

APLICATII

1. Calcularea rutinelor de întârziere

a) Exemplu de rutină de întârziere:

```
CALL DELAY1           ;apelez rutina de întârziere (în 2μs)

DELAY1:
    MOV R0,#46        ;încarc registrul R0 cu constanta 46 zecimal (în 2μs)
DELAY10:
    DJNZ R0,DELAY10   ;decrementez R0. Dacă R0 nu este egal cu zero, sari la DELAY1 (în 46*2μs)
    RET               ;întoarcere din apelare (în 2μs)
```

Să facem un simplu calcul matematic:

$$2\mu s + 2\mu s + 46 * 2\mu s + 2\mu s = 98\mu s$$

b) Exemplu de rutină de întârziere:

```
CALL DELAY2           ;apelez rutina de întârziere (în 2μs)

DELAY2:
    MOV R0,#8         ;încarc registrul R0 cu, constanta 8 zecimal (în 2μs)
    MOV R1,#196       ;încarc registrul R1 cu, constanta 196 zecimal (în 2μs)
DELAY20:
    DJNZ R1,DELAY20   ; decrementez R1. Dacă R1 nu este egal cu zero, sari la DELAY20
    DJNZ R0,DELAY20   ;decrementez R0. Dacă R0 nu este egal cu zero, sari la DELAY20
    RET               ;întoarcere din apelare (în 2μs)
```

Să facem un simplu calcul matematic:

$$2\mu s + 2\mu s + 2\mu s + 196 * 2\mu s + 256 * 2\mu s * (8-1) + 8 * 2\mu s + 2\mu s = 4000\mu s = 4ms$$

c) Exemplu de rutină de întârziere:

Să presupunem că ne dorim calcularea unei rutine de întârziere de 100μs. Din primul moment, îmi voi scădea cele 2μs – corespunzător instr. CALL și 2μs – corespunzător instr. RET.

$$100\mu s - 2\mu s - 2\mu s = 96\mu s$$

Timpul rămas, îl împart la 2 (instr. DJNZ este pe 2μs).

$$96\mu s / 2 = 48\mu s$$

În mod normal, ar trebui să încarc registrul ales (de exp. R0), cu valoarea 48 (am lucrat NUMAI în zecimal); dar să nu uităm că și timpul de încărcare al Registrului ales, durează și el 2μs (exp. MOV R0,#xxx) → atunci voi face în felul următor: $(96\mu s - 2\mu s) / 2 = 47$. Abia această valoare o încarc în Registrul ales.

Intuitiv, puteam din start să scad cele 2μs necesare execuției (exp. MOV R0,#xxx), dar acest lucru este valabil doar la Rutine de întârziere mici (ce folosesc doar un singur Registru de lucru. exp. R0, R1, R2, ..., un octet din memoria de date (exp. 40H), etc.).

```
CALL DELAY1           ;apelez rutina de întârziere (în 2μs)

DELAY1:
    MOV R0,#47        ;încarc registrul R0 cu constanta 47 zecimal (în 2μs)
DELAY10:
    DJNZ R0,DELAY10   ;decrementez R0. Dacă R0 nu este egal cu zero, sari la DELAY1 (în 47*2μs)
    RET               ;întoarcere din apelare (în 2μs)
```

d) Exemplu de rutină de întârziere:

Să presupunem că ne dorim calcularea unei rutine de întârziere de 10ms. Din primul moment, îmi voi scădea cele $2\mu s$ – corespunzător instr. CALL și $2\mu s$ – corespunzător instr. RET.

$$10ms - 2\mu s - 2\mu s = 9.996\mu s$$

Dacă împart 9.996 la 2 rezultă 4.998, valoare ce nu pot să o încarc într-un octet (val. max. este 255). În această situație mai am nevoie de un alt registru. Atunci:

$$9.996 / 256 = 39,03$$

Această nouă valoare este împărțită la 2 (deoarece instr. DJNZ este executată în $2\mu s$).

$$39,03 / 2 = 19,5$$

Deoarece 19,5 nu este nr. întreg ; nu poate fi memorat într-un registru. Voi trunchia această valoare, obțin 19.

Reiau calculul cu noua valoare:

$$19 * 2(\mu s) = 38 - \text{se înmulțește cu 2 deoarece DJNZ se execută în } 2\mu s.$$

$$38 * 256 = 9.728(\mu s) - \text{această valoare se obține cu cel de-al doilea registru.}$$

Cu ce valoare încarc primul registru ?

$$9.996 - 9.728 = 268$$

Din această valoare scad timpul necesar încărcării celor două registre:

$$268 - 2 - 2 = 264$$

Tot din acest timp, mai scad timpul de execuție al instr. DJNZ de la celălalt registru:

$$264 - (19 + 1) * 2\mu s = 224\mu s - \text{se înmulțește cu 2 deoarece DJNZ se execută în } 2\mu s.$$

IMPORTANT: observați apariția în paranteză a cifrei 1!

Abia în acest moment obțin valoarea de încărcare pt. primul registru:

$$224 / 2\mu s = 112$$

CALL DELAY2 ;apelez rutina de întârziere (în $2\mu s$)

DELAY2:

MOV R2,#20 ;încarc registrul R0 cu, constanta 20 zecimal (în $2\mu s$)

MOV R1,#112 ;încarc registrul R1 cu, constanta 112 zecimal (în $2\mu s$)

DELAY20:

DJNZ R1,DELAY20 ; decrementez R1. Dacă R1 nu este egal cu zero, sari la DELAY20

DJNZ R2,DELAY20 ;decrementez R0. Dacă R0 nu este egal cu zero, sari la DELAY20

RET

Observați că Reg. R2 este încărcat cu valoarea 20 , și nu cu 19, așa cum ne-am fi așteptat; apare 1 în plus deoarece mai întâi este decrementat R1, și apoi de 19 ori se execută: $(19 * 2 \mu s) * 256$

d) Exemplu de rutină de întârziere:

Să presupunem că ne dorim calcularea unei rutine de întârziere de 20ms. Din primul moment, îmi voi scădea cele $2\mu s$ – corespunzător instr. CALL și $2\mu s$ – corespunzător instr. RET.

$$20ms - 2\mu s - 2\mu s = 19.996\mu s$$

$$19,996 / 256 = 78,109$$

$$78,109 / 2(\mu s) = 39,054$$

Trunchiez 39,054, rezultă: 39.

Recalculez cu noua valoare:

$$(39 * 2\mu s) * 256 = 19.968 \mu s$$

Cu cât încarc primul registru?

$$19,996 \mu s - 19.968 \mu s = 28 \mu s$$

$$28 \mu s - 2 \mu s - 2 \mu s = 24 \mu s \text{ (am scăzut timpul de încărcare al celor două registre)}$$

$$24\mu s - (39 * 2\mu s) = -54 \mu s$$

Nu este bine.

Am să scad o unitate de la registrul al doilea: $39 - 1 = 38$. Recalculez:

$$(38 * 2\mu s) * 256 = 19.456 \mu s$$

Cu cât încarc primul registru?

$$19,996 \mu s - 19.456 \mu s = 540 \mu s$$

$$540 \mu s - 2 \mu s - 2 \mu s = 536 \mu s \text{ (am scăzut timpul de încărcare al celor două registre)}$$

$$536\mu s - (38 + 1) * 2\mu s = 458\mu s$$

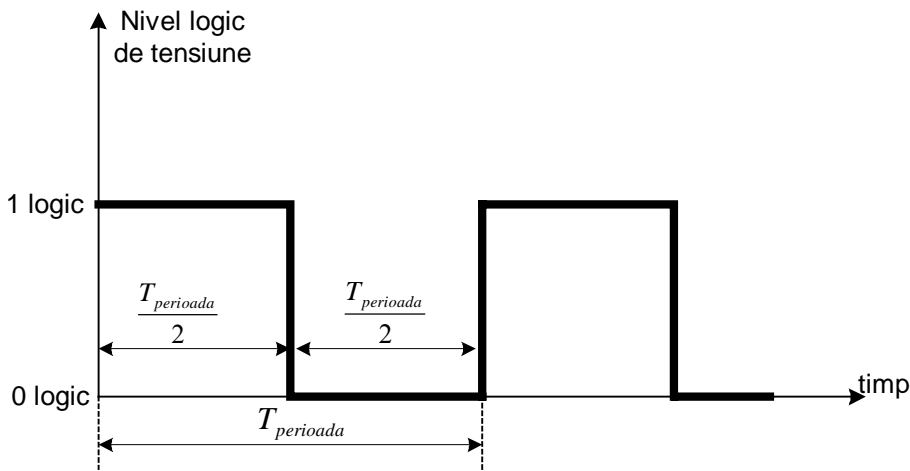
Obțin valoarea de încărcare pentru primul registru:

$$458 \mu s / 2 \mu s = 229$$

```
CALL DELAY2                ;apelez rutina de întârziere (în 2μs)

DELAY2:
MOV R2,#39                 ;încarc registrul R0 cu, constanta 39 zecimal (în 2μs)
MOV R1,#229                ;încarc registrul R1 cu, constanta 229 zecimal (în 2μs)
DELAY20:
DJNZ R1,DELAY20            ;decrementez R1. Dacă R1 nu este egal cu zero, sari la DELAY20
DJNZ R2,DELAY20            ;decrementez R0. Dacă R0 nu este egal cu zero, sari la DELAY20
RET
```

2. Program: Generarea unui semnal sonor (cu frecvența de 1KHz)



Să presupunem că vrem generarea unui semnal sonor cu frecvența de 1KHz. Va trebui mai întâi să calculăm perioada semnalului.

$$T_{perioada} = \frac{1}{frecvanta} = \frac{1}{1KHz} = 10^{-3} \text{ secunde} = 1ms$$

Vom împărți această Perioadă în doua segmente de timp egale, astfel încât, la un bit dintr-un port să am nivel de tensiune 1 logic = 0,5ms și nivel de tensiune 0 logic = 0,5ms; apoi iar să am nivel de tensiune 1 logic = 0,5ms și iar nivel de tensiune 0 logic = 0,5ms, etc.

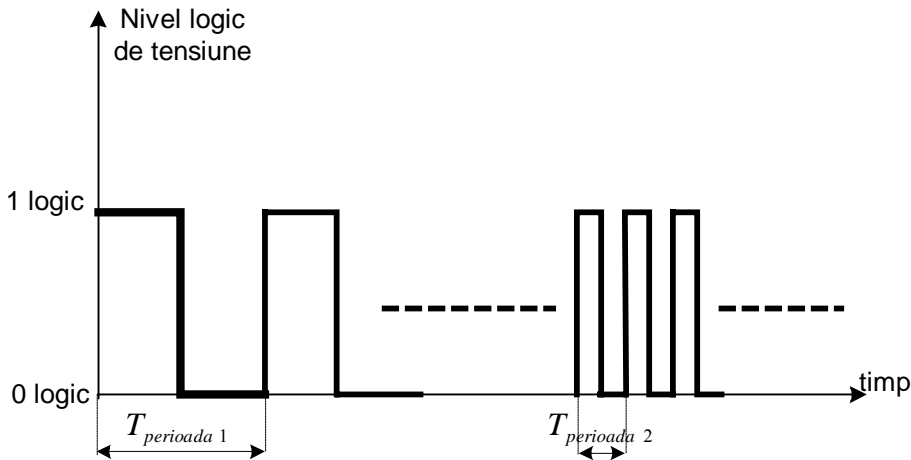
Programul:

```
name ton                    ;nume program
ORG 0000H                  ;INCEPE PROGRAMUL DE LA ADRESA 0000H
BUZZ EQU P3.0              ;ETCHETEAZA PINUL DE PORT P3.0
START:
MOV P3,#00H                ;STINGE TOATE LED-urile CONECTATE LA PORTUL P3 - daca am asa ceva...
SETB BUZZ                  ;PUNE IN 1 LOGIC BITUL P3.0
CALL DELAY                  ;APELEAZA O INTARZIERE DE 500 MICROSECUNDE
CLR BUZZ                    ;PUNE IN 0 LOGIC BITUL P3.0
CALL DELAY                  ;APELEAZA O INTARZIERE DE 500 MICROSECUNDE
JMP START                  ;SARI LA START SI REPETA

DELAY:                      ;RUTINA DE INTARZIERE DE 500 uS
MOV R0,#246                 ;INCARCA REGISTRUL R0 CU CONSTANTA DIRECTA 246 ZECIMAL

DELAY1:
DJNZ R0,DELAY1              ;DECREMENTEAZA R0 PANA CAND ACESTA ESTE EGAL CU ZERO
NOP                          ;PIERDE TIMPUL 1uS
NOP                          ;PIERDE TIMPUL 1uS
RET                          ;INTOARCERE DIN SUBRUTINA DE INTARZIERE
END                          ;SFARSIT DE PROGRAM
```

Program: Generarea unui semnal sonor cu două tonuri



Să presupun că vreau ca cele două frecvențe să fie de $F1 = 400\text{Hz}$ și $F2 = 2\text{KHz}$, ceea ce corespunde unor perioade de $T_{perioada\ 1} = 2,5\text{ms}$ și $T_{perioada\ 2} = 0,5\text{ms}$. Perioada unui semnal este formată din doua semialternanțe ce corespunde celor doua nivele logice de zero și de 1 logic. Pentru generarea semnalului de frecventa 400Hz, va trebui ca să „stau” pe 1 logic $2,5\text{ms}/2$ și pe zero logic $2,5\text{ms}/2$. Repet această perioadă (ce îmi dă frecvența semnalului meu de 400Hz) de cate ori vreau eu, iar dacă vreau să am semnal de această frecvență timp de 1 secundă, va trebui să repet această perioadă de: $1000\text{ms} / 2,5\text{ms} = 400$ ori. La fel se întâmplă și cu semnalul de frecvență 2KHz. Repet perioada de $0,5\text{ms}$, într-o secundă de: $1000\text{ms} / 0,5\text{ms} = 2000$ ori.

Programul:

```

name biton
ORG 0000H
BUZZ EQU P3.0
START:
    MOV R0,#144    ;STABILESC PRIN R0 SI R1 REPETAREA DE 400 ORI A LUI F1
    MOV R1,#2      ;144 + 256*1 = 400
    MOV R2,#208    ;STABILESC PRIN R2 SI R3 REPETAREA DE 2000 ORI A LUI F2
    MOV R3,#8      ;208 + 256*7 = 2000

F1:    SETB BUZZ    ;APLIC 1 LOGIC LA BUZZ
        CALL DELAY1 ;ASTEPT 1,25 MILISECUNDE
        CLR BUZZ   ;APLIC 0 LOGIC LA BUZZ
        CALL DELAY1 ;ASTEPT 1,25 MILISECUNDE
        DJNZ R0,F1 ;REPET CELE DOUA SEMIPERIOADE (1,25ms + 1,25ms =2,5ms) CE-MI DA O
        ;FRECVENTA
        DJNZ R1,F1 ;A SEMNALULUI AUDIO DE 400 Hz, DE 400 ORI , ADICA 400*2,5ms=1secunda

F2:    SETB BUZZ    ;APLIC 1 LOGIC LA BUZZ
        CALL DELAY2 ;ASTEPT 0,25 MILISECUNDE
        CLR BUZZ   ;APLIC 0 LOGIC LA BUZZ
        CALL DELAY2 ;ASTEPT 0,25 MILISECUNDE
        DJNZ R2,F2 ;REPET CELE DOUA SEMIPERIOADE (0,25ms + 0,25ms =0,5ms) CE-MI DA O
        ;FRECVENTA
        DJNZ R3,F2 ;A SEMNALULUI AUDIO DE 2 KHz, DE 2000 ORI , ADICA
        ;2000*0,5ms=1secunda
        JMP START  ;SARI LA START

DELAY1: ;RUTINA DE INTARZIERE DE 1,25 MILISECUNDE
    
```

```
MOV R4,#3
MOV R5,#106
DELAY10:
  DJNZ R5,DELAY10
  DJNZ R4,DELAY10
  RET

DELAY2:                                ;RUTINA DE INTARZIERE DE 0,25 MILISECUNDE
  MOV R6,#121
DELAY20:
  DJNZ R6,DELAY20
  RET

END
```