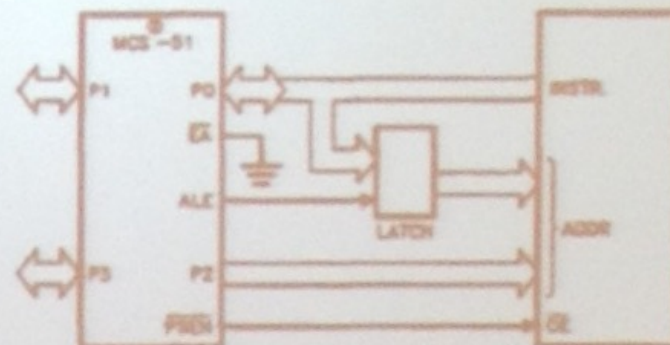


Hallgató neve:	Neptun kódja:	Ulóhely száma:	Felügyelő aláírása:	Pont, (jegy):
----------------	---------------	----------------	---------------------	---------------

Zárthelyi feladatok megoldása
a Mikrokontroller alapú rendszerek c. tárgyhoz
2015. március 31.

1) Az alábbi ábra egy külső egység illesztésének vázlatát mutatja 8051-es mikrokontrollerhez.



- Pontosan milyen típusú egység illesztéséről van szó, mi mutatja ezt egyértelműen?
Külső programmemória, mivel a kimenetét (/OE) a /PSEN (Program Store Enable) jel vezérli.
- Milyen információ(k) továbbítása történik a P0 porton? Milyen irányba?
Kettős szerepe van (amint azt a nyilak iránya is mutatja): itt kerül továbbításra a cím alsó bájtja (processzor kimenet), majd ezt követően történik meg a 8 bites utasításkód beolvasás (proc. bemenet)
- Milyen információ(k) továbbítása történik a P2 porton? Milyen irányba?
Ezen a porton továbbítódik a cím felső bájtja (proc. kimenet)

c) Milyen információ(k) továbbítása történik a P2 porton? Milyen irányba?

Ezen a porton továbbítódik a cím felső bájtja (proc. kimenet)

d) Mi az ALE jel és mi a /PSEN jel feladata?

Az ALE jel írja be az átmeneti tárolóba a cím alsó bájtját, a /PSEN jel engedélyezi a programmemória kimenetét megfelelő időpontban.

[Elérhető pontszám: 8 pont]

2) A következő kérdések az ARM processzorokra vonatkoznak. (Válaszok: I=igaz, H=hamis - rossz válasz: -)p)

Sorsz.	Kérdés	Válasz
a)	Elsősorban robot karok irányítására tervezték (nevét is innen kapta)	H
b)	RISC architektúrájú processzorcsalád	I
c)	Kis fogyasztású mobil eszközök népszerű processzora	I
d)	Periféria egységeket nem tartalmaz, azokat kívülről kell illeszteniük hozzá	H
e)	Rendelkezik egy ún. tömörített utasításkészlettel, amely elsősorban a kisebb alkalmazásokban (pl. kisebb adatbusz-szélesség) lehet előnyös	I
f)	Elsősorban assembly szintű programozását ajánlják, nem támogatja a magas szintű nyelveket	H
g)	A Cortex M processzorok a legnagyobb teljesítményű családtagok (M = Maximus)	H
h)	Legtöbb utasításának végrehajtása feltételhez köthető, a feltétel maga az utasításkód tartalmazza	I

[Elérhető pontszám: 8 pont]

f)	Cortex-M assembly szintű programozását ajánlják, nem támogatja a magas szintű nyelveket	H
g)	A Cortex M processzorok a legnagyobb teljesítményű családtagok (M = Maximus)	H
h)	Legtöbb utasításának végrehajtása feltételhez köthető, a feltétel maga az utasításkód tartalmazza	I

[Elérhető pontszám: 8 pont]

3) Mi a különbség a CY (átvitel) és az OV (túlesordulás) jelzőbitek között (8 bites mikrokontrollerek)?

a) Mikor lesz a CY értéke =1? (Melyik regiszterben milyen értékek közötti átmenetkor?)

Ha átvitel keletkezik az ACC legfelső (7.) bitjéről: $FF \Rightarrow 00$ túl-, vagy $00 \Rightarrow FF$ alulesordulásnál, ill. olyan logikai léptető/forgató műveleteknél, ahol CY érintve van a műveletben (átvitel a legfelső bitről). A CY a PSW regiszterben található.

b) Mikor lesz az OV értéke =1? (Melyik regiszterben milyen értékek közötti átmenetkor?)

Aritmetikai műveletek túl-/alulesordulásánál (kettes kompl. értelemben), tehát $7F \Rightarrow 80$ ill. $80 \Rightarrow 7F$ átmenetnél. (OV a legfelső bitről és az alatta lévő bitről keletkező átvitelek XOR-ja.) Az OV flag is a PSW regiszterben található.

c) Melyiket használjuk logikai műveletek (léptetés, forgatás) esetén?

a)-ból következően csak a CY flag használható ilyen célra

d) Mitől függ, hogy melyiket használjuk aritmetikai műveletek esetén?

Absz. értékű számábrázolás esetén (0..255) a CY, előjeles számábrázolás (-128..+127) esetén az OV használandó túl-/alulesordulás ellenőrzésére.

[Elérhető pontszám: 8 pont]

4) Tekintsük az alábbi 8051-es programrészletet (egy adott feladatot megvalósító assembly szubrutin):

```
Rutin:    mov R3, #0
Rutin1:  mov A, R3
         movc A, @A+DPTR
         clr C
         subb A, @R1
         jnz Rutin2
         inc R1
         inc R3
         djnz R2, Rutin1
         ret
Rutin2:  mov R2, #1
         ret
```

- Hány bemenő paramétere van a rutinnak, és hol található(k) az(ok)?
DPTR (egyik cím, kódmemória), R1 (másik cím, adatmemória), R2 (adatbájtok száma)
- Hány kimenő paramétere van a rutinnak, és hol található(k) az(ok)?
R2=0 (egyezés), R2=1 (eltérés), utóbbi esetben DPTR+R3 és R1 címen nem egyeznek az adatok
- Mi változik meg a CPU-ban a rutin hívása és visszatérése között (programozási szempontból)?
A, R1, R2 (visszatérési érték), R3, processzor státusz (jelzőflagek)
- Normál függvény vagy megszakítási rutin ez? Miért?
Normál függvény, mivel nem ment semmit és „ret”-tel (nem „reti”-vel) végződik
- Milyen feladatot lát el a programrészlet?
Összehasonlít max. R2 db bájtot DPTR ill. R1 címtől kezdődően – ha egyeznek, R2=0, ha nem, R2=1 a visszatérési érték és a címmutatók (R3-mal együtt) az eltérő bájtokra mutatnak

[Elérhető pontszám: 10 pont]

e) Milyen feladatot lát el a programrészlet?

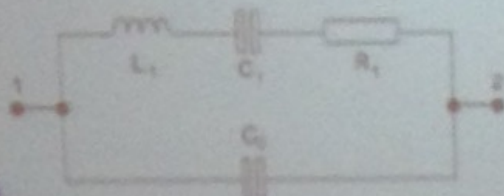
Összehasonlít max. R2 db bajtot DPTR ill. R1 címtől kezdődően – ha egyeznek, R2=0, ha nem, R2=1 a visszatérési érték és a címmutatók (R3-mal együtt) az eltérő bajtokra mutatnak.

[Elérhető pontszám: 10 pont]

5) Egy rezgőkvarc elektromos helyettesítő kapcsolásának paramétereit (a szokásos elnevezéssel): $R_1=20\Omega$, $L_1=11\text{mH}$, $C_1=0,00256\text{pF}$ és $C_0=1\text{pF}$.

- Rajzolja fel a rezgőkvarc elektromos helyettesítő kapcsolását!
- Számítsa ki a soros rezonanciafrekvencián való állandósult rezgés frekvenciáját (f_r)!
- Számítsa ki a párhuzamos rezonanciafrekvencián való állandósult rezgés frekvenciáját (f_s)!
- A tanult Pierce oszcillátor kapcsolásban $C_s=C_b=10\text{pF}$ blokkkondenzátort használunk. Mekkora lesz így az oszcillátor frekvenciája (f_p)?

[Elérhető pontszám: 12 pont]



a)

$$b) f_r = \frac{1}{2\pi\sqrt{L_1 C_1}} = \frac{1}{2\pi\sqrt{0.011 \cdot 0.00256 \cdot 10^{-12}}} \text{ Hz} = 29\,991\,888 \text{ MHz}$$

$$c) f_s = \frac{1}{2\pi\sqrt{L_1 C_1}} \sqrt{1 + \frac{C_1}{C_0}} = \frac{1}{2\pi\sqrt{0.011 \cdot 0.00256 \cdot 10^{-12}}} \sqrt{1 + \frac{0.00256}{1}} \text{ Hz} = 30\,030\,253 \text{ MHz}$$

$$d) f_p = \frac{1}{2\pi\sqrt{L_1 C_1}} \sqrt{1 + \frac{C_1}{C_0 + (C_s \times C_b)}} = \frac{1}{2\pi\sqrt{0.011 \cdot 0.00256 \cdot 10^{-12}}} \sqrt{1 + \frac{0.00256}{1+5}} \text{ Hz} = 29\,998\,285 \text{ MHz}$$

6) Mutassa be, hogyan végzi el a 16 bites fixpontos aritmetika (8 bites egészrész + 8 bites törtrész) a következő

$$d) f_r = \frac{1}{2\pi\sqrt{L_1 C_1}} \sqrt{1 + \frac{C_2}{C_1 + (C_2 \times C_3)}} = \frac{1}{2\pi\sqrt{0.011 \cdot 0.00256 \cdot 10^{-12}}} \sqrt{1 + \frac{0.00256}{1+5}} \text{ Hz} = 29.998285 \text{ MHz}$$

- 6) Mutassa be, hogyan végzi el a 16 bites fixpontos aritmetika (8 bites egészrész + 8 bites törtrész) a következő műveletet:

$$50.375 - 60.1$$

A megoldáshoz az operandusokat alakítsa át a megfelelő (16 bites) bináris számokká, és vegye figyelembe, hogy a mikrokontrollerek az a-b kivonást a + (-b) műveletssorral végzik el!

	+ 50.375 =	0	0	1	1	0	0	1	0	0	1	1	0	0	0	0	0
	+ 60.1 =	0	0	1	1	1	1	0	0	0	0	0	1	1	0	0	1
	+ 50.375 =	0	0	1	1	0	0	1	0	0	1	1	0	0	0	0	0
+	- 60.1 =	1	1	0	0	0	0	1	1	1	1	1	0	0	1	1	1
=	-9.725 =	1	1	1	1	0	1	1	0	0	1	0	0	0	1	1	1

A teljes értékű megoldáshoz szükséges annak bemutatása, hogy a számok egész- és törtrészei hogyan kerültek átalakításra bináris számokká, ill. kettes komplementes alakjukba, valamint a végeredmény helyességének ellenőrzése (külön lapon mutassa be a segédszámításait)!

A teljes értékű megoldásban szükséges annak bemutatása, hogy a számok egész- és törtrészre hogyan kerültek alakításra bináris számokká, ill. kettős komplementes alakjukba, valamint a végeredmény helyességének ellenőrzése (külön lapon mutassa be a végeredményt!)

Segédszámítások:

$50 = 32 + 16 + 2 = 110010_2$ és $0.375 = 0.25 + 0.125 = 0.011_2$
 vagy

Minden lépésben 2-vel osztva Minden lépésben 2-vel szorozva

50		0	LSB	0.375		0.75		0	MSB
25		1		0.750		1.50		1	
12		0		0.500		1.00		1	
6		0		0.000				0	
3		1						0	
1		1	MSB					0	
0								0	
								0	LSB

$60 = 32 + 16 + 8 + 4 = 111100_2$ (0.1-et is lehet próbálgatni.)

vagy

Minden lépésben 2-vel osztva Minden lépésben 2-vel szorozva

60		0	LSB	0.1		0.2		0	MSB
30		0		0.2		0.4		0	
15		1		0.4		0.8		0	
7		1		0.8		1.6		1	
3		1		0.6		1.2		1	
1		1	MSB	0.2		0.4		0	
0				0.4		0.8		0	
				0.8		1.6		1	LSB

Kivonó kettős komplementes képzése:

$-60.1 = -(0011100.00011001)_2 = 1100011.11100111_2$

Eredmény ellenőrzése:

$11110110.01000111_2 = -(00001001.10111001)_2 =$
 $= -(9 + (128+32+16+8+1)/256) = -(9 + 185/256) = -9.72265625$
 az eltérés a számbírási pontatlanságából fakad

[Elérhető pontszám: 10 pont]

7) Egy egyenes sínen létező rakétarobót 25 m-es pályán mozog, maximuman 5 m/s sebességgel. A rakétarobó maximális felhajtóereje 3000 newton, továbbá azonnal kikapcsolhat az erő 1000 newton

$= -(9 + (128+32+16+8+1)/256) = -(9 + 185/256) = -9.72265625$
az eltérés a számbázis pontatlanságából fakad

[Elérhető pontszám: 10 pont]

- 7) Egy egyenes úton közlekedő rakodórobot 25 m-es pályán mozog, maximálisan 5 m/s sebességgel. A hajtómotor maximális fordulatszáma 3000/perc, tengelyével közvetlen kapcsolatban van egy 1000-fordulat felbontású inkrementális szöghelyzet adó. Az adó optocsatolókon keresztül kapcsolódik a számítógépben található, négyszeres kiértékelés szerint dolgozó számlálóegységhez. A robot pillanatnyi sebességét a következő módon képezzük: az operációs rendszer pontosan 1 msec-ként kiolvassa a robot aktuális pozícióját és ezek különbségét képezi, az így kapott számérték (kettes komplementes egész) a sebességgel lesz arányos.
- Mi az a maximális impulzsfrekvencia, amire a mikrokontroller jelillesztő áramköröket méretezni kell?
 - Milyen pontossággal (pl. 1 cm) tudjuk meghatározni azt, hogy hol helyezkedik el a robot a pályán belül (az

rendszere rendszeresen megújított és az új generációs rendszereket a vállalatoknál található, négyszeres kiértékelés szerint dolgozó szakemberekhez. A robot pillanatnyi sebességét a következő módon képezzük: az operációs rendszer pontosan 1 msec-ként kiolvassa a robot aktuális pozícióját és ezek különbségét képezi, az így kapott számérték (kettes komplementes egész) a sebességgel lesz arányos.

- Mi az a maximális impulzsfrekvencia, amire a mikrokontrolles jelkezelő tranziszorú műveleti kell?
- Milyen pontossággal (pl. 1 cm) tudjuk meghatározni azt, hogy hol helyezkedik el a robot a pályán belül (az áttételek hibáit hanyagoljuk el)?
- Legalább hány biten kell a pozícióinformációt (pozitív egész) ábrázolnunk ahhoz, hogy a robot a teljes pályát befuthassa?
- Legalább hány biten kell a sebességinformációt (kettes komplementes egész) ábrázolnunk ahhoz, hogy a robot bármilyen sebessége ($-v_{max}$, $+v_{max}$) esetén helyes eredményt kapjunk?
- Mekkora az a minimális haladási sebesség, amikor a fenti módszer szerint képzett robotsebesség még legalább 10% pontossági?

[Elérhető pontszám: 12 pont]

- 3 -

a) A motor maximális fordulatszáma: $n_{max} = 3000 \text{ ford/perc} = 50 \text{ ford/s}$. Az adó felbontása 1000 ford.

- a) A motor maximális fordulatszáma: $n_{\max} = 3000 \text{ ford/perc} = 50 \text{ ford/s}$. Az adó felbontása 1000/ford, tehát a maximális impulzusfrekvencia $f_{\max} = 50 \cdot 1000 = 50\,000 \text{ Hz}$
- b) A négyszeres kiértékelés miatt $f_{\text{inkr}} = 4000/\text{ford}$. Maximális sebességnél 1 fordulat ideje ($1/50 \text{ s}$) alatt a robot $5 \text{ m/s} \cdot 1/50 \text{ s} = 100 \text{ mm}$ -t tesz meg emiatt a keresett felbontás = $100/4000 \text{ mm} = 25 \mu\text{m}$
- c) $v_{\max} = 5 \text{ m/s}$, $L = 25 \text{ m}$, tehát a futásidő max. sebességgel: $t_{\text{pálya}} = 5 \text{ s}$
A motor fordulatainak száma: $n_{\text{pálya}} = 5 \text{ s} \cdot 50 \text{ ford/s} = 250 \text{ ford}$, az ábrázolandó pozícióértékek száma $4000/\text{ford} \cdot 250 \text{ ford} = 1\,000\,000$, tehát a pozíciót min. 20 biten kell tárolni (valójában inkább 24 vagy 32 biten, mivel $2^{20} = 1\,048\,576$, ez így nagyon „cipókanalas” lenne).
- d) Ha a robot maximális sebességgel halad, az 1 msec alatt beérkező inkrementek (4x kiértékelés !) száma: $v_{\max} = 4 \cdot 50\,000 \text{ Hz} \cdot 1/1000 \text{ s} = 200 \text{ inkr.}$ A sebesség kettes komplementes kódú ábrázolásához tehát 9 bit elegendő (8 adat + 1 előjelbit).
- e) Maximális sebesség esetén 1 msec alatt 200 inkrement érkezik be, tehát a pontosság $1/200 = 0.5\%$. A megkívánt 10%-os pontossághoz a sebesség $10/0.5 = 20$ -ad részére (v_{\max} 5%-ára) csökkenthető (ekkor 10 inkrement érkezik be 1 msec alatt), ekkor a haladási sebesség $5 \text{ m/s} / 20 = 250 \text{ mm/s}$.

3) A bekeretezett mezőben egy 8051-es projekt valamelyik fájljának tartalmát látja. Válaszoljon az alábbi kérdésekre 1-1 mondattal!

megkívánt 10%-os pontossághoz a sebesség $10/0.5=20$ -ad részére (v_{max} 5%-ára) csökkenthető (ekkor 10 inkrement érkezik be 1 msec alatt), ekkor a haladási sebesség $5 \text{ m/s} / 20 = 250 \text{ mm/s}$.

8) A bekeretezett mezőben egy 8051-es projekt valamelyik fájljának tartalmát látja. Válaszoljon az alábbi kérdésekre 1-1 mondattal!

```
:1000030075840F75E3C075FFDE75FFAD75A54022DE
:0B001F00120003A2B7B3929680F922F2
:03000000020013E8
:0C001300787FE4F6D8FD75810702001F1D
:00000001FF
```

- a) Milyen típusú fájlról lehet szó (mi a fájl kiterjesztése és szokásos elnevezése)?
.HEX, Intel hex formátumú adatfájl
- b) Milyen célra használunk egy ilyen fájlt?
Tipikusan a programkód tárolására, majd a programmemória programozására.
- c) Hány fontosabb mező különíthető el minden sorban és melyek ezek?
6 db = Rekordkezdet (:), adatbájtok száma (nn), tárolási cím (aaaa), rekordtípus (tt), nn db adatbájt, ellenőrző összeg (CS).
- d) A végső felhasználás szempontjából hány adat típusú bájt található a fájlban? (Indokolja!)
 $16 (10h) + 11 (0bh) + 3 (03h) + 12 (0ch) = 42$ bájt
- e) Az utolsó sor kivételével felcserélhetők-e tetszőlegesen a fájl sorai egymás között? (Indokolja!)
Esetünkben igen, mivel nincs címkiterjesztő rekord típus (2 v. 4) és minden sor saját címmel rendelkezik.
- f) Mi az utolsó sor szerepe?
Fájl vége rekord (1-es típus) – hossz=0, cím=0

[Elérhető pontszám: 12 pont]

9) Két 32 bites (egyszeres pontosságú) lebegőpontos szám – melyek felépítése: 1 bit előjel, 8 bit exponens, 23 bit mantissa – értéke hexadecimális alakban a következő:

$$1.111000 \cdot 2^2 = +7.5$$

1) Mi az utolsó sor szerepe?

Fájl vége rekord (1-es típus) – hossz=0, cím=0

[Elérhető pontszám: 12 pont]

9) Két 32 bites (egyszeres pontosságú) lebegőpontos szám – melyek felépítése: 1 bit előjel, 8 bit exponens, 23 bit mantissa – értéke hexadecimális alakban a következő:

$$\begin{array}{l} \text{a) } 40F00000h = \boxed{+} \boxed{1.1111000} * 2^{\boxed{2}} = \boxed{+7.5} \\ \text{b) } BF980000h = \boxed{-} \boxed{1.0011100} * 2^{\boxed{1}} = \boxed{-1.1875} \end{array}$$

Adja meg a számok decimális értékét! Teljes értékű megoldás: normál alak (bináris számokkal) = dec. szám.

[Elérhető pontszám: 10 pont]

$$40F00000h = 0|100\ 0000\ 1|111\ 0000\ 0000\ 0000\ 0000\ 0000$$

+| =129 | (ez csak a törtrész)

$$\Rightarrow 1.1111000 * 2^2 = 111.100 = 7 + 0.5 = 7.5$$

$$BF980000h = 1|011\ 1111\ 1|001\ 1000\ 0000\ 0000\ 0000\ 0000$$

-| =127 | (ez csak a törtrész)

$$\Rightarrow 1.0011100 * 2^0 = 1.0011100 = 1 + 0.125 + 0.0625 = 1.1875$$

10) Elemezze az alábbi, STM32F407 mikrokontrollerre készült programrészlet, és válaszoljon a következő kérdésekre (az inicializáló program a szükséges portlábakat kimenetnek állította be nulla kezdőértékkel):

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    uint16_t LedVal = 0;
    uint16_t PortVal = 0;

    PortVal = LedVal = GPIOD->ODR;
    PortVal &= 0x00ff; // (1)
    LedVal &= 0xff00; // (2)
    LedVal <<= 2; // (3)
    if (LedVal == 0)
        LedVal = 0x0300; // (4)
    PortVal |= LedVal; // (5)
    GPIOD->ODR = PortVal;
}
```

a) A fenti kódrészlet melyik port mely lábainak értékét módosítja?

D port, 8-15 lábak

b) Adja meg a rutinban szereplő értékadások részeredményeit a rutin első 3 meghívásakor!

Lépés	PortVal (1)	LedVal (2)	LedVal (3)	(LedVal) (4)	PortVal (5)
1	0x0000	0x0000	0x0000	0x0300	0x0300
2	0x0000	0x0300	0x0c00	(0x0c00)	0x0c00
3	0x0000	0x0c00	0x3000	(0x3000)	0x3000

c) Mit csinál a fenti kódrészlet? (Valamennyi kimenet egy-egy világító diódát hajt meg.)

A D port felső 16 bitjére kapcsolódó LED-ek közül mindig két LED világít egyszerre, és így kettesével balra lépdelve (majd az elejére ugorva) mozgó futófényt valósít meg:

00000011 - 00001100 - 00110000 - 11000000 - 00000011

[Elérhető pontszám: 10 pont]