Technika Mikroprocesorowa

Laboratorium 3

Podprogramy,procedury, funkcje, zmienne

Cel ćwiczenia: Głównym celem ćwiczenia jest wprowadzenie porządku w pisanych programach czyli wprowadzenia podprogramów. Dzięki podprogramom można ten sam fragment kodu wykorzystać wielokrotnie w różnych częściach programu. Na przykład jeśli chcemy dodać zmienne A+B a następnie B+C możemy napisać kod dodający A i B a następnie skopiować ten kod, lekko go zmodyfikować i dodać do B zmienną C, lub napisać podprogram/procedurę która dodaje dwie liczby i wywołać ją dwa razy z argumentami A i B a następnie z argumentami B i C.
Podprogram procedura czy funkcja ma zawsze początek i koniec oraz może mieć argumenty wejściowe i wyjściowe. Przyjmijmy, że na potrzeby niniejszego kursu będziemy tylko używać pojęcia procedura. A więc pod pojęciem procedury będziemy rozumieć fragment programu (podprogram) który ma lub nie ma argumentów wejściowych i analogicznie posiada lub też nie argumenty wyjściowe.

Zacznijmy od najprostszego przypadku napiszmy fragment kodu który będziemy nazywać procedurą.
Niech nasza procedura nazywa się PUSH

```
PUSH:           ;początek procedury jak widać jest to klasyczna etykieta
                ;ciało procedury czyli zestaw instrukcji wykonywanych w naszej procedurze.
      CLR  P1.7 ;zeruj linię 7 w porcie P1
                ;nasza procedura jest bardzo prosta ale ma wszystko: początek, ciało, koniec.
RET             ;koniec jest to instrukcja która kończy procedurę i powoduje powrót do miejsca jej wywołania np. do programu głównego lub procedury wywołującej.
```

Cały program dla ww. procedury będzie wyglądał tak

pętla główna

```
        LJMP START

        ORG 100H     ;Program główny
START:
        CALL ZAPAL         ; wywołanie procedury
          INC R1     ;przykładowa instrukcja w programie głównym
        CALL ZAPAL         ;wywołanie procedury

        LJMP START ; zapętlenie programu głównego tak by wnętrze było wykonywane non stop
END:
```

procedura

```
          ;Procedura o nazwie ZAPAL
        ORG 200H     ;procedura zostanie umieszczona w pamięci programu od adresu 200H
ZAPAL:               ;początek procedury
                     ;ciało procedury
        CLR  P1.7    ;zeruj linię 7 w porcie P1
        RET          ;koniec procedury
```

Procedura z parametrami

```
AL SET 50H ;Zdefiniowanie zmiennej AL pod adresem 50H – pierwszy argument sumowania
BL SET 51H ;Zdefiniowanie zmiennej BL pod adresem 51H – drugi argument sumowania
CL SET 52H ;Zdefiniowanie zmiennej CL pod adresem 52H – wynik sumowania
CL2 SET 53H ;Zdefiniowanie zmiennej CL2 pod adresem 53H – wynik sumowania

        LJMP START

        ORG 100H
START:
        MOV AL, #10     ;przypisanie wartości pierwszemu argumentowi
        MOV BL, #12     ;przypisanie wartości drugiemu argumentowi
        CALL SUMA       ;wywołanie procedury sumującej
        MOV AL, P0      ;przypisanie wartości pierwszemu argumentowi
        MOV BL, P1      ;przypisanie wartości drugiemu argumentowi
        CALL SUMA       ;ponowne wywołanie procedury sumującej
        MOV A,CL        ;
        MOV CL2,A       ;przepisanie wyniku sumowania do CL2 pamięć wewnętrzna 53H
        LJMP START      ;powrót do początku pętli głównej
END:                    ;ta etykieta w programie jest używana, można ją pominąć, jednak
zamieszczono ją by zaznaczyć koniec programu głównego

;Procedura z dwoma argumentami wejściowymi AL., BL i jednym wyjściowym CL
        ORG 200H        ;procedura zostanie umieszczona w pamięci programu od adresu 200H
SUMA:
        MOV A, AL       ; wpisanie jednego argumentu do akumulatora
        MOV R1, BL      ; wpisanie drugiego argumentu do rejestru uniwersalnego R1
        ADD A,R1        ;wyliczenie sumy A + R1
        MOV CL, A       ; wpisanie wyniku sumowania do zmiennej CL
        RET             ; powrót do programu głównego
```

W ramach ćwiczenia należy wykonać procedurę sumującą dwie liczby dwubajtowe.

Elementy wymagane przy sprawozdaniu:

- Napisany program z komentarzami (kod oraz opis programu)