

OPERATIVNI SISTEMI

ZADACA 3 PRIMJERAK 5

STUDENT: Mujkić Daris 19413 RI-2b

Potražiti podatke o uređaju PC Speaker na PC računarima i navesti koje ulazno/izlazne portove koristi taj uređaj te na koji način se pristupa podacima ili statusnim registrima na niskom nivou (primjer u C ili assembleru).

Funkcionalnost: PC Speaker je zvučni uređaj koji je bio ugrađen u mnoge stare računare (PC-eve) i koristio se za emitiranje jednostavnih zvučnih signala kao što su beepovi, zvukovi upozorenja i melodije. Na starijim PC računarima, PC Speaker je bio jedini dostupni izvor zvuka prije nego što su se razvili moderni zvučni sistemi.

Povezivanje: Ugrađen je na matične ploče starijih računara, a povezan je s računarom preko 4-pinskog konektora.

Registri i portovi: PC Speaker koristi jedan I/O port, a to je port 0x61 (0061). Taj port se koristi za kontrolu PC zvučnika. Za emitiranje zvuka, bitovi 0-5 registra 0x61 se postavljaju na odgovarajuće vrijednosti (bitovi su se koristili za podešavanje jačine zvuka), a zaustavljanje zvuka se postiže resetovanjem tih bitova na 0. Frekvencija se mogla podešavati pomoću brojčanih vrijednosti koje su se pisale na Portove 0x43 (0043, upisivanje konfiguracionih informacija i odabir odgovarajućeg kanala) i 0x42 (0042, upisivanje vrijednosti niskih i visokih bajtova frekvencije).

Pristup podacima i statusnim registrima: Na niskom nivou, pristupanje podacima i statusnim registrima PC Speakera se ostvaruje upisivanjem i čitanjem vrijednosti iz portova. U C programskom jeziku se npr. Koriste funkcije `outp()` i `inp()` za pisanje i čitanje iz tih portova. U assembleru se koriste odgovarajuće instrukcije za upis i čitanje (`mov al, out` itd.).

Moderni računari: Moderni računari najčešće koriste zvučne kartice za reprodukciju kvalitetnog zvuka. Dakle, PC Speaker nije zastupljen u novijim računarima.

Primjer u programskom jeziku C za emitiranje zvuka na PC Speakeru može izgledati ovako:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <dos.h>
```

```
#define PORT 0x61
```

```
void beep(int duration, int frequency) {
```

```
    int i;
```

```
    int divisor = 1193180 / frequency;
```

```

    outp(PORT, inp(PORT) | 3); // uključi zvučnik

    for (i = 0; i < duration; i++) {
        outp(0x43, 0xb6);
        outp(0x42, divisor & 0xff);
        outp(0x42, divisor >> 8);

        delay(1); // kratka pauza

        outp(PORT, inp(PORT) & ~3); // isključi zvučnik
        delay(1); // kratka pauza
    }
}

int main() {
    beep(500, 440); // emitiraj ton 440Hz, trajanja 500ms
    return 0;
}

```

Ovaj program koristi funkciju **outp()** iz zaglavlja **<dos.h>** za pisanje u I/O portove. Bitovi 0 i 1 registra 0x61 se koriste za uključivanje i isključivanje zvučnika. Funkcija **beep()** (prima dva argumenta: **duration** (trajanje zvuka u milisekundama) i **frequency** (frekvencija zvuka u Hz)) računa vrijednost djelioca (divisor) potrebnog za generisanje željene frekvencije, te koristi registre 0x42 i 0x43 za slanje podataka u timer čip. Emitiranje zvuka se postiže uključivanjem zvučnika na početku, te isključivanjem na kraju svakog ciklusa tona. Pauza od 1ms između tonova se postiže pozivom **delay()** funkcije koja je također dio **<dos.h>** zaglavlja. U main funkciji je dat jedan primjer korištenja funkcije beep sa tonom od 440Hz i trajanjem od 500ms.

Evo primjera u kojem koristimo assembler:

; Postavljanje bita 0x02 na portu 0x61 kako bismo omogućili reprodukciju zvuka

mov al, 0x02

out 0x61, al

; Pauziranje izvršavanja programa neko vrijeme za emitiranje zvuka

mov cx, 1000

delay_loop:

loop delay_loop

; Postavljanje bita 0x00 na portu 0x61 kako bismo zaustavili emitiranje zvuka

mov al, 0x00

out 0x61, al

U ovom primjeru, prvih nekoliko linija koda postavlja bit 0x02 na portu 0x61 kako bi se omogućila reprodukcija zvuka. Korišten je registar **AL** za smeštanje vrijednosti 0x02, a zatim se ta vrijednost iz registra AL šalje na port 0x61 pomoću instrukcije **out**.

Nakon toga se izvršava pauza. U ovom primjeru, vrednost 1000 je smeštena u registar **CX**, a zatim se koristi petlja **loop** za ponavljanje izvršavanja sljedećeg bloka koda (u ovom slučaju, **delay_loop**) 1000 puta. Ovo pravi pauzu i omogućava reprodukciju zvuka tokom određenog vremenskog perioda.

Na kraju, kod postavlja bit 0x00 na portu 0x61 kako bi se zaustavila reprodukcija zvuka. Slično kao i prije, vrijednost 0x00 je smještena u registar AL, a zatim se ta vrijednost šalje na port 0x61 pomoću instrukcije **out**.

Ukratko, ovaj kod omogućava reprodukciju zvuka postavljanjem odgovarajućeg bita na portu 0x61, pravi pauzu za reprodukciju zvuka i zatim zaustavlja reprodukciju postavljanjem drugog bita na istom portu.

BONUS ČINJENICE:

PC Speaker se sastoji od više komponenti, a neke od njih su:

- Piezo zvučnik: fizički zvučnik koji generiše zvuk pomoću piezo-električnog materijala (razlog zašto je PC Speaker osuđen na emitiranje samo jednostavnih zvukova)

- Kontrolni sklop: sadrži logiku i prekidače koji kontrolišu uključivanje i isključivanje zvučnika i podešavanje jačine zvuka