| Maciej Byczko<br>Bartosz Matysiak | Prowadzący:<br>dr inż. Jacek Mazurkiewicz | Numer ćwiczenia<br>2 |
|---|---|---|
| Cz 13:15 TN | Temat ćwiczenia:<br>Arytmetyka, logika, pamięć, diody i brzęczyki | Ocena: |
| Grupa:<br>B | Data wykonania:<br>11 Marca 2021 | |

# 1 Zadanie 1

## 1.1 Polecenie

Przygotować program odejmujący dwie liczby 16-bitowe.

## 1.2 Rozwiązanie

```
1  ljmp start
2
3  org 0100h
4    start:
5      mov r1, #00h ;pierwszy 16-bitowy argument w r1-r0
6      mov r0, #05h
7
8      mov r3, #01h ;drugi 16-bitowy argument w r3-r2
9      mov r2, #03h
10
11     clr c        ;wyczysc bit pozyczki
12     mov a, r0    ;zaladuj do akumulatora mlodsze bity odjemnej
13     subb a, r2   ;odejmij mlodsze bity odjemnika
14     mov r4, a    ;zapisz mlodsze bity wyniku do r4
15
16     mov a, r1    ;zaladuj do akumulatora starsze bity odjemnej
17     subb a, r3   ;odejmij starsze bity odjemnika uwzgledniajac
           pozyczke
18     mov r5, a  ;zapisz wynik w r5
19
20   nop
21   nop
22   nop
23   jmp $
24   end start
```

# 2 Zadanie 2

## 2.1 Polecenie

Przygotować program realizujący sortowanie bąbelkowe lub znajdujący minimum albo maksimum tablicy 1-wymiarowej rozpoczynającej się od adresu 8000H pamięci zewnętrznej danych (XRAM) i obejmującej 16 kolejnych komórek tej tablicy.

## 2.2 Rozwiązanie

```
ljmp start;

org 0100h;
  start:

  mov dptr, #8000h ;adres w pamieci zewnetrznej
  mov a, #0f0h     ;akumulator przechowuje bierzaca wartosc do
     wpisania
  zapis16:     ;petla zapisujaca 16 kolejnych wartosci bajtow 0x01,0
     x02,...,0x10
  add a, #10h      ;odtworzenie wartosci z poprzedniej iteracji
  inc a            ;utworzenie nowej wartosci
  movx @dptr, a    ;zapisanie wartosci pod adres
  inc dptr         ;inkrementacja rejestru przechowujacego adres w
     pamieci zewnetrzenej
  clr c            ;przygotowanie do operacji odejmowania
  subb a, #10h     ;odejmowanie, sluzy sprawdzeniu, czy osiagnieto
     koniec tablicy
  jc zapis16       ;skok, jesli nie osiagnieto konca tablicy

  mov a, #00h      ;znacznik konca tablicy (17-ty bajt)
  movx @dptr, a    ;zapisz znacznik konca tablicy


  //alternatywny kod, w ktorym rozpisano instrukcje
  //w razie, gdyby uzytkowanik mial potrzebe edycji wartosci bajtow
  /*
  mov dptr, #8000h;przejscie na adres 8000 oraz wpisanie liczb do
     pierwszych 16 komorek
  mov a, #0fh     ;wartosc do przechowania, przetestowano takze dla
     wartosci #05h
  movx @dptr, a;
  inc dptr;
  mov a, #02h;
  movx @dptr, a;
  inc dptr;
  mov a, #03h;
  movx @dptr, a;
  inc dptr;
  mov a, #04h;
  movx @dptr, a;
  inc dptr;
  mov a, #05h;
  movx @dptr, a;
  inc dptr;
  mov a, #06h;
  movx @dptr, a;
  inc dptr;
  mov a, #07h;
```

```
44  movx @dptr, a;
45  inc dptr;
46  mov a, #08h;
47  movx @dptr, a;
48  inc dptr;
49  mov a, #09h;
50  movx @dptr, a;
51  inc dptr;
52  mov a, #0ah;
53  movx @dptr, a;
54  inc dptr;
55  mov a, #0bh;
56  movx @dptr, a;
57  inc dptr;
58  mov a, #0ch;
59  movx @dptr, a;
60  inc dptr;
61  mov a, #0dh;
62  movx @dptr, a;
63  inc dptr;
64  mov a, #0eh;
65  movx @dptr, a;
66  inc dptr;
67  mov a, #0fh;
68  movx @dptr, a;
69  inc dptr;
70  mov a, #10h;
71  movx @dptr, a;
72  inc dptr;
73  mov a, #00h;
74  movx @dptr, a;
75  */
76
77  mov r0, #00h      ;maksimum w tablicy
78  mov dptr, #8000h  ;adres poczatkowy
79
80  odczyt16max:    ;petla odczytujaca 16 elementow tablicy (+ 17-ty,
       znacznik konca tablicy #00h)
81  movx a, @dptr   ;odczyt wartosci spod adresu
82  jz koniecmax
83  clr c           ;przygotowanie do odejmowania
84  subb a, r0      ;odejmowanie sprawdzajace, czy nowa wartosc jest
       wieksza od dotychczasowej
85  jc niekopiujmax   ;skok warunkowy, jesli nie jest wieksza
86  add a, r0     ;odtworzenie wartosci w rejestrze a
87  mov r0,a      ;przechowanie nowej maksymalnej wartosci
88  niekopiujmax:
89  movx a,@dptr    ;sprawdzenie, czy nie nastapil koniec tablicy
90  inc dptr
91  jnz odczyt16max
92
```

```
93   koniecmax:
94
95   mov r1, #0ffh      ;minimum w tablicy
96   mov dptr, #8000h ;adres poczatkowy
97
98   odczyt16min:       ;petla odczytujaca 16 elementow tablicy (+ 17-ty,
         znacznik konca tablicy #00h)
99   movx a, @dptr   ;odczyt wartosci spod adresu
100  jz koniecmin       ;dodatkowy skok sprawdzajacy, czy pobrano bajt
        #00h
101  clr c           ;przygotowanie do odejmowania
102  subb a, r1      ;odejmowanie sprawdzajace, czy nowa wartosc jest
        wieksza od dotychczasowej
103  jnc niekopiujmin    ;skok warunkowy, jesli nie jest wieksza
104  add a, r1      ;odtworzenie wartosci w rejestrze a
105  mov r1,a       ;przechowanie nowej maksymalnej wartosci
106  niekopiujmin:
107  movx a,@dptr    ;sprawdzenie, czy nie nastapil koniec tablicy
108  inc dptr
109  jnz odczyt16min
110
111  koniecmin:
112  ;teraz wartosc maksymalna znaduje sie w r0, a minimalna w r1
113  nop;
114  nop;
115  nop;
116  jmp $;
117  end start;
```

# 3 Zadanie 3

## 3.1 Polecenie

Przygotować program realizujący ciekawe zapalanie/gaszenie diód podłączonych do portu P1.

## 3.2 Rozwiązanie

UWAGA! Kilka programów w jednym, linijka 56 służy za wybór trybu.

```
1  ljmp start
2
3    ; W tym programie zmienna r0 oraz r1 bedzie dzialac jako licznik
        petli
4    ; r3 bedzie zmienna do petli poza funkcjami
5    ; r4
6    ; r5 przechowanie tymczasowe
7    ; r7, r6przechowuja zapisane wartosci
8  org 050H
9    ; Opóznienie do wyswietlania (wziete z programu wstawionego na
        eportalu
10   delay:  mov r0, #0FFH
11   one:  mov r1, #0FFH
```

```
12    dwa:   djnz r1, dwa
13      djnz r0, one
14    ret
15
16    ; Przesuwaj diode w lewo 8 razy
17    wlewo:   mov r0, #08h; petla 8 razy
18    skok:    mov p1, a; przypisanie dla p1 wartosci a
19         rl a; przesuniecie a w lewo o jeden bit
20         ;delay; (skomentowane aby bylo latwiej testowac)
21         djnz r0, skok; skok oraz dekrementacja o 1 zmiennej r0
22    ret; koniec funkcji
23    ; Przesuwaj diode w prawo 8 razy
24    wprawo: mov r0, #08h; petla 8 razy
25    skok2:   rr a;przesuniecie a w prawo o jeden bit
26         mov p1, a; przypisanie dla p1 wartosci a
27         djnz r0, skok2; skok oraz dekrementacja o 1 zmiennej r0
28    ret; koniec funkcji
29    ; przesuwanie a w lewo
30    wlewosave:   mov r0, #08h; petla 8 razy
31    skok3:   mov r5, a;tymczasowe przechowanie wartosci a
32         orl a, r7; suma a oraz r7
33         mov p1, a; przypisanie p1 sumy wartosci a oraz r7
34         mov a, r5; przywrocenie wartosci a przed suma
35         rl a; przesuniecie a w lewo o jeden bit
36         ;delay; (skomentowane aby bylo latwiej testowac)
37         djnz r0, skok3; skok oraz dekrementacja o 1 zmiennej r0
38    ret; koniec funkcji
39    ; zsumowanie przesuniecia do swiecenia diody2
40    sumaprzesun:
41         mov a, r6; cykl przesuniecia r6
42         rr a; przesuniecie o 1 bit w prawo
43         mov r6, a; przywrocenie wartosci
44         orl a,r7; suma a oraz r7
45         mov r7, a; zapisanie sumy do r7
46         mov a, #01h; reset wartosci a
47         ret; koniec funkcji
48  org 0100h
49    start:   mov p1, #00h; reset wartosci p1
50         mov a, #01h; reset wartosci a
51
52         mov r6, #01h; reset wartosci r6
53         mov r7, #00h; reset wartosci r7
54
55
56         jmp diody2;wybór trybu swiecenia,do wyboru:
57         ; 1.diody1: przemieszczanie sie diody z lewej do prawej i
              spowrotem
58         ; 2.diody2: przemieszczanie sie wraz z ladowaniem od prawej
              strony
59         ; 3.diody3: pasek ladowania
60
```

```
61      ; Swiecenie tam i spowrotem
62      diody1: acall wlewo; wywloanie funkcji do przesuwania 8 razy
           wartosci w lewo
63           acall wprawo; wywloanie funkcji do przesuwania 8 razy
              wartosci w prawo
64           jmp diody1; ponowne wykonanie
65
66      ; Swiecenie wraz z ladowaniem z prawej strony
67      diody2:
68      acall wlewo; ; wywloanie funkcji do przesuwania 8 razy
           wartosci w lewo
69      mov r2, #07h; utworzenie petli 7-krotnej
70      petli:  acall sumaprzesun; suma przesuniecia
71           acall wlewosave; ; wywloanie funkcji do przesuwania 8 razy
                wartosci w lewo wraz z zapamietaniem
72           djnz r2, petli; ponowne wykonanie
73      mov r7, #00h; reset
74      mov r6, #01h; reset
75      mov a, #01h; reset
76      jmp diody2; ponowne wykonanie
77      ; Swiecenie jak pasek ladowania
78      diody3:
79      mov p1, #00h; wylaczenie diód
80      mov r2, #08h; petla 8-krotna
81      ;delay; (skomentowane aby bylo latwiej testowac)
82      mov r5, a; przypisanie zmiennej tymczasowej wartosci a
83      loadskok:   mov p1, a; wpisanie wartosci
84            rl a; przesuniecie wartosci
85            orl a, r6; uzupelnienie pustych bitow z tylu
86            mov r6, a
87            ;delay; (skomentowane aby bylo latwiej testowac)
88            djnz r2, loadskok; ponowne wykonanie
89      mov r6, #01h; reset
90      jmp diody3; ponowne wykonanie
91
92      ; standardowy koniec programu
93      nop
94      nop
95      nop
96      jmp $
97      end start
```