to, że gdy zgłoszonych zostanie kilka przerwań, to do rejestru przerwań trafi zawsze najwyższe, niezamaskowane przerwanie.

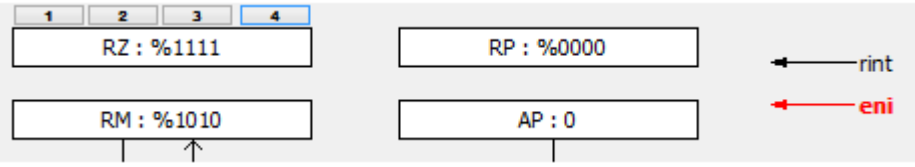
Gdy maszyna W przyjmie przerwanie i przejdzie do wykonywania jego procedury obsługi, układ przerwań może zostać zresetowany. Służy do tego sygnał *rint*. Oznacza to, że można wyzerować rejestr przerwań (RP), ponieważ dane przerwanie zostało już (lub jest aktualnie) obsłużone. Oprócz tego, rejestr adresu procedury obsługi przerwania (AP) również może zostać wyzerowany – zostanie to opisane w następnej części. Skoro dane przerwanie zostało już obsłużone, może zostać wycofane i jego zgłoszenie – odpowiedni bit rejestru zgłoszeń (RZ) zostaje wyzerowany. Rejestr maski, będący rejestrem kontrolującym pracę układu przerwań nie jest modyfikowany przez układ przerwań.

3.4 Adres procedury obsługi przerwania

Gdy dane przerwanie zostanie przyjęte, w rejestrze AP pojawia się (zostaje wpisany przez układ przerwań) adres procedury obsługi przerwania. Adres ten może zostać następnie pobrany przez procesor, celem przeskoczenie do komórki pamięci wskazywanej przez niego. Adresy poszczególnych przerwań można ustawić w ustawieniach symulatora maszyny W – nie jest to dostępne programowo – nie można tego zrobić żadnym rozkazem, ani sygnałem sterującym.

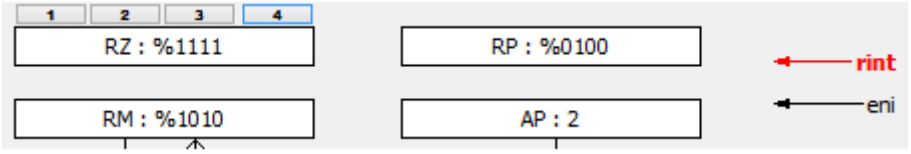
3.5 Przykład

W celu lepszego zobrazowania sposobu, w który działa układ przerwań, zostanie wykorzystany przykład przedstawiony poniżej.



Rysunek 3. Zgłoszenie przerwań 1, 2, 3 i 4

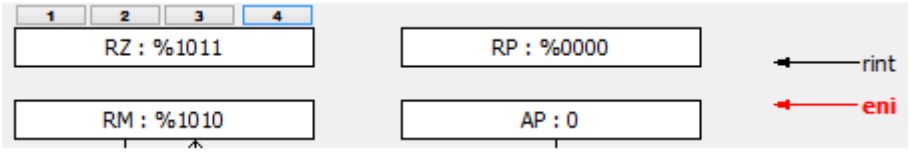
Na rysunku powyżej (Rysunek 3) zostały zgłoszone wszystkie przerwania (1, 2, 3 i 4). W związku z tym, w rejestrze zgłoszeń (RZ) pojawiły się same jedynki. Do rejestru maski (RM) zostały wprowadzone dwie jedynki: maskujące przerwanie nr 1 i 3. Został też uaktywniony sygnał *eni* – chcemy przyjąć przerwanie.



Rysunek 4. Przyjęcie przerwania nr 2

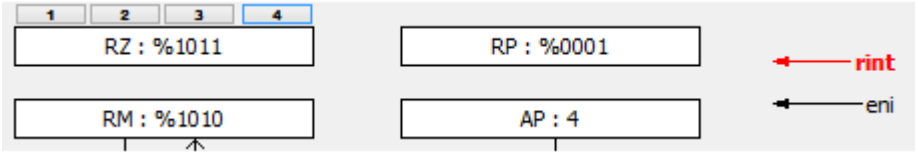
Rysunek powyżej (Rysunek 4) przestawia stan rejestru układu przerwań w następnym takcie. W rejestrze przerwań ustawiony został bit odpowiadający przerwaniu nr 2. Jest to bowiem przerwanie o najwyższym priorytecie, spośród niezamaskowanych przerwań (2 i 4). W rejestrze AP pojawił się też

adres procedury obsługi przerwania nr 2, w tym przypadku jest to adres równy 2. Został również aktywowany sygnał resetujący przerwanie. W normalnej sytuacji sygnał ten pojawia się po pobraniu przez procesor adresu procedury obsługi przerwania.



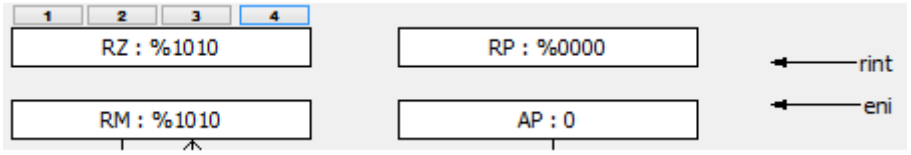
Rysunek 5. Stan rejestrów po resecie

Rysunek 5 przedstawia stan rejestrów układu przerwań po jego resecie. Rejestry RP i AP zostały całkowicie wyzerowane, a w rejestrze zgłoszeń (RZ) nastąpiło wyzerowanie bitu odpowiadającego przerwaniu, które zostało przyjęte do obsługi. Sygnał *eni* został znowu uaktywniony, bowiem chcemy przyjąć kolejne przerwanie.



Rysunek 6. Przyjęcie przerwania nr 4

Rysunek 6 przedstawia przyjęcie kolejnego przerwania. Zgodnie z oczekiwaniami, jest to przerwanie nr 4, bowiem tylko jego zgłoszenie nie zostało zamaskowane. W rejestrze RP bit nr 4 został ustawiony, a w rejestrze AP znajduje się odpowiedni adres. Sygnał *rint* został uaktywniony, celem kolejnego zresetowania układu przerwań.



Rysunek 7. Stan rejestrów po ponownym resecie

Na rysunku powyżej (Rysunek 7) został przedstawiony stan rejestrów po kolejnym resecie. W rejestrze RZ bit nr 4 został wyzerowany. Kolejne uaktywnianie sygnału *eni* w kolejnych taktach nie spowoduje już żadnych zmian w rejestrach – pozostałe przerwania (1 i 3) zostały zamaskowane i są niewidoczne.

4 Rozkazy przerywalne

Do tej pory przerwania były omawiane z punktu widzenia układu przerwań. Warto też przyjrzeć się drugiej stronie tego zagadnienie – co tak naprawdę jest przerywane. Na pewno wiadomo, że jest to program, ale istotne jest zrozumienie, że stwierdzenie wystąpienia przerwania i jego przyjęcie następuje na poziomie rozkazów, a nie na poziomie instrukcji. Oznacza to, że programista piszący program w języku asemblera nie sprawdza ręcznie, czy nastąpiło przerwanie – jest ono sprawdzane podczas wykonywania *rozkazów przerywalnych*. Są to rozkazy realizujące zwykłe operacje (jak np. dodawanie, odejmowanie itp.), które są odpowiednio zaprojektowane, tzn. dostosowane do przyjmowania przerwań.

Większość rozkazów można napisać w wersji przerywalnej. Są jednak takie, które nie powinny dawać takiej możliwości. Przykładowo rozkazy operujące na maskach, a więc służące de facto do kontroli układu przerwań.

Żeby dany rozkaz był rozkazem przerywalnym, powinien on wykorzystywać sygnały sterujące układem przerwań. Są to zarówno sygnały *eni* oraz *rint*, jak i sygnały czytające i piszące do odpowiednich jego rejestrów. Poniżej znajduje się kod rozkazu dodawania, napisanego w wersji przerywalnej.

```
1. ROZKAZ DOD;
2. czyt wys wei il;
3. wyad wea eni
4. JEŻELI INT TO @przerw GDY NIE @norm;
5. @norm czyt wys weja dod weak wyl wea KONIEC;
6. @przerw czyt wys weja dod weak dws;
7. wyls wes wyws wea;
8. pisz wyap wel wea rint;
```

W powyższym przykładzie wyróżnione zostały elementy związane z obsługą przerwań. Pierwszym z nich jest sygnał *eni*. Pojawia się on w 3 linijce, właściwie na początku rozkazu. Powoduje on, że do rejestru przerwań trafia wartość, reprezentujące przyjęte przerwanie, oczywiście w przypadku, w którym jakieś przerwanie zostało zgłoszone.

W linijce 4 znajduje się sprawdzenie, czy jakieś przerwanie zostało przyjęte. Jeżeli tak (czyli jeśli w rejestrze RP znajduje się wartość różna od 0), to fraza nastąpi przejście do etykiety *@przerw*, w przeciwnym przypadku – do etykiety *@norm*.

Gdy żadne przerwanie nie zostało zgłoszone, lub zgłoszone przerwania zostały zamaskowane, uaktywniane są standardowe sygnały sterujące towarzyszące kolejnej fazie rozkazu dodawania (linijka 5).

W sytuacji, gdy zostało przyjęte jakieś przerwanie (etykieta *@przerw*), również uaktywniana jest część sygnałów rozkazu dodawania, ale bez dwóch ostatnich: *wyl wea*. Dzieje się tak, ponieważ następną instrukcją wykonaną po rozkazie dodawania, będzie nie ta, o adresie przechowywanym w liczniku, lecz ta, której adres znajduje się w rejestrze AP.

Warto zwrócić uwagę, że niezależnie od tego, czy przerwanie zostało przyjęte, czy nie, to operacja wykonywana przez ten rozkaz (tutaj jest to dodawanie) zostaje wykonana. Dopiero po jej wykonaniu, może dojść do ewentualnych czynności związanych z samym przerwaniem.

W celu zachowania stanu procesora (tj. licznika rozkazów) następuje jego zapisanie na stosie. Odbywa się to w 3 linijkach: 6, 7 i 8, a konkretniej odpowiadają za to następujące sygnały:

- *dws* – dekrementacja wskaźnika stosu,
- *wyls wes wyws wea* – przesłanie zawartości licznika do rejestru S i wprowadzenie do rejestru A zawartości wskaźnika stosu,
- *pisz* – zapisanie licznika na szczycie stosu.

Gdy stan licznika jest zachowany, następuje pobranie adresu procedury obsługi przerwania (z rejestru AP) i wprowadzenie go do licznika i rejestru A: *wyap wel wea*. Dzięki temu, następnym wykonywanym rozkazem będzie pierwszy rozkaz procedury obsługi przerwania.

Na samym końcu aktywowany jest sygnał *rint*, który powoduje zresetowanie układu przerwań tak, by następny przerywalny rozkaz mógł działać poprawnie.

5 Obsługa przerwań w programach

Adresy poszczególnych przerwań w maszynie W zostały domyślnie ustawione na wartości odpowiadające numerom tychże przerwań, czyli 1, 2, 3 i 4. Oznacza to, że w momencie wystąpienie i przyjęcia danego przerwania, rozpocznie się wykonywanie programu od któregoś z tych adresów.

5.1 Szkielet programu

Jak można łatwo zauważyć, adresy przerwań są kolejnymi wartościami, przez co miejsce na jakiekolwiek instrukcje ich procedur obsługi właściwie ogranicza się do pojedynczej instrukcji. Ze względów oczywistych, jeżeli procedura obsługi ma zrobić coś więcej niż tylko przerwać działanie programu (instrukcja STP), to najsensowniejszym rozwiązaniem jest napisanie jej w innym miejscu, oznaczonym stosowną etykietą i zastosowaniu instrukcji skoku bezwarunkowego pod danym adresem zarezerwowanym na przerwanie. Podobna sytuacja ma miejsce w przypadku programu głównego, który zaczyna się od adresu 0. Przedstawia to poniższy przykład.

```
        SOB PROGRAM
        SOB PRZERW1
        SOB PRZERW2
        SOB PRZERW3
        SOB PRZERW4
PROGRAM: …
        …
        …
        STP
PRZERW1: …
        …
        …
        PWR
PRZERW2: …
        …
        …
        PWR
PRZERW3: …
        …
        …
        PWR
PRZERW4: …
        …
        …
        PWR
```

Pięć pierwszych linijek programu składa się wyłącznie z samych rozkazów skoków bezwarunkowych. Ponieważ program zaczyna się wykonywać od adresu 0, to następuje skok do etykiety PROGRAM. W momencie rozpoczęcia wykonywania procedury obsługi przerwania, następuje przejście do odpowiedniego adresu (1 – 4), a następnie skok do etykiety związanej z danym przerwaniem (tutaj PRZERWn).

5.2 Procedura obsługi przerwania

Na listingu przedstawiającym szkielet programu można zauważyć, że każda procedura kończy się instrukcją PWR. Służy ona do *powrotu* z tejże procedury, podobnie jak to ma miejsce w przypadku

podprogramów. Jej zadaniem jest pobranie ze stosu zapisanego tam stanu licznika i przywrócenie go w odpowiednie miejsce.

Istnieje jeszcze jeden rejestr, którego stan winien być zachowany podczas obsługi przerwań. Jest nim akumulator – przechowuje on aktualne wyniki działań i mógłby być nadpisany innymi wartościami podczas wykonywania procedury obsługi przerwania. W związku z tym, należy zadbać (już w samej procedurze) o jego zachowanie – przykładowo na stosie.

Kolejnym aspektem, na który należy zwrócić uwagę jest możliwość przerwania aktualnie wykonywanej procedury obsługi innego przerwania. Taka sytuacja jest czasami dopuszczalna, ale w przypadku tego laboratorium ograniczymy się do trybu *bez wywłaszczania* czyli takiego, w którym procedura obsługi przerwania jest nieprzerywalna.

Do tego celu można wykorzystać rejestr maski. Z rejestrem tym związane są 3 dodatkowe rozkazy:

1. **CZM** – pobiera zawartość rejestru maski i zapisuje ją pod adres będący jego argumentem.
2. **MSK** – zapisuje do rejestru maski wartość odczytaną spod adresu będącego jego argumentem.
3. **MAS** – zapisuje do rejestru maski wartość będącą jego argumentem (adresowanie natychmiastowe).

Poniżej znajduje się przykładowy szkielet procedury obsługi przerwania nr 1.

```
PRZERW1:  CZM MASKA   // zapis aktualnej maski
          MAS 15      // zablokowanie wszystkich przerwań
          DNS         // zachowanie na stosie akumulatora
          …           // operacje
          PZS         // pobranie ze stosu akumulatora
          MSK MASKA   // przywrócenie starej maski
          PWR         // powrót z procedury obsługi przerwania
```

Wpisanie wartości 15 do rejestru maski powoduje zamaskowanie wszystkich przerwań (binarnie %1111). Bieżąca wartość maski jest zapisywana w zmiennej (pod adresem wskazywanym etykietą MASKA). Zawartość akumulatora zapisywana jest na stosie.

6 Zadania do wykonania

1. Napisać program wypisujący na ekran pojedynczy znak w pętli. Zgłoszenie przerwania powinno poskutkować wyświetleniem jego numeru na ekranie.
2. Wykorzystując program z poprzedniego laboratorium (rysowanie kształtów), uzupełnić go o obsługę przerwań, polegającą na:
    a) wyświetlaniu numeru przerwania w momencie jego pojawiania się,
    b) zliczaniu ilości wystąpień każdego z przerwań i wypisaniu ich na koniec pracy programu,
    c) zakończenia pracy programu na skutek wyświetlenia całej figury lub przekroczenia przez któryś z liczników wartości będącej dwukrotnością numeru danego przerwania (2, 4, 6, 8).