	LABORATORIUM ***PODSTAWY TECHNIKI MIKROPROCESOROWEJ***
L II	***Symulator***

I. SYMULACJA PROGRAMOWA PRACY MIKROKOMPUTERA

Po zakończeniu etapu pisania programu i uzyskaniu pliku końcowego (plik z rozszerzeniem .hex), umożliwiającego zaprogramowanie mikroprocesorowej pamięci programu, należy sprawdzić poprawność funkcjonowania napisanego programu. Do tego celu można użyć symulatora, który umożliwia wyeliminowanie z programu ewentualnych błędów logicznych. Symulator pozwala dokładnie, krok po kroku, prześledzić pracę procesora wykonującego program użytkownika. Ponieważ wszystkie zasoby mikrokontrolera ’51 są symulowane programowo przez komputer PC, dlatego *czas wykonywania kroków symulacyjnych jest o wiele dłuższy od czasu wykonywania programu użytkownika w warunkach rzeczywistych*. Czas ten w znacznej mierze zależy także od szybkości działania komputera PC. Nie można również w pełni zasymulować rzeczywistych sygnałów otoczenia dochodzących do mikrokontrolera. Możliwe jest tylko ustawianie w odpowiednim miejscu programu symulującego sygnałów logicznych na wirtualnych wyprowadzeniach.

II.SYMULOWANIE PROJEKTU

1. Uruchomienie programu µVISION

Uruchom program µVISION dwukrotnie klikając w ikonę skrótu 

umieszczoną na pulpicie.

2. Otwarcie projektu *Nowy* utworzonego na poprzednich zajęciach

Z menu *Project* wybierz opcję *Close Project...*, w celu zamknięcia projektu domyślnego.
Z menu *Project* wybierz opcję *Open Project...*, w celu otwarcia projektu zachowanego na poprzednich zajęciach (katalog *d:\Student\Lxx* , plik z rozszerzeniem *.Uv2*).

3. Ustawienie opcji symulatora

Klikając prawym klawiszem myszy na folderze *Target 1* w okienku menadżera projektów rozwiń lokalne menu i wybierz opcję *Options for Target ’Target 1’*. Ustaw odpowiednio parametry przetwarzania pliku źródłowego.

A. Zakładka *Device*:
- wybierz odpowiedni mikroprocesor; Analog Devices ADuC845.

B. Zakładka *Target*:
- ustaw częstotliwość rezonatora kwarcowego na 0.032768MHz.

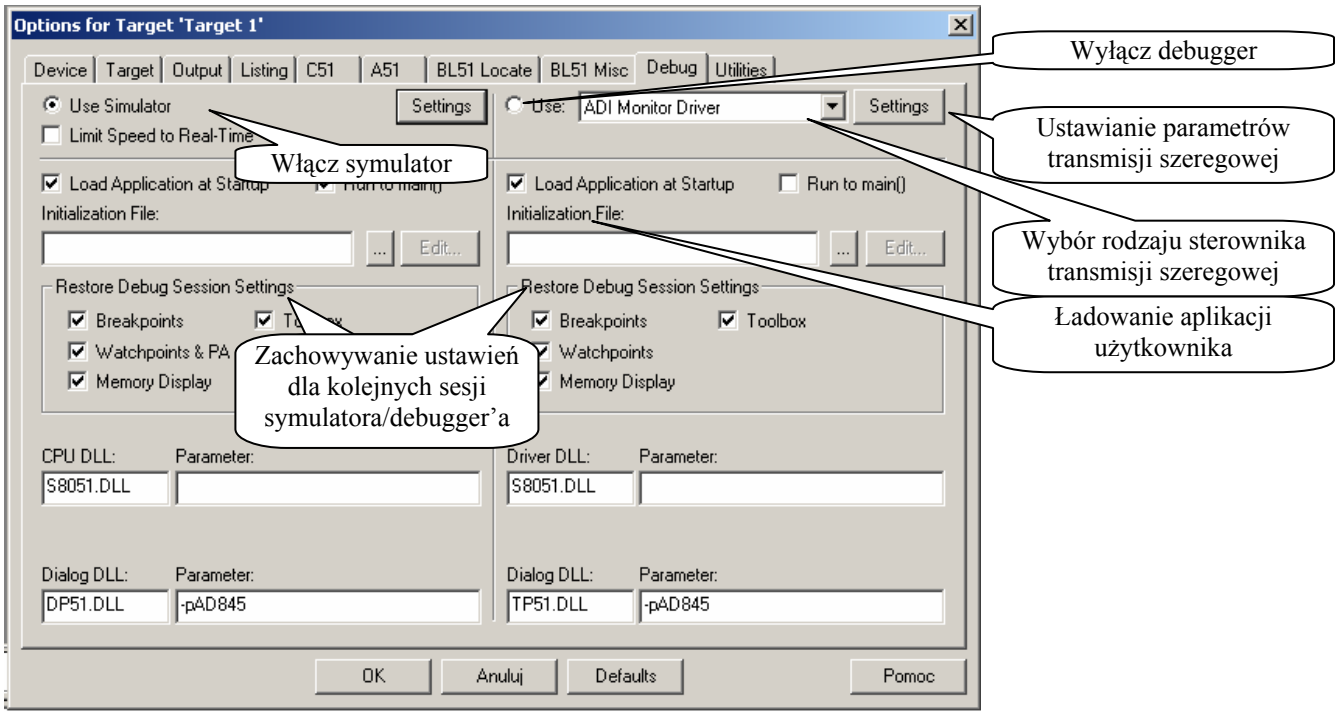
C. Zakładka *Output*:
- wybierz odpowiedni katalog dla zbiorów wynikowych;
- nadaj odpowiednią nazwę zbiorom wynikowym;
- włącz tworzenie zbioru *.hex* (HEX−80);
- włącz dołączanie informacji dla debuger’a;
- włącz tworzenie listingu.

D. zakładka *Listing*:
- wybierz odpowiedni katalog dla zbiorów z listingami;

- ustaw zgodnie z potrzebami opcje listingu asemblera;
- ustaw zgodnie z potrzebami opcje listingu linkera.

E. Zakładka *Debug*:

- ustaw opcje debugger'a zgodnie z poniższym rysunkiem.



4. Przeprowadzenie ponownej kompilacji

Używając menu *Project/Rebuild all target files* przeprowadź ponownie proces kompilacji, dzięki czemu zostaną uwzględnione zmiany wprowadzone w punkcie 3.

5. Uruchomienie symulatora

A. Korzystając z menu Debug / Start/Stop Debug Session uruchom symulator.

B. Odszukaj poszczególne rejestry.

- W okienku menadżera projektów pojawi się zestaw dostępnych rejestrów mikrokontrolera:

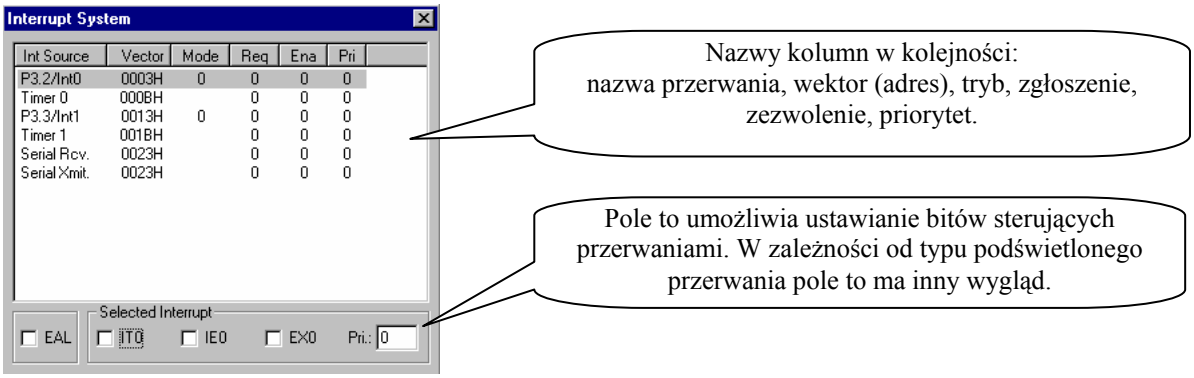
✓ rejestry robocze (R0 − R7),
✓ rejestry systemowe (akumulator A, rejestr B, wskaźnik stosu SP, wskaźnik danych DPTR, licznik rozkazów PC, rejestr stanu PSW z jego poszczególnymi bitami).

UWAGA

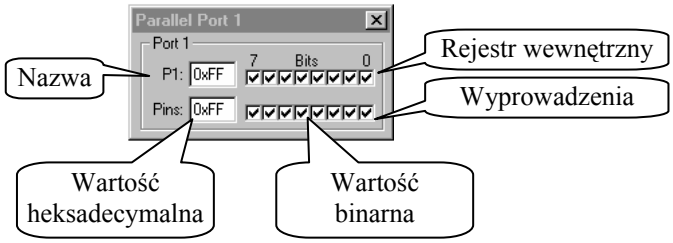
Sprawdź, jakie znaczenie mają poszczególne bity rejestru PSW.

C. Zapoznaj się z menu Peripherals:

✓ Reset CPU (zerowanie wirtualnego mikrokontrolera, powrót do ustawień początkowych),
✓ Serial (rejestry i bity sterujące portem szeregowym),
✓ Timer (rejestry i bity sterujące układem czasowo − licznikowym),
✓ Interrupt (ustawienia systemu przerwań mikrokontrolera),



✓ *I/O−Ports* (rejestry wewnętrzne i wyprowadzenia z układu scalonego portów wejścia − wyjścia).



UWAGA

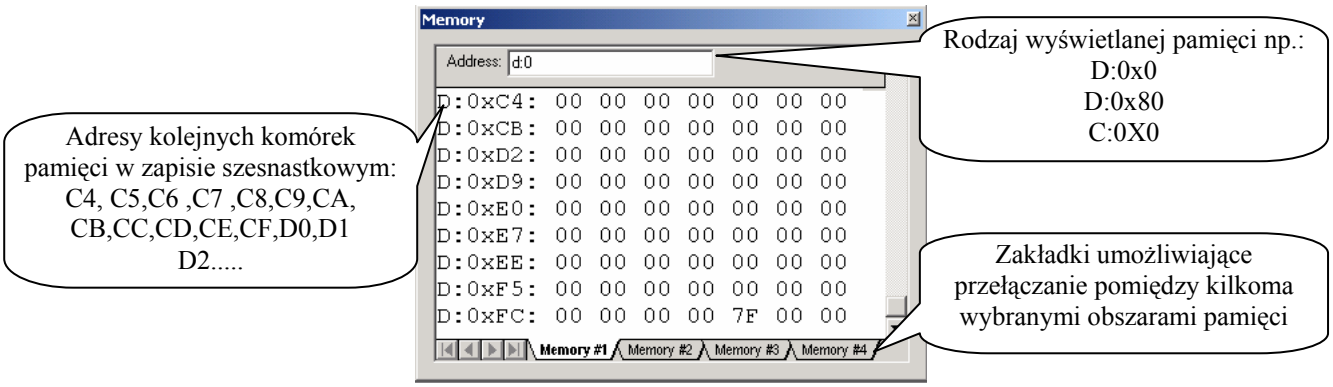
Wszystkie jasne pola można zmieniać poprzez zaznaczenie lub wpisując odpowiednie wartości.

D. Wyświetl kod źródłowy programu.

W trakcie symulacji w okienku edytora może być wyświetlany kod źródłowy programu lub kod wynikowy poddany procesowi disasemblacji. Przełączenia można dokonać wykorzystując menu *View / Disassembly Window*. Dalszy przebieg symulacji zostanie przedstawiony w okienku zawierającym program po disasemblacji.

E. Przygotuj pulpit roboczy.

- Pracę z symulatorem może znacznie ułatwić pasek narzędziowy *Debuger'a*, otwierany z menu *View / Debug Toolbar* .
- Z menu *View / Memory Window* otwórz okienko wyświetlające zawartość komórek pamięci (okienko to można dowolnie formatować i przemieszczać po ekranie).



UWAGA

- ✓ Sprawdź zawartość pamięci programu poczynając od adresu C:0x00 oraz C:0x200.
- ✓ Sprawdź zawartość pamięci danych w obszarze SFR tzn. D:0x80.

6. Przebieg procesu symulacji

A. Korzystając z menu *Peripherals / Reset CPU* wyzeruj mikrokontroler.

B. Symulacja krok po kroku

- W celu wykonania kolejnego rozkazu programu należy wcisnąć klawisz funkcyjny **F11**, skorzystać z menu *Debug/Step* lub paska narzędziowego *Debug Toolbar*.
- Wciśnij **F11** => wykonanie wskazanego strzałką rozkazu LJMP START (C:0200) (skocz do adresu 200h w pamięci programu).
- ✓ ***Zwróć uwagę na***: zawartość rejestru PC (licznika rozkazów) przed i po naciśnięciu klawisza.
- Wciśnij **F11** => wykonanie wskazanego rozkazu MOV SP(0x81),#0x60 (wpisz do rejestru SP wartość 60h ⇔ ustaw wskaźnik stosu).
- ✓ ***Zwróć uwagę na***: zawartość rejestru SP (wskaźnika stosu) przed i po naciśnięciu klawisza oraz obszar pamięci wewnętrznej D:0x7F w okienku *Memory* (w szczególności komórkę 0x81). Cały czas obserwuj zawartość licznika rozkazów PC.
- Wciśnij **F11** => wykonanie rozkazu MOV R0,#0x7F (wpisz do rejestru R0 wartość 7Fh).
- ✓ ***Zwróć uwagę na:*** zawartość rejestru R0 i pierwszą komórkę pamięci wewnętrznej (obszar D:0x00).
- Wciśnij **F11** => wykonanie rozkazu CLR A (wyzeruj akumulator).

✓ ***Zwróć uwagę na***: zawartość rejestru A.

- Spróbuj wprowadzić inną wartość szesnastkową do akumulatora (należy zaznaczyć rejestr A i po chwili ponownie kliknąć w liczbę określającą jego zawartość) i dopiero wtedy wykonaj rozkaz.

C. Symulacja krok po kroku – *PĘTLA ZERUJĄCA PAMIĘĆ WEWNĘTRZNĄ*

- W okienku *Memory* kliknij prawym klawiszem myszy na dowolnej wartości określającej zawartość komórki pamięci. Otworzy się wówczas podręczne menu, z którego można wybrać funkcją *Modify Memory at D:0x...* umożliwiającą zmianę zawartości wskazanej komórki pamięci. W ten sposób zmodyfikuj wartości przypisane adresom 70h,73h,78h,7Ah,7Bh,7Ch,7Dh,7Eh,7Fh. Nie zmieniaj zawartości rejestru R0, czyli pierwszej komórki pamięci oraz rejestrów z obszaru SFR.
- Wciśnij **F11** => wykonanie rozkazu MOV @R0,A (do komórki wewnętrznej pamięci danych o adresie wskazanym zawartością rejestru R0 przepisz zawartość akumulatora ⇔ zeruj komórkę pamięci).

✓ ***Zwróć uwagę na***: zawartość ostatniej komórki pamięci o adresie 7Fh − została wyzerowana.

- Wciśnij **F11** => wykonanie rozkazu DJNZ R0,RESET (C:0206) (zmniejsz o 1 zawartość rejestru R0 i jeżeli jest ona różna od zera to skocz do adresu 206h).

✓ ***Zwróć uwagę na***: zawartość rejestru R0 i licznika rozkazów PC.

- Wciskając **F11** wykonaj kolejne kroki pętli (128 kroków), aż do wyzerowania całej pamięci wewnętrznej.

✓ ***Zwróć uwagę na***: zawartość rejestru R0.

D. Symulacja krok po kroku – *INICJOWANIE PRACY SYSTEMU*

- Wciśnij **F11** –> wykonanie rozkazu ORL IE,#1000$0010b (ustaw na jeden wskazane jedynką bity rejestru IE ⇔ ustaw system przerwań).

✓ ***Zwróć uwagę na***: bity sterujące przerwaniami (bity EAL oraz ET0 w okienku *Interrupt System* − p5.).

UWAGA

Zapis rozkazu poddanego disasemblacji może nieznacznie się różnić od odpowiadającej mu linijce w kodzie źródłowym programu. W celu zwiększenia czytelności przetwarzanego kodu Disasembler zastępuje liczby nazwami rejestrów oraz wstawia informacje dodatkowe, co nie zawsze ma uzasadnienie logiczne. Nie mniej jednak kod rozkazu nie ulega zmianie.

- Wciśnij **F11** => wykonanie rozkazu ORL TMOD,#0000$0001b (ustaw na jeden wskazane bity rejestru TMOD ⇔ ustaw tryb pracy licznika T0)

✓ ***Zwróć uwagę na***: licznik T0 (okienko *Timer/Counter 0*− p.5).

- Wciśnij **F11** => wykonanie rozkazu MOV TH0 (0x8C),#0xD8 (wpisz do rejestru TH0 wartość 0D8h ⇔ wpisz wartość początkową do starszego rejestru licznika T0).

✓ ***Zwróć uwagę na***: licznik T0 (okienko Timer/Counter 0).

- Wciśnij **F11** => wykonanie rozkazu MOV TL0(0x8A),#0xEF (wpisz do rejestru TL0 wartość 0EFh ⇔ wpisz wartość początkową do młodszego rejestru licznika T0).

✓ ***Zwróć uwagę na***: licznik T0 (okienko Timer/Counter 0).

- Wciśnij **F11** => wykonanie rozkazu SETB TR0(0x88.4) (ustaw bit TR0 ⇔ włącz zliczanie licznika T0).

✓ ***Zwróć uwagę na***: licznik T0 (po wykonaniu każdego rozkazu będzie zwiększał swoją zawartość).

- Wciśnij **F11** => wykonanie rozkazu MOV P1(0x90),#0x00 (wyzerowanie portu P1).

✓ ***Zwróć uwagę na***: zwróć uwagę na bity portu P1 (okienko *Parallel Port 1* − p.5).

E. Symulacja krok po kroku – *PĘTLA GŁÓWNA PROGRAMU*

- Wciśnij **F11** => wykonanie rozkazu CJNE R1,#0x19,PETLA (C:021A) (porównaj zawartość rejestru R1 z liczbą 19h i jeżeli liczby te są różne to skocz do tej samej linii, czyli ponownie wykonaj rozkaz).

✓ ***Zwróć uwagę na***: zawartość rejestru R1 − jak wykona się rozkaz w tym kroku?.

- Kolejne przyciśnięcia klawisza **F11** powodują zwiększanie zawartości licznika T0. W celu przyspieszenia procesu symulacji do starszego rejestru licznika T0 wpisz wartość 0FFh. Kolejne przyciśnięcia klawisza **F11** spowodują szybkie przepełnieni licznika T0.

✓ ***Zwróć uwagę na***: moment przepełnienia licznika T0, kiedy zawartość rejestru TL0 zmienia się z 0xFF na 0x00.

- W momencie przepełnienia licznika następuje skok do procedury obsługi przerwania od licznika T0. Wykonywane są wówczas, w sposób sprzętowy niezależny od użytkownika, następujące kroki:

✓ wyzerowanie licznika T0, rejestrów TH0 i TL0;
✓ zwiększenie wskaźnika stosu SP o1;
✓ wpisanie na stos młodszego bajtu z licznika rozkazów PC, pod adres w wewnętrznej pamięci danych (okienko *Memory*) wskazany przez SP;
✓ zwiększenie wskaźnika stosu SP o1;
✓ wpisanie na stos starszego bajtu z licznika rozkazów PC, pod adres w wewnętrznej pamięci danych (okienko *Memory*) wskazany przez SP;
✓ wpisanie do licznika rozkazów adresu 000Bh.
✓ ***Zwróć uwagę na***: Zwróć uwagę na zawartość rejestrów PC, SP, komórek wewnętrznej pamięci danych o adresie 60h, 61h, 62h.

F. Symulacja krok po kroku – *OBSŁUGA PRZERWANIA*

- Wciśnij **F11** => wykonanie rozkazu LJMP INT_T0 (C:0223) (skocz do adresu 0223h).

✓ ***Zwróć uwagę na***: zawartość licznika rozkazów PC.

- Wciśnij **F11** => wykonanie rozkazu MOV TH0(0x8C),#0xD8 (wpisz wartość początkową do starszego rejestru licznika T0).

✓ ***Zwróć uwagę na:*** licznik T0.

- Wciśnij **F11** => wykonanie rozkazu MOV TL0(0x8A),#0xEF (wpisz wartość początkową do młodszego rejestru licznika T0).

✓ ***Zwróć uwagę na:*** licznik T0.

- Wciśnij **F11** => wykonanie rozkazu INC R1 (zwiększ o 1 zawartość rejestru R1 ⇔ inkrementacja).

✓ ***Zwróć uwagę na***: licznik R1 i pierwszą komórkę pamięci wewnętrznej.

- Wciśnij **F11** => wykonanie rozkazu RETI – powrót z przerwania. Wykonywane są wówczas, w sposób sprzętowy niezależny od użytkownika, następujące kroki:

✓ odczytanie ze stosu (spod adresu wskazanego przez rejestr SP) starszego bajtu licznika rozkazów PC,
✓ zmniejszenie wskaźnika stosu SP o1,
✓ odczytanie ze stosu (spod adresu wskazanego przez rejestr SP) młodszego bajtu licznika rozkazów PC,
✓ zmniejszenie wskaźnika stosu SP o1.
✓ ***Zwróć uwagę na***: zawartość rejestrów PC, SP, licznika T0.

G. Symulacja krok po kroku – *POWRÓT DO PĘTLI GŁÓWNEJ PROGRAMU*

- Wciśnij **F11** => wykonanie rozkazu CJNE R1,#0x19,PETLA (C:021A) (porównaj zawartość rejestru R1 z liczbą 19h i jeżeli liczby te są różne to skocz do tej samej linii, czyli ponownie wykonaj rozkaz). Kolejne rozkazy zostaną wykonane dopiero wtedy, kiedy do rejestru R1 zostanie wpisana wartość 19h. Ponieważ zawartość rejestru R1 jest zwiększana w przerwaniu, dlatego dopiero po 19h przerwaniach nastąpi przejście do dalszej części programu. W celu przyspieszenia procesu symulacji do rejestru R1 można wpisać liczbę 19h.
- Wciśnij **F11** => wykonanie rozkazu MOV R1,#0x00 – wyzerowanie programowego licznika przerwań.

✓ ***Zwróć uwagę na***: zawartość rejestru R0.

- Wciśnij **F11** => wykonanie rozkazu CPL 0x90.0 − neguj zerową linię portu P1.

✓ ***Zwróć uwagę na***: zawartość portu P1.

- Wciśnij **F11** => wykonanie rozkazu SJMP PETLA(C:021A) – koniec pętli głównej programu.

✓ ***Zwróć uwagę na***: zawartość licznika rozkazów PC.

UWAGA

Utwórz i uzupełnij tabelę, której wzór został podany poniżej.

Kolejne rozkazy	Sposób wykonania rozkazów Wypisz wartości rejestrów, które uległy zmianie	
	Przed wykonaniem rozkazu	Po wykonaniu rozkazu

Uwzględnij kolejne kroki związane z symulacją programu.
ljmp Start;
mov SP,#STOS
mov R0,#IRAM_roz
PĘTLA ZERUJĄCA
clr A
mov @R0,A
djnz R0,Reset
INICJALIZACJA
mov IE,#1000$0010B
mov TMOD,#0000$0001B
mov TH0,#HIGH(T0_rel)
mov TL0,#LOW(T0_rel)
setb TR0
mov P1,#0
PĘTLA GŁÓWNA
cjne R1,#25,Petla
mov R1,#0
cpl P1.0
jmp Petla
PRZERWANIE
stos
licznik rozkazów
ljmp Int_T0
mov TH0,#HIGH(T0_rel)
mov TL0,#LOW(T0_rel)
inc R1
reti
	PC =
	....
		PC =
		....

III. DODATKOWE ZAGADNIENIA

A. Dodatkowe funkcje symulatora

- Proces symulacji można kontynuować korzystając z innych funkcji menu *Debug:*

✓ *Go* − symulacja ciągła;
✓ *Step Over* − symulacja krokowa bez wchodzenia do podprogramów;
✓ *Insert/Remove Breakpoint* − wstawianie punktów zatrzymań.

UWAGA

Punkty zatrzymań mają na celu wspomaganie procesu symulacji. Punkt taki jest przypisywany do rozkazu, i jeżeli zostanie wykryty, następuje zatrzymanie procesu symulacji. W celu ustawienia punktu „*Breakpoint"* należy podświetlić rozkaz, przy wykonaniu, którego proces symulacji ma zostać zatrzymany, a następnie wybrać funkcję *Debug / Insert/Remove Breakpoint*. Wskazana linijka w programie zostanie zaznaczona czerwonym punktem.

- Zmiany zawartości licznika rozkazów PC

Wprowadzenie nowych, dowolnych wartości do rejestru PC (np. zwiększ jego zawartość o 1, o 2, lub o 3), może spowodować całkowicie błędną interpretację wykonywanych rozkazów. Po dokonaniu takiej zmiany, w obszarze programu użytkownika pojawią się zupełnie inne rozkazy, nie należące do symulowanego programu. Wynika to z bajtowej struktury kodu rozkazu:

✓ licznik PC wskazuje kolejne bajty w pamięci,
✓ jeden rozkaz może zajmować 1, 2 lub 3 bajty w pamięci.

Po wykonaniu takiej próby należy wpisać poprawną wartość do licznika PC lub wyzerować mikrokontroler.

UWAGA

Ustaw „Breakpoint" na rozkazie LJMP INT_T0. Używając funkcji *Go* przeprowadź proces symulacji. Obserwuj rejestr R1 i wyjścia portu P1.

IV. PRZYGOTOWANIE DO NASTĘPNYCH ZAJĘĆ

1. Wiedza teoretyczna

A. Cykl rozkazowy mikrokontrolera.

B. Współpraca mikrokontrolera z otoczeniem.

C. Organizacja wewnętrznej oraz zewnętrznej pamięci danych.

D. Organizacja pamięci programu.

2. Wiadomości z ćwiczenia drugiego

A. Struktura i znaczenie rozkazów użytych w przykładowym programie.

B. Budowa rejestrów sterujących użytych w przykładzie, znaczenie poszczególnych bitów.

C. Umiejętność posługiwania się symulatorem µVISION w zakresie wskazanym w instrukcji do ćwiczenia.