

Mikroprocesory i mikrokontrolery Laboratorium nr 4	Temat: Przerwania
Grupa: 21b	Michał Lechowicz Mateusz Moneta

Cel ćwiczenia:

Celem ćwiczenia było zapoznanie się z systemem przerwań w procesorze **Intel 8051**. Przerwania implementuje się, pisząc skoki do odpowiednich procedur w odpowiednich miejscach pamięci. Dodatkowo możemy konfigurować system przerwań przy pomocy odpowiednich rejestrów i bitów kontrolnych. Poniższe zadanie miało nas wprowadzić w zastosowanie tego przydatnego podsystemu.

Zadanie na ocenę 5.0:

1. Opis zadania:

W zadaniu należało napisać program, który będzie zliczał sekundy i wyświetlał je w postaci binarnej za pomocą zestawu diod LED portu 2.

2. Kod programu:

<pre> CSEG AT 0 AJMP INIT CSEG AT 0Bh ; przepelnienie Timera 0 AJMP INTERRUPT_OVERFLOW ; idz do instrukcji przerwania CSEG AT 30h INIT: SETB EA ; globalne odblokowanie przerwan SETB ET0 ; odblokowanie przerwan na Timerze 0 MOV TMOD, #00000001b ; Ustaw Timer 1 w tryb 16-bitowy SETB TR0 ; Start Timera 0 LICZ_CZAS: CJNE R0, #14, \$; Czekaj az R0 rowne 14, czyli uplynie 1 sekunda INC R2 ; inkrementuj rejestr R2 sluzacy do wyswietlania diod MOV A, R2 ; przenies wartosc rejestru R2 do A XRL A, #0FFh ; wykonaj alternatywe rozlaczna (funkcja XOR) miedzy A oraz wartoscia FF MOV P2, A ; zapal diody MOV R0, #0 ; Zeruj rejestr R0 SJMP LICZ_CZAS ; uplynieniu sekundy wroc do LICZ_CZAS INTERRUPT_OVERFLOW: ; Po przepelnieniu licznika wykonaj: INC R0 ; inkrementuj R0 - wykonano jeden cykl Timera RETI ; powrot z przerwania END </pre>
Kod programu

3. Obliczenia:

Na początku należy obliczyć co ile przepełnień ma być inkrementowany licznik bitowy. Jako że częstotliwość rezonatora równa jest 11,0592 MHz, to:

$$\frac{11,0592 \text{ MHz}}{12} = 921\,600 \text{ cykli}, \text{ gdzie } 12 - \text{ilość taktów zegara na jeden cykl maszynowy.}$$

Jako że używamy licznika 16-bitowego, to przepełnienie nastąpi co 65 536 cykli.

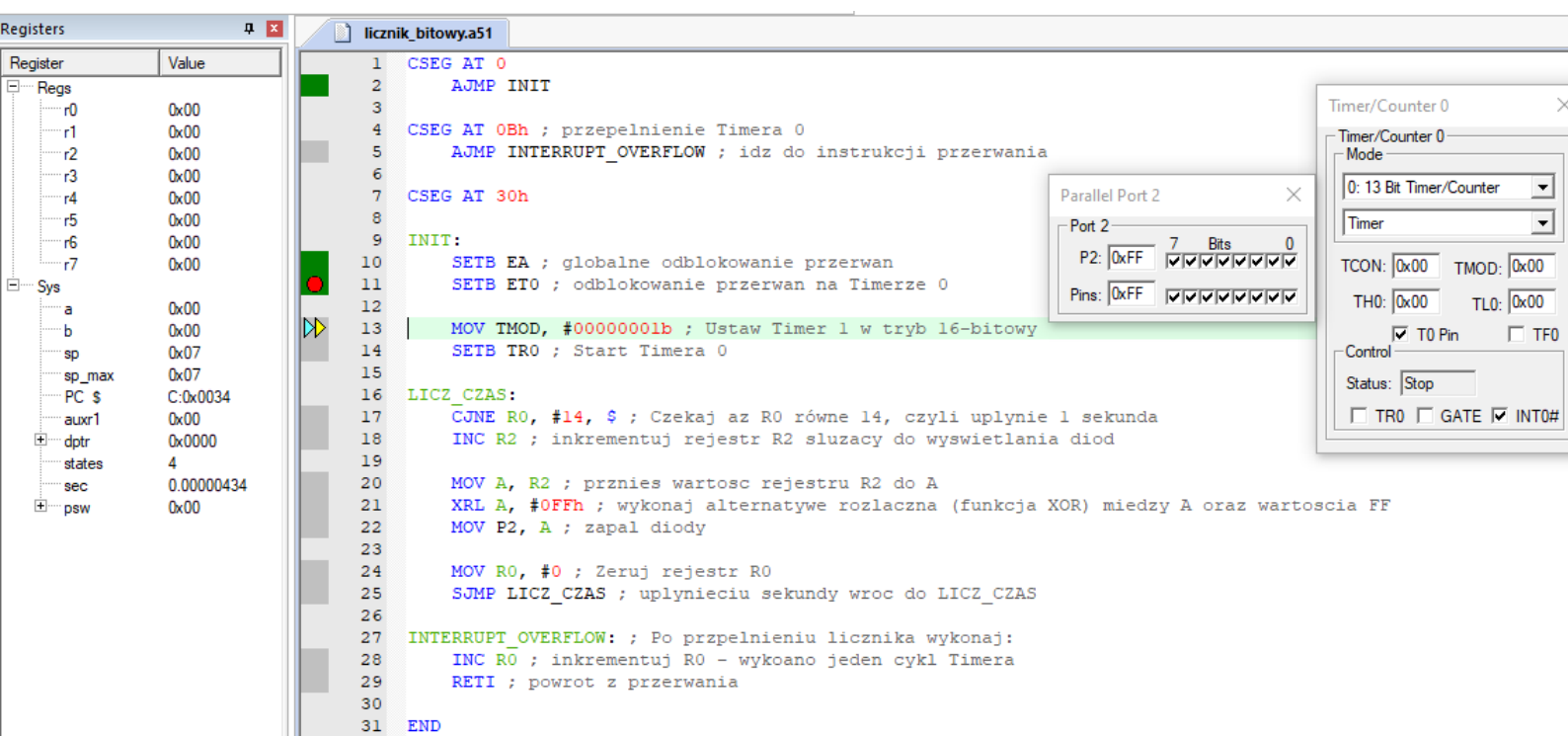
Obliczamy, że jedna sekunda wystąpi co:

$$\frac{921\,600 \text{ cykli}}{65\,536} = 14,0625 \text{ przepełnień zegara}$$

Po zaokrągleniu: 1 sekunda \approx 14 przepełnień zegara.

4. Opis programu:

1. Inicjalizacja wektora przerwań i ustawienie przerwania wywołanego Timerem 0;



Rys. 1. Inicjalizacja programu

2. Ustawienie Timera 0 w tryb licznika 16-bitowego i włączenie Timera 0;

The screenshot displays the Keil IDE interface for the assembly file `licznik_bitowy.a51`. The main window shows the assembly code with the following key sections:

- INIT:** Configures the timer and starts it.
 - `SETB EA`; globalne odblokowanie przerwan
 - `SETB ET0`; odblokowanie przerwan na Timerze 0
 - `MOV TMOD, #00000001b`; Ustaw Timer 1 w tryb 16-bitowy
 - `SETB TR0`; Start Timera 0
- LICZ_CZAS:** Main loop for counting.
 - `CJNE R0, #14, $`; Czeka az R0 równe 14, czyli uplynie 1 sekunda
 - `INC R2`; inkrementuj rejestr R2 sluzacy do wyswietlania diod
 - `MOV A, R2`; prznies wartosc rejestru R2 do A
 - `XRL A, #0FFh`; wykonaj alternatywe rozlaczna (funkcja XOR) miedzy A oraz wartoscia FF
 - `MOV P2, A`; zapal diody
 - `MOV R0, #0`; Zeruj rejestr R0
 - `SJMP LICZ_CZAS`; uplyniecie sekundy wroc do LICZ_CZAS
- INTERRUPT_OVERFLOW:** Interrupt service routine.
 - `INC R0`; inkrementuj R0 - wykoano jeden cykl Timera
 - `RETI`; powrot z przerwania

The **Registers** window on the left shows the state of registers, with `PC` at `0x0039` and `sec` at `0.00000760`. The **Timer/Counter 0** configuration window on the right shows the timer set to **1: 16 Bit Timer/Counter** mode, with `TCON` at `0x10`, `TMOD` at `0x01`, `TH0` at `0x00`, and `TL0` at `0x01`. The **Parallel Port 2** window shows `P2` at `0xFF` and `Pins` at `0xFF`.

Rys. 2. Ustawienie Timera

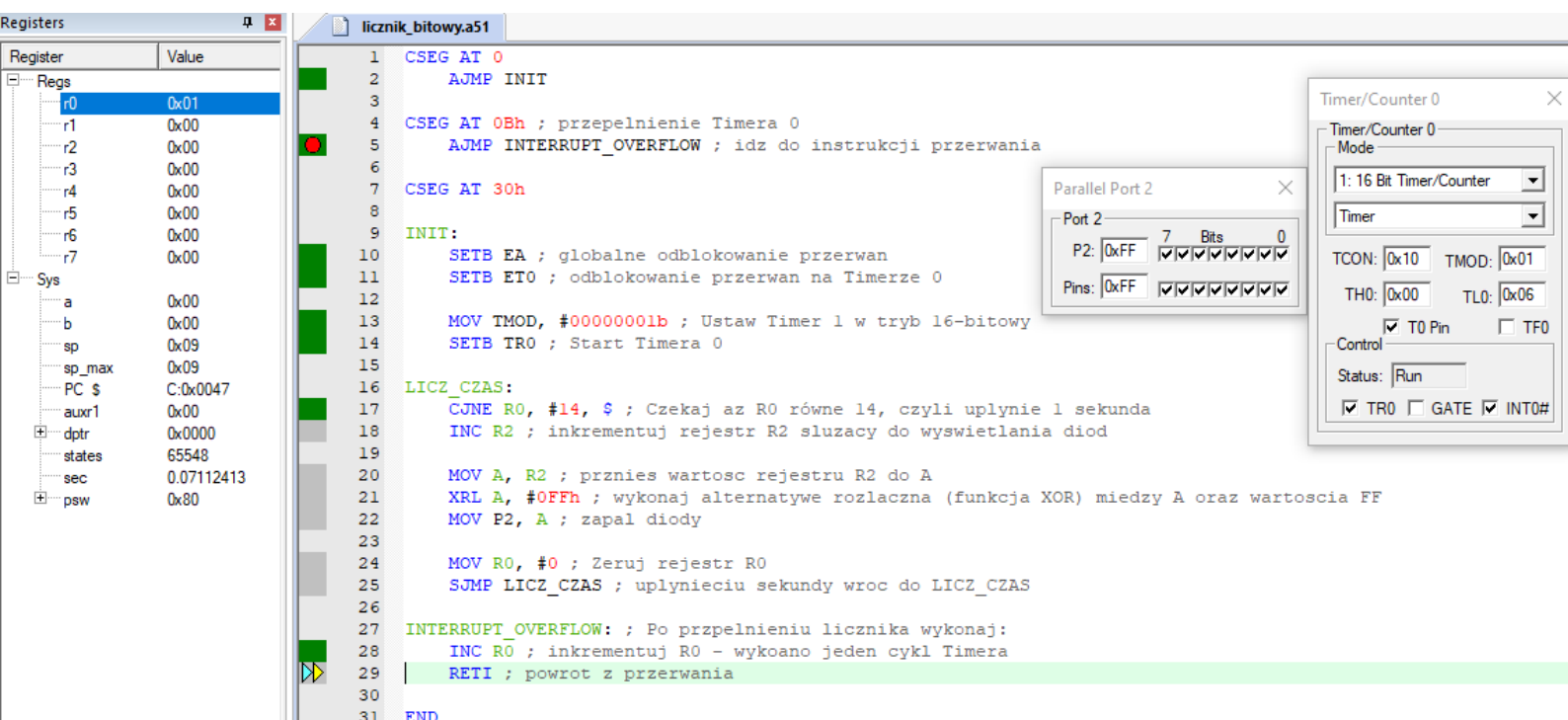
3. Następnie porównujemy wartość R0 z wartością 14 i jeżeli jest taka sama, to program idzie dalej. Jeśli nie, czeka na przepełnienie Timera 0;

The screenshot displays the Keil IDE interface for the assembly file `licznik_bitowy.a51`. The main window shows the assembly code with line numbers 1 through 31. The code includes comments in Polish describing the timer configuration and the loop logic. The 'Registers' window on the left shows the state of registers, with `sp_max` at `0x09`. The 'Timer/Counter 0' window on the right shows the timer is configured as a 16-bit timer in mode 1 (TMOD: 0x01) and is currently running.

```
1 CSEG AT 0
2 AJMP INIT
3
4 CSEG AT 0Bh ; przepelnienie Timera 0
5 AJMP INTERRUPT_OVERFLOW ; idz do instrukcji przerwania
6
7 CSEG AT 30h
8
9 INIT:
10 SETB EA ; globalne odblokowanie przerwan
11 SETB ET0 ; odblokowanie przerwan na Timerze 0
12
13 MOV TMOD, #00000001b ; Ustaw Timer 1 w tryb 16-bitowy
14 SETB TR0 ; Start Timera 0
15
16 LICZ_CZAS:
17 CJNE R0, #14, $ ; Czekaj az R0 rowne 14, czyli uplynie 1 sekunda
18 INC R2 ; inkrementuj rejestr R2 sluzacy do wyswietlania diod
19
20 MOV A, R2 ; prznies wartosc rejestru R2 do A
21 XRL A, #0FFh ; wykonaj alternatywe rozlaczna (funkcja XOR) miedzy A oraz wartoscia FF
22 MOV P2, A ; zapal diody
23
24 MOV R0, #0 ; Zeruj rejestr R0
25 SJMP LICZ_CZAS ; uplyniecie sekundy wroc do LICZ_CZAS
26
27 INTERRUPT_OVERFLOW: ; Po przepelnieniu licznika wykonaj:
28 INC R0 ; inkrementuj R0 - wykoano jeden cykl Timera
29 RETI ; powrot z przerwania
30
31 END
```

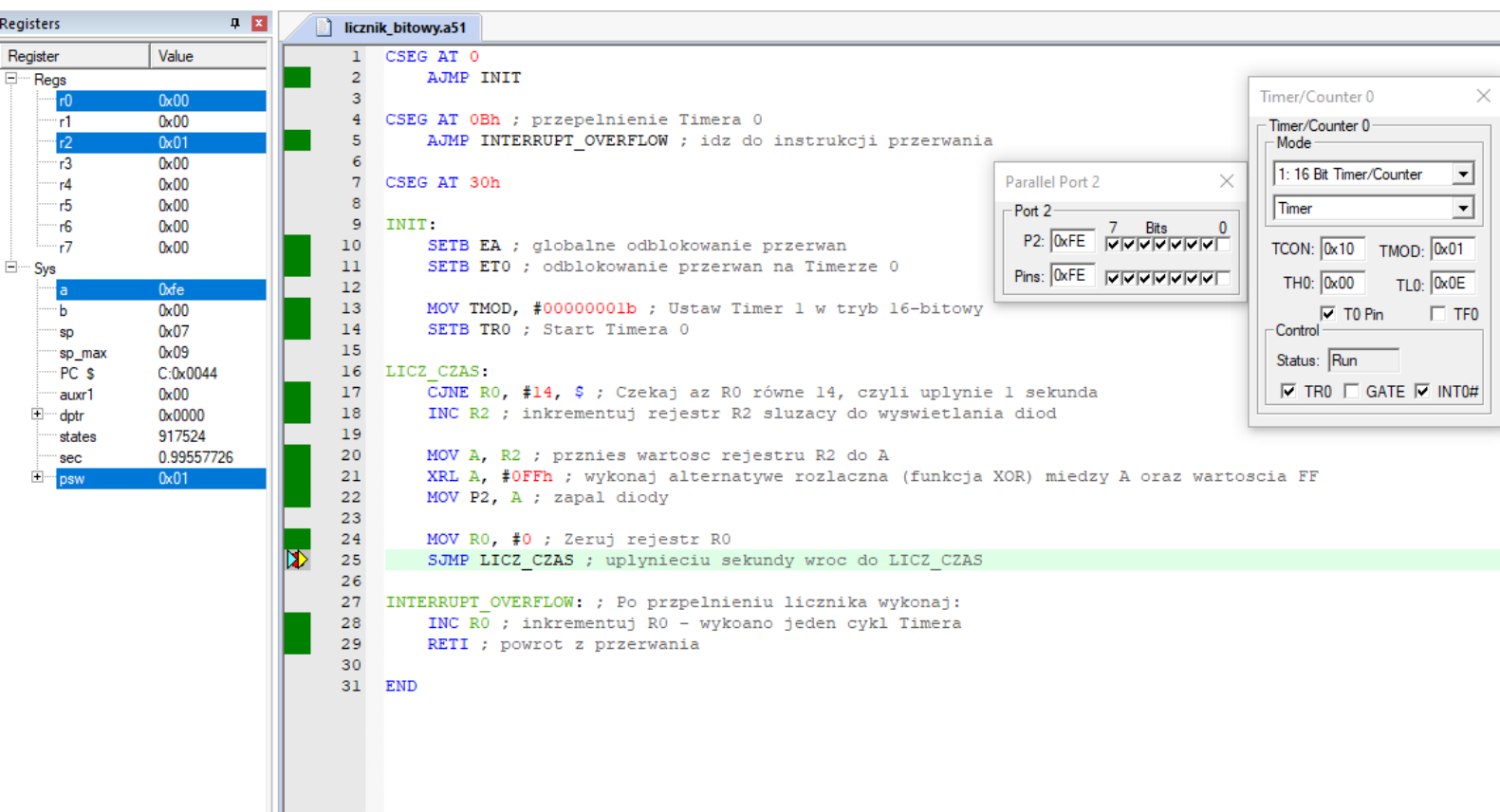
Rys. 3. Oczekiwanie na przepełnienie Timera

4. Jeśli nastąpi przepełnienie Timera, wtedy wywołane jest przerwanie 0Bh i program przechodzi do etykiety **INTERRUPT_OVERFLOW**;



Rys. 4. Obsługa przerwania

5. W przerwaniu inkrementowany jest licznik R0 i następuje powrót z przerwania za pomocą komendy RETI;
6. Minie jedna sekunda, kiedy przerwanie zostanie wywołane 14 razy. Wówczas inkrementowany jest rejestr R2;
7. Za pomocą instrukcji **XRL** (alternatywa rozłączna) następuje maskowanie diod do zapalenia, gdyż **Intel 8080** stosuje logikę ujemną;
8. Następnie wyświetla za pomocą diod P2 wartość binarną rejestru, po czym zeruje wartość **R0**, by mogła znów służyć liczeniu przepełnień;



Rys. 5. Wyświetlanie sekund za pomocą diody

9. Następnie program wraca do etykiety **LICZ_CZAS** i zaczyna się zliczanie kolejnej sekundy.

To rozwiązanie ma tę zaletę, że liczenie przepełnień timera jest niezależne od reszty programu, dzięki czemu zegar powstały w ten sposób liczy czas bardzo dokładnie.

Gdy licznik sekund osiągnie maksymalną wartość (wszystkie diody zapalone, a następnie dojdzie do przepełnienia), liczenie sekund zacznie się od nowa (od wartości 1).

4. Podsumowanie i wnioski:

Podsumowując laboratorium należy uznać, że przerwania są ważnym podsystemem, który oferuje wiele możliwości. Pozwalają one wykonywać krytyczne operacje, niezależnie od tych wykonywanych aktualnie. Należy pamiętać, że powinny one implementować możliwie jak najmniej złożone czasowo operacje, by być wykonywane jak najszybciej. Konfiguracja przerwań odbywa się w prosty sposób przy pomocy rejestrów. Źródła przerwań mogą mieć zarówno zewnętrzne, jak i wewnętrzne czynniki. Ich użycie pozwala realizować, np. zadania, które muszą być wykonane w jednakowych odstępach czasowych.