

Przerwania w systemie mikroprocesorowym

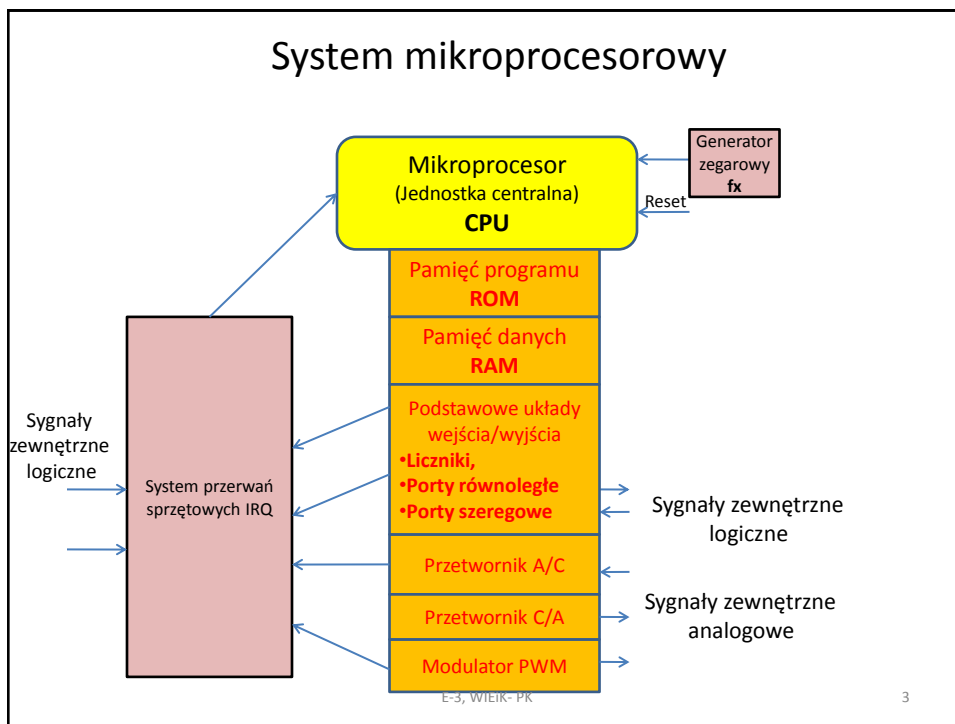
E-3, WIEIK- PK

1

Obsługa urządzeń wejścia/wyjścia

- W każdym systemie mikroprocesorowym oprócz pamięci programu i pamięci danych znajduje się szereg układów lub urządzeń wejścia/wyjścia, które są niezbędne do komunikacji systemu np. z użytkownikiem (przyciski, klawiatura, myszka, monitor) lub do komunikacji z innym systemem mikroprocesorowym, np. porty szeregowo lub równoległe do transmisji danych.
- Oprócz tego każdy system zawiera podstawowe układy wejścia/wyjścia, typu układy czasowo-licznikowe, porty do transmisji szeregowej.
- Każde z tych układów wymaga „obsługi”, na co się składa: wstępne zaprogramowanie, np. ustawienie parametrów i trybu pracy, jest to przeważnie realizowane jednokrotnie na początku programu.
- Późniejsza obsługa wymaga już wielokrotnego i cyklicznego wykonywania pewnych operacji (fragmentu programu) przez jednostkę centralną aby efekt działania układu wejścia/wyjścia nie był stracony lub by nie nastąpiło nieprawidłowe działanie programu w systemie, a tym samym nieprawidłowe działanie urządzenia.
- Każdy układ wejścia/wyjścia zawiera przynajmniej jeden bit (tzw. flagę) informującą system o potrzebie przeprowadzenia obsługi.

2



Obsługa sytuacji wyjątkowych

- Oprócz tego system mikroprocesorowy musi czasami reagować na sygnały zewnętrzne (zdarzenia zewnętrzne) w ściśle określonym czasie. Sygnały te mogą występować w sposób cykliczny lub mogą mieć charakter przypadkowy (losowy).
- Od czasu do czasu mogą się zdarzyć sytuacje wyjątkowe w systemie mikroprocesorowym, np. pojawienie się sygnału zerowania (Reset) od jakiegoś układu lub elementu w systemie, może się zdarzyć błąd programowy polegający na dzieleniu przez zero lub próba wykonania skoku do adresu w pamięci programu lub w pamięci danych, którego nie ma w systemie, może wystąpić błąd parzystości przy zapisie/odczytaniu pamięci RAM.
- Na takie zdarzenia dobrze zaprogramowany i zaprojektowany system mikroprocesorowy, powinien tak zareagować aby nie nastąpiło np. zawieszenie systemu i tym samym zaprzestanie działania jakiegoś urządzenia lub nieprzewidywalne działanie urządzenia.

Metoda odpytywania

- Najprostszym sposobem obsługi poszczególnych układów wejścia/wyjścia jest ciągłe sprawdzanie flag (bitów) w kolejnych układach czy nie wymagają obsługi (tzw. metoda odpytywania, z ang. polling). Jest to proste rozwiązanie ale zabiera dużo czasu jednostce centralnej i tracone jest dużo czasu na oczekiwanie że flaga zostanie ustawiona przez układ wejścia/wyjścia. Jest bardzo nieefektywna metoda obsługi układów lub urządzeń wejścia/wyjścia.
- W tej metodzie występuje np. niebezpieczeństwo spóźnienia odczytania już odebranego przez port szeregowy znaku. W takim przypadku nie odczytany na czas znak i uległ zapisaniu nowo odebrany znakiem. Aby metoda ta działała poprawnie, program odbierający znaki musi wystarczająco często sprawdzać, czy na wejściu pojawił się nowy znak - jest to strata czasu przez CPU.

Metoda przerwania

- Lepszym rozwiązaniem jest tak skonstruowanie (zaprojektowanie) systemu mikroprocesorowego i jego oprogramowanie aby poszczególne układy zgłaszały potrzebę obsługi do jednostki centralnej tylko wtedy kiedy będzie taka potrzeba.
- Wtedy jednostka centralna przerywa aktualnie wykonywany fragment programu i skacze do innego miejsca w pamięci programu i wykonuje fragment programu, który „obsługuje” układ wejścia/wyjścia, który zgłosił „przerwanie”. Po zakończeniu obsługi, CPU bezpiecznie wraca do miejsca w pamięci programu, w którym nastąpiło „przerwanie”.

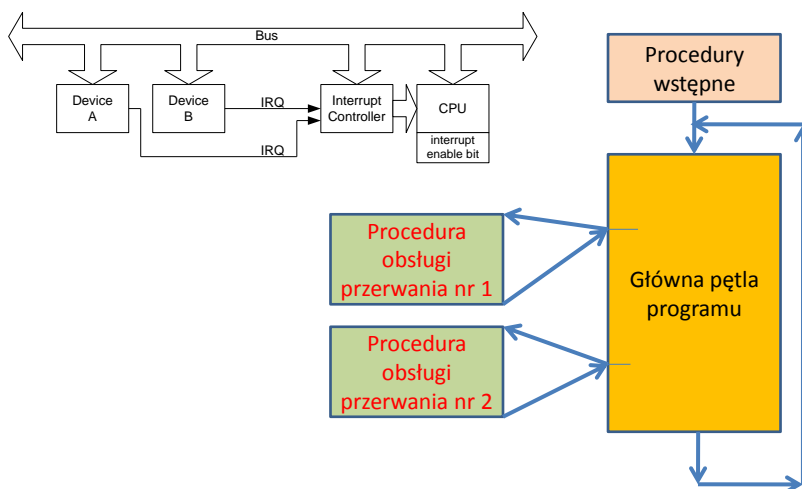
Metoda przerwań

- Ten sposób obsługi urządzeń lub układów wejścia/wyjścia nazywamy systemem przerwań z ang. *Interrupts System*.
- Najczęściej w ten sposób obsługuje się układy wejścia/wyjścia i wtedy mówimy o przerwaniach sprzętowych.
- Program do obsługi „przerwania” związany jest z odczytem/zapisaniem rejestrów układu wejścia/wyjścia, wykonaniem pewnych obliczeń i odczytem/zapisem danych do pamięci RAM lub odczytem/zapisem innych układów wejścia/wyjścia.

E-3, WIEIK- PK

7

System przerwań



E-3, WIEIK- PK

8

Rodzaje przerwań

Przerwania można podzielić na kilka kategorii:

- sytuacje wyjątkowe (ang. exception interrupts), tzw. **przerwania niemaskowalne (ang. non-maskable interrupt, NMI)**.
- **przerwania maskowalne sprzętowe (ang. event interrupts)**,
- **przerwania programowe (ang. software interrupts)**,
- **pułapki (ang. traps)**.

Przerwania niemaskowalne

- **Przerwania niemaskowalne** – tzn. użytkownik (programista) nie może je zablokować ani sprzętowo ani programowo. Programista musi uwzględnić programy do obsługi tych przerwań w oprogramowaniu całego systemu. Są to przerwania najszybciej obsługiwane o najwyższym priorytecie.
- Do przerwań niemaskowalnych zaliczymy następujące zdarzenia:
 - sygnały z CPU – dzielenie przez zero, błąd parzystości, przepełnienie (ang. overflow), itp., błąd działania CPU,
 - sygnały RESETU pochodzące z układu zerowania (zewnętrznego lub wewnętrznego), układu monitorowania napięcia zasilania, układu tzw. watchdoga, układu monitorowania generatora zegarowego,

Przerwania maskowalne

- **Przerwania maskowalne** programista może odblokować lub zablokować w sposób programowy, poprzez ustawienie lub zerowanie odpowiednich bitów w rejestrach do ustawiania systemu przerwania.
- W mikrokontrolerach system przerwania sprzętowych uwzględnia sygnały przerwania od układów wewnętrznych a także sygnały zewnętrzne pochodzące od urządzeń podłączonych do mikrokontrolera.
- Przerwania maskowalne sprzętowe są zgłaszane przez układy wewnętrzne mikrokontrolera systemu, np.:
 - układy czasowo-licznikowe (Timers, Counters),
 - porty szeregowo (UART, USART, SPI, I2C, CAN..),
 - przetwornik A/C,
- Przerwania maskowalne sprzętowe są też zgłaszane do systemu mikroprocesorowego przez urządzenia lub układy zewnętrzne, np.:
 - klawiaturę,
 - napędy dysków,
 - karty sieciowe,
 - karty dźwiękowe
- Reakcja na sygnały zewnętrzne (przerwania zewnętrzne) może być ustawiona:
 - reakcja na zbocze opadające, lub zbocze narastające,
 - reakcja na zmianę stanu logicznego, na stan logiczny „0” lub „1”

Przerwania programowe

- **Przerwania programowe** są wywoływane przez program (świadome działanie programisty).
- W kodzie programu umieszcza się rozkaz który wywołuje procedurę do obsługi przerwania.
- Przerwania programowe najczęściej wykorzystywane są do komunikacji z systemem operacyjnym, np. w komputerze klasy PC, w procedurze obsługi przerwania (np. w DOS 21h, 2fh, Windows 2fh, Linux x86 przerwanie 80h) umieszcza się kod wywołujący odpowiednie funkcje systemowe w zależności od zawartości rejestrów ustawionych przez program wywołujący.
 - Np. w systemie DOS przerwanie int 21h z parametrem ah=01h
 - Czytanie znaku z echem z standardowego wejścia (klawiatura i ekran monitora). Funkcja oczekuje na znak w standardowym strumieniu wejściowym. Po odebraniu znaku funkcja w rejestrze AL zwraca kod tego znaku.

Pułapki

- **Traps (pułapki)** – sytuacja, która nie jest błędem, jej wystąpienie ma na celu wykonanie określonego kodu. Wykorzystywane przede wszystkim w debuggerach - testowanie oprogramowania, śledzenie wykonywanego programu, krokowe wykonywanie programu.
- Procedura do obsługi pułapki polega najczęściej na wyprowadzeniu zawartości wybranych rejestrów procesora lub zawartości fragmentu pamięci RAM na zewnątrz systemu np. poprzez port szeregowy do komputera typu PC na którym można podglądać zmiany zachodzące w rejestrach i pamięci RAM. Dzięki temu można znaleźć błędy działania w programie.
- Po zakończeniu obsługi pułapki procesor powraca do wykonywania przerwanej kodu.

Priorytet przerwania

Priorytet przerwania.

- Każde przerwanie ma określony priorytet (ważność). Jeśli zostanie zgłoszonych jednocześnie kilka przerwania to jako pierwsze zostanie obsłużone przerwanie o najwyższym priorytecie. Po zakończeniu procedury obsługi przerwania obsługiwane jest następnie przerwanie o najwyższym priorytecie.
- W większości przypadków przerwania niemaskowalne mają najwyższy priorytet.
- Priorytet poszczególnych przerwania na ogół jest już ustalony przez projektanta systemu mikroprocesorowego (producenta mikroprocesora lub mikrokontrolera) ale może być modyfikowany w pewnym zakresie przez programistę.
- Np. dla mikrokontrolera 8051 priorytet przerwania maskowalnych:
 1. **Najwyższy dla przerwania zewnętrznego INT0,**
 2. przerwanie z licznika T0,
 3. przerwanie zewnętrznego INT1,
 4. przerwanie z licznika T0,
 5. Najniższy dla przerwania z portu szeregowego,

Wektor przerwań

Wektor przerwań.

- Kiedy wywoływane jest przerwanie, do licznika rozkazów w CPU ładowany jest adres procedury obsługi przerwania (adres pierwszego rozkazu programu do obsługi przerwania).
- Adres ten pobierany jest z tabeli umieszczonej w pamięci programu zwanej wektorem przerwań. Wektor przerwań zawiera adresy procedur obsługi przerwań oraz adres procedury wywoływanej po resecie mikrokontrolera.
- Wektor przerwań (początkowe adresy programów do obsługi przerwań) może być ustalony na stałe przez producenta mikroprocesora/mikrokontrolera lub w niektórych przypadkach programista może narzucić własne adresy procedur do obsługi przerwania.
- W większości przypadków i tak pod pierwszym adresem procedury do obsługi przerwania umieszcza się rozkaz skoku do innego miejsca w pamięci programu.

Wektor przerwań

W zależności od budowy i stopnia skomplikowania systemu mikroprocesorowego po pojawieniu się sygnału przerwania jednostka centralna:

- dokładnie wie z którego urządzenia pojawił się sygnał żądania obsługi przerwania i skacze do pierwszego adresu w pamięci programu obsługującego przerwanie,
- CPU rozpoznaje że jest zgłoszone przerwanie ale nie wie z jakiego miejsca (od jakiego urządzenia). Wtedy program (świadome działanie programisty) musi rozpoznać skąd pochodzi zgłoszenie przerwania i skoczyć do procedury obsługi przerwania.
- Np. dla mikrokontrolera 8051 wektor przerwań:
 - 0003H - dla przerwania zewnętrznego INTO,
 - 000BH - dla przerwania z licznika T0,
 - 0013H - dla przerwania zewnętrznego INT1,
 - 001BH - dla przerwania z licznika T0,
 - 0023H - dla przerwania z portu szeregowego,

Mechanizm obsługi przerwania

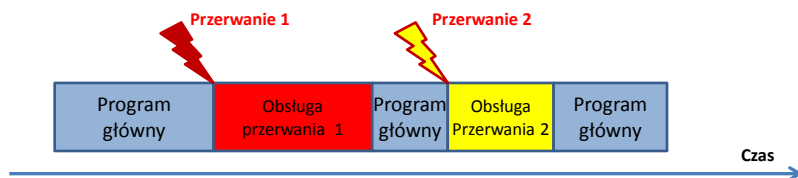
- Pierwszą czynnością wykonywaną po wystąpieniu przerwania jest zachowanie obecnego stanu programu, tak by było możliwe jego odtworzenie po zakończeniu obsługi przerwania i bezpieczny powrót do przerwanego programu.
- Dokonywane jest to poprzez automatyczne zachowanie na stosie: licznika rozkazów, akumulatora, odpowiednich rejestrów oraz rejestru stanu (rejestru kontrolnego).
- Niektóre rejestry i fragmenty pamięci RAM muszą być odłożone na stos samodzielnie przez programistę (lub zapewnia to kompilator np. języka C).
- Po zachowaniu stanu procesora procedura obsługi przerwania może korzystać z wyżej wymienionych rejestrów bez niebezpieczeństwa zakłócenia pracy programu głównego.
- Aby zapobiec przerywaniu procedury obsługi przerwania przez inne przerwanie można zablokować inne przerwanie. Czynności wykonywane przez procedurę obsługi przerwania zależą od źródła przerwania. W większości przerwania od urządzeń peryferyjnych konieczne jest wyzerowanie znacznika przerwania. Zerowanie tego znacznika najczęściej wykonywane jest poprzez odczyt odpowiedniego rejestru statusowego, a następnie zapis lub odczyt rejestru stowarzyszonego z danym urządzeniem peryferyjnym. Na przykład skasowanie flagi przerwania od portu szeregowego wykonywane jest przez odczyt rejestru danych.

E-3, WIEIK- PK

17

Mechanizm obsługi przerwania

- Czasami jedna procedura obsługi przerwania wywoływana jest przez kilka źródeł (np. przerwanie układu czasowego może być wywołane przez 5 różnych źródeł). W takim przypadku na początku procedury obsługi należy wykryć, które źródło wywołało przerwanie. Dokonujemy tego przez testowanie odpowiednich bitów w rejestrach statusowych urządzeń peryferyjnych.
- Na końcu procedury obsługi przerwania programista musi odtworzyć stan wybranych rejestrów i fragmentu pamięci RAM.
- Ostatnią instrukcją procedury obsługi jest specjalna instrukcja powrotu z przerwania. Instrukcja ta powoduje przywrócenie stanu z przed wywołania przerwania. To znaczy, zdejmuje ze stosu akumulator, wybrane rejestry i rejestr stanu oraz licznik rozkazów. Zerowany jest bit maski przerwania.

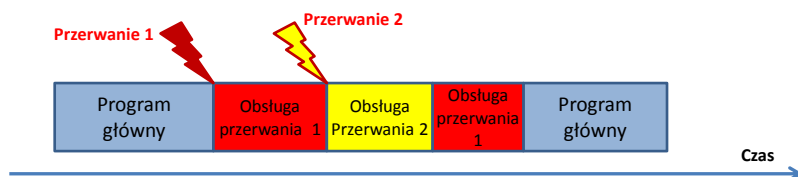


E-3, WIEIK- PK

18

Czas obsługi przerwania

- Obsługa urządzeń wejścia/wyjścia i sytuacji wyjątkowych w systemie mikroprocesorowym jest jednym z najważniejszych zadań dla programisty.
- Oprogramowanie układów mikroprocesorowych przewidzianych do sterowania i kontroli w głównej mierze składa się z programów do obsługi przerwania. Program główny w takim przypadku może być nawet pustą pętlą, która nic nie robi.
- W zależności od budowy i stopnia skomplikowania systemu mikroprocesorowego czas przejścia CPU do obsługi przerwania może być stały i taki sam dla wszystkich przerwania lub w bardziej zaawansowanych systemach można wyróżnić przerwanie które są „wolniej” albo „szybciej” obsługiwane po pojawieniu się sygnału zgłoszenia (np. mikrokontrolery typu ARM).



E-3, WIEIK- PK

19