

Podstawy techniki mikroprocesorowej lab.

Prowadzący: Mgr inż. Antoni Sterna (E00-74ap, środa 13:15)

sprawozdanie 1 - 2018.03.21

Jakub Dorda 235013
Marcin Kotas 235098

16 maja 2018

L^AT_EX

1 Wprowadzenie/cel ćwiczeń

Celem ćwiczeń było wykonanie trzech programów. Pierwszy program kopiował dane z pamięci wewnętrznej do zewnętrznej, drugi sterował diodami, a trzeci kopiował dane z pamięci zewnętrznej do zewnętrznej.

2 Kod programu pierwszego

```
org 0
; test kopiowania

mov r1, #30h
mov r4, #5
mov dptr, #0030H
lcall copy

sjmp $

;-----
; Copy IRAM -> XRAM
; R0 - src DPTR - dest
;-----
copy:
    mov a, @r1
    movx @dptr, a
    inc dptr
    inc r1
    djnz r4, copy
    ret
end
```

Aby skopiować do zewnętrznej pamięci należy umieścić adres komórki w 16 bitowym rejestrze DPTR. Przy korzystaniu z tego rejestru należy użyć komendy movx zamiast mov.

3 Kod programu drugiego

```
org 0
```

```
mov a, #254
```

```
mov r4, #7
```

```
blinkL:
```

```
    mov r3, #50
```

```
    call delay
```

```
    mov p1, a
```

```
    rl a
```

```
    djnz r4, blinkL
```

```
mov r4, #7
```

```
blinkR:
```

```
    mov r3, #50
```

```
    call delay
```

```
    mov p1, a
```

```
    rr a
```

```
    djnz r4, blinkR
```

```
mov r4, #7
```

```
jmp blinkL
```

```
;-----
```

```
; DELAY: r3 * 10ms
```

```
;-----
```

```
delay:
```

```
    mov r1, #10          ; 10*1ms
```

```
d1:
```

```
    mov r0, #250         ; 250*4us=1ms
```

```
d0:
```

```
    nop                  ; 1
```

```
    nop                  ; 1
```

```
    djnz r0, d0          ; 2
```

```
;-----
```

```
; 4
```

```
    djnz r1, d1
```

```
    djnz r3, delay
```

```
ret
```

```
end
```

Drugi program wykorzystuje funkcję nop (no operation) aby wymusić opóźnienie wykonywania programu. Wykonanie pętli d0 trwa w przybliżeniu 4 us. Poprzez dodanie dodatkowych liczników w r0 i r1, pętla delay wykonuje się $10\text{ms} \cdot r3$. W naszym przypadku w r3 znajduje się 50, więc pętla wykonuje się pół sekundy. Sterowanie diodami odbywa się za pomocą rejestru p1. Dla 0 dioda się świeci, dla 1 nie. Efekt przesuwającej się diody uzyskany został przez użycie funkcji rr oraz rl, które przesuwają odpowiednio w prawo oraz w lewo bity znajdujące się w akumulatorze.

4 Kod programu trzeciego

```
org 0
    mov r0 , #00h
    mov r1 , #30h
    mov r2 , #00h
    mov r3 , #38h
    mov r4 , #5

    mov dph , r0
    mov dpl , r1
    mov r4 , #5
    lcall copy
    sjmp $

copy:

    mov dph , r0
    mov dpl , r1
    movx a , @dptr
    inc dptr
    mov r0 , dph
    mov r1 , dpl

    mov dph , r2
    mov dpl , r3
    movx @dptr , a
    inc dptr
    mov r2 , dph
    mov r3 , dpl

    djnz r4 , copy
    ret
end
```

Aby program mógł kopiować dane z pamięci zewnętrznej do zewnętrznej, należy użyć dwa razy rejestru DPTR. W tym celu na początku pętli do DPTR wstawiany jest adres z rejestrów r0 i r1, z którego mają zostać skopiowane dane. Wtedy odbywa się kopiowanie do akumulatora zawartości pamięci pod adresem wskazywanym przez DPTR oraz zwiększenie adresu DPTR. Następnie do rejestrów r0 i r1 odstawiany jest aktualny adres DPTR oraz wstawiany nowy adres zachowany w rejestrach r2 i r3. Dopiero wtedy kopiuje zawartość akumulatora pod nowy adres wskazywany przez DPTR, oraz zwiększa adres DPTR. Na końcu pętli adres wskazywany przez dptr jest wstawiany z powrotem do rejestrów r2 i r3.

5 Wnioski/podsumowanie

Wszystkie programy uruchomiły się poprawnie.