

NUME _____

DATA _____

PRENUME _____

GRUPA _____

TEST SOC – NR. 2

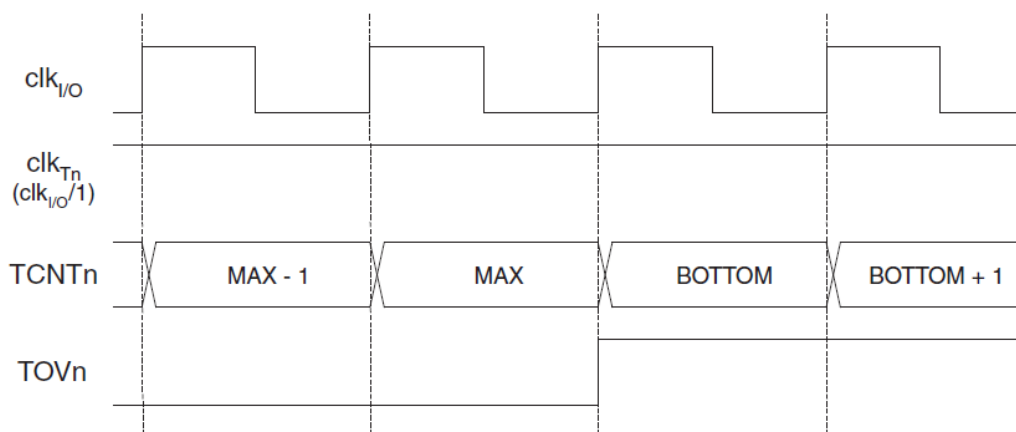
	A	B	C	D
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11	4 ms			
12	~ 2 ori/secundă			
13	A028			
14	2.083 ms			
15	20.66 ori/secundă			

Obs. Considerați frecvența la care operează microcontrolerul de 4MHz (acolo unde este cazul!).

1. Care este valoarea registrului UBRR pentru a seta viteza de comunicație pe interfața serială la 19200 baud (biți pe secundă) ? Microcontrolerul lucrează la 4 MHz cu U2X = 1.
 - a. 47
 - b. 11
 - c. 25
 - d. 23
2. Ce cantitate de memorie de program are microcontrolerul ATmega16?
 - a. 8 K
 - b. 1 K
 - c. 16 K
 - d. 32 K
3. Care este numărul registrelor interne de uz general al microcontrolerului ATmega16?
 - a. 32
 - b. 16
 - c. 8
 - d. 64
4. Care este frecvența de lucru maximă pentru microcontrolerul ATmega16?
 - a. 4 MHz
 - b. 16 MHz
 - c. 1 MHz
 - d. 10 MHz

5. În figura este prezentată diagrama de timp pentru perifericul Timer/Counter0 în următorul mod de lucru:

- Normal, prescaler /8
- Clear Timer on Compare Match (CTC), prescaler /8
- Normal, fara prescaler
- Clear Timer on Compare Match (CTC), fara prescaler



- Câte întreruperi (de depășire) sunt generate de timerul 1 al microcontrolerului ATmega16 în 5 secunde, dacă acesta funcționează în mod normal cu un prescaler de 8?
 - 38
 - 13
 - 84
 - 28
- Care este durata de timp măsurată de timerul 0 al microcontrolerului ATmega16 între valorile 100 și 210 ale registrului TCNT0, dacă acesta funcționează în mod normal cu un prescaler de 8?
 - 120 us
 - 210 us
 - 110 us
 - 220 us
- Care este frecvența reală de lucru a microcontrolerului ATmega16 pentru care instrucțiunea `__delay_cycles(200000L)` se execută în 44 ms ?
 - 4.16 MHz
 - 4.54 MHz
 - 4.45 MHz
 - 4.35 MHz
- Care este durata impulsului pozitiv al unui semnal dreptunghiular generat cu ajutorul timerului 1 al microcontrolerului ATmega16 care funcționează în modul 14 cu $ICR1 = 999$ și $OCR1A = 500$? Timerul folosește prescaler de 64 iar la începutul ciclului de numărare, pinul pe care este generat semnalul este 1 logic.
 - 10 ms
 - 8 ms
 - 1 ms
 - 500 us

10. Care este valoarea de pe pinul PD2 după execuția următoarei secvențe de instrucțiuni la începutul unui program?

```
PORTD      = 0xE6;
DDRD       = 0xB5;
```

- a. Nedefinit
- b. Impedanță ridicată (Hi-Z)
- c. 1 (5V)
- d. 0 (0V)

11. Care este perioada de timp calculată de următoarea expresie în care TCNT1 are valoarea 16000? Timerul 1 funcționează fără prescaler. Precizați în mod deosebit unitatea de măsură!

```
unsigned long period = (TCNT1 >> 2) / 1000;
```

12. De câte ori se va executa în fiecare secundă zona de cod marcată cu gri?

```
unsigned int count = 0;
unsigned int flag = 0;
unsigned int count = 0;
#pragma vector = TIMER1_OVF_vect
__interrupt void isr_TIMER1_overflow(void) {
    count++;
    if (count == 100) {
        count = 0;
        flag = 1;
    }
    TCNT1 = 45500;
}
void main(void) {
    TCNT1      = 45500;
    TCCR1B     |= (1<<CS10);
    TIMSK      |= (1<<TOIE1);
    __enable_interrupt();
    while (5>4) {
        if (flag == 1) {
            flag = 0;
        }
    }
}
```

13. Care este șirul de caractere transmis pe interfața serială la apelul funcției print de mai jos?

```
int value = 40;
print(value+1);
```

Funcția print este prezentată în continuare:

```
void print(int a) {
    const char tab[] = "0123456789ABCDEF";
    char data[5];
    int b = a * 1000, i = 0;
    while (i<4) {
        data[i] = b & 0x000F;
        b >>= 4;
        i++;
    }
    i = 0;
    while (i<4) {
        USART_Transmit(tab[ data[4-i-1] ]);
        i++;
    }
}
```

14. Cât timp durează transmisia efectivă a întregului șir de caractere de la exemplul precedent dacă interfața serială funcționează la 19200 baud, cu 8 biți pe caracter, fără paritate și cu un bit de stop?
15. Care este numărul de reset-uri pe secundă provocate de WATCHDOG pentru următorul program? Se consideră datele (ideale) din foaia de catalog.

```
no_init int flag;
void main(void) {
    if (flag == 1) {
        flag = 0;
        TCCR1B |= (1<<CS11);
        while (TCNT1 < 8000);
    } else {
        flag = 1;
        TCCR1B |= (1<<CS11) | (1<<CS10);
        while (TCNT1 < 2000);
        WDTCR |= (1<<WDP0);
    }
    asm("WDR");
    WDTCR |= (1<<WDE);
    while(5);
}
```