**آزمایش 10**

**بازی ریاضی**

**1. زمان مسابقه را با استفاده از کدامیک از میکروکنترلرها شمارش کردید؟**

در میکرو B با استفاده از timer1 ؛ زیرا برای این کار مناسب‌تر است. مقدار تایمر بر روی 100 میلی ثانیه تنظیم شده است. زمان پیشفرض بازی نیز 20 سایکل است.

**2. برای تولید سوالات چگونه عمل کردید؟**

در میکرو A دو تابع rand\_num و rand\_op تعریف شده است که برای ما عملوندها و عملگرهای تصادفی تولید می‌کنند. در نهایت از این دو تابع در تابع gen\_exp استفاده شده است تا سؤالات تصادفی تولید شود. این تابع پاسخ صحیح عبارت تولید شده را ریترن می‌کند.

در تابع generate نیز ابتدا با استفاده از gen\_exp یک سؤال تولید می‌شود و با استفاده از getInput منتظر وارد کردن پاسخ کاربر می‌ماند. پس از دریافت پاسخ و مقایسه با پاسخ صحیح، سیگنال‌های لازم جهت اعمال امتیاز را به میکرو B ارسال می‌کند.

**3. کدام INT‌ها در میکروکنترلرها فعال هستند و هر کدام چه زمانی و توسط کدام میکرو فعال می‌شوند؟**

در هر دو میکرو، وقفه‌های صفر و یک فعال هستند.

وقفه صفر در میکرو A : زمانی که شمارش تایمر به پایان رسید، میکرو B با ارسال سیگنال، آن را فراخوانی می‌کند و موجب clear شدن lcd و متوقف شدن فعالیت میکرو A می‌شود.

وقفه یک در میکرو A : زمانی که کاربر دکمه on را در کیپد سمت راست فشار داد، میکرو B سیگنالی به آن ارسال می‌کند و فعالیت میکرو A جهت تولید سؤالات، آغاز می‌شود.

وقفه صفر در میکرو B برای کاهش امتیاز و وقفه یک برای افزایش امتیاز استفاده شده است و میکرو A پس از دریافت پاسخ کاربر و مقایسه با پاسخ صحیح، با ارسال سیگنال‌های مناسب آن‌ها را فراخوانی می‌کند.

**4. کد برنامه**

*// Micro A*

*#include* <mega32.h>

*#include* <alcd.h>

*#include* <delay.h>

*#include* <stdio.h>

*#include* <stdlib.h>

*#define* SIGN\_NONE 255

*#define* SIGN\_DIVISION 10

*#define* SIGN\_MULTIPLICATION 11

*#define* SIGN\_MINUS 12

*#define* SIGN\_ON 13

*#define* SIGN\_EQUALS 14

*#define* SIGN\_PLUS 15

char isActive = 0;

char getKey() {

    char k = SIGN\_NONE;

    DDRA = 0b11110000;

    PORTA = 0b11101111;

*while* (PINA.0 == 0) k = 7;

*while* (PINA.1 == 0) k = 8;

*while* (PINA.2 == 0) k = 9;

*while* (PINA.3 == 0) k = SIGN\_DIVISION;

    PORTA = 0b11011111;

*while* (PINA.0 == 0) k = 4;

*while* (PINA.1 == 0) k = 5;

*while* (PINA.2 == 0) k = 6;

*while* (PINA.3 == 0) k = SIGN\_MULTIPLICATION;

    PORTA = 0b10111111;

*while* (PINA.0 == 0) k = 1;

*while* (PINA.1 == 0) k = 2;

*while* (PINA.2 == 0) k = 3;

*while* (PINA.3 == 0) k = SIGN\_MINUS;

    PORTA = 0b01111111;

*while* (PINA.0 == 0) k = SIGN\_ON;

*while* (PINA.1 == 0) k = 0;

*while* (PINA.2 == 0) k = SIGN\_EQUALS;

*while* (PINA.3 == 0) k = SIGN\_PLUS;

*return* k;

}

void signal\_0() {

    PORTD.0 = 1;

    delay\_ms(1);

    PORTD.0 = 0;

}

void signal\_1() {

    PORTD.1 = 1;

    delay\_ms(1);

    PORTD.1 = 0;

}

void lcds(flash unsigned char \*format, ...) {

    char buffer[128];

    va\_list va;

    va\_start(va, format);

    vsprintf(buffer, format, va);

    va\_end(va);

    lcd\_clear();

    lcd\_puts(buffer);

}

void lcda(flash unsigned char \*format, ...) {

    char buffer[128];

    va\_list va;

    va\_start(va, format);

    vsprintf(buffer, format, va);

    va\_end(va);

    lcd\_puts(buffer);

}

int rand\_num(char digitsCount) {

    int t = 10;

    int i = 1;

*while* (i < digitsCount) {

        t \*= 10;

        i++;

    }

*return* rand() % t;

}

char rand\_op() {

    char t = rand() % 4;

*if* (t == 0) *return* '+';

*if* (t == 1) *return* '-';

*if* (t == 2) *return* '\*';

*if* (t == 3) *return* '/';

}

int getInput() {

    char digits[10];

    char i = 0;

    char j = 0;

    int num = 0;

*while* (1) {

        char k = getKey();

*if* (!isActive) *return* 0;

*if* (k == SIGN\_EQUALS) {

*for* (j = 0; j < i; j++)

                num = num \* 10 + digits[j];

*return* num;

        }

*if* (k > 9) *continue*;

        lcda("%d", k);

        digits[i++] = k;

    }

}

int eval(int num1, int num2, char op) {

*if* (op == '/' && num2 == 0) *return* 0;

*if* (op == '+') *return* num1 + num2;

*if* (op == '-') *return* num1 - num2;

*if* (op == '\*') *return* num1 \* num2;

*if* (op == '/') *return* num1 / num2;

}

int gen\_exp() {

    int num1 = rand\_num(2);

    int num2 = rand\_num(1);

    char op = rand\_op();

    int ans = eval(num1, num2, op);

    lcds("%d %c %d = ", num1, op, num2);

*return* ans;

}

void generate() {

    int ans, input;

*if* (!isActive) *return*;

    ans = gen\_exp();

    input = getInput();

*if* (!isActive) *return*;

*if* (input == ans) signal\_1();

*else* signal\_0();

}

*// External Interrupt 0 service routine*

interrupt [EXT\_INT0] void ext\_int0\_isr(void) {

    isActive = 0;

    lcd\_clear();

}

*// External Interrupt 1 service routine*

interrupt [EXT\_INT1] void ext\_int1\_isr(void) {

    isActive = 1;

}

void main(void) {

*while* (1) {

        generate();

    }

}

*// Micro B*

*#include* <mega32.h>

*#include* <alcd.h>

*#include* <delay.h>

*#include* <stdio.h>

*#include* <stdlib.h>

*#define* SIGN\_NONE 255

*#define* SIGN\_DIVISION 10

*#define* SIGN\_MULTIPLICATION 11

*#define* SIGN\_MINUS 12

*#define* SIGN\_ON 13

*#define* SIGN\_EQUALS 14

*#define* SIGN\_PLUS 15

int time = 20;

int timeCnt = 0;

int score = 0;

char isActive = 0;

char getKey() {

    char k = SIGN\_NONE;

    DDRA = 0b11110000;

    PORTA = 0b11101111;

*while* (PINA.0 == 0) k = 7;

*while* (PINA.1 == 0) k = 8;

*while* (PINA.2 == 0) k = 9;

*while* (PINA.3 == 0) k = SIGN\_DIVISION;

    PORTA = 0b11011111;

*while* (PINA.0 == 0) k = 4;

*while* (PINA.1 == 0) k = 5;

*while* (PINA.2 == 0) k = 6;

*while* (PINA.3 == 0) k = SIGN\_MULTIPLICATION;

    PORTA = 0b10111111;

*while* (PINA.0 == 0) k = 1;

*while* (PINA.1 == 0) k = 2;

*while* (PINA.2 == 0) k = 3;

*while* (PINA.3 == 0) k = SIGN\_MINUS;

    PORTA = 0b01111111;

*while* (PINA.0 == 0) k = SIGN\_ON;

*while* (PINA.1 == 0) k = 0;

*while* (PINA.2 == 0) k = SIGN\_EQUALS;

*while* (PINA.3 == 0) k = SIGN\_PLUS;

*return* k;

}

void signal\_0() {

    PORTD.0 = 1;

    delay\_ms(1);

    PORTD.0 = 0;

}

void signal\_1() {

    PORTD.1 = 1;

    delay\_ms(1);

    PORTD.1 = 0;

}

void lcds(flash unsigned char \*format, ...) {

    char buffer[128];

    va\_list va;

    va\_start(va, format);

    vsprintf(buffer, format, va);

    va\_end(va);

    lcd\_clear();

    lcd\_puts(buffer);

}

void lcda(flash unsigned char \*format, ...) {

    char buffer[128];

    va\_list va;

    va\_start(va, format);

    vsprintf(buffer, format, va);

    va\_end(va);

    lcd\_puts(buffer);

}

void setScore(int n) {

    score = n;

    lcds("Score: %d", n);

}

void getOnKey() {

    char k = getKey();

*if* (k != SIGN\_ON) *return*;

*if* (isActive) *return*;

    setScore(0);

    isActive = 1;

    signal\_1();

}

*// External Interrupt 0 service routine*

interrupt [EXT\_INT0] void ext\_int0\_isr(void) {

    setScore(score - 1);

}

*// External Interrupt 1 service routine*

interrupt [EXT\_INT1] void ext\_int1\_isr(void) {

    setScore(score + 3);

}

*// Timer1 overflow interrupt service routine*

interrupt [TIM1\_OVF] void timer1\_ovf\_isr(void) {

*// Reinitialize Timer1 value*

    TCNT1H=0xBC80 >> 8;

    TCNT1L=0xBC80 & 0xff;

*// Place your code here*

*if* (!isActive) *return*;

    timeCnt = (timeCnt + 1) % time;

*if* (timeCnt == 0) {

        lcds("Your Final Score is %d", score);

        signal\_0();

        isActive = 0;

    }

}

void main(void) {

    lcds("Press ON/C to start");

*while* (1) {

        getOnKey();

    }

}