

Számrendszerek, számábrázolás, alapfogalmak

F1. Az Internet kommunikáció a TCP/IP protokollon és az eszközökhöz rendelt IP címeken alapul. Az IP cím az IPv4 rendszerben 32 bites (az IPv6-ban 128 bites). A cím könnyebb kezelhetősége érdekében szokásosan a 32 bites cím 4 bájtos formátumban adott, az egyes bájtok értékét decimálisan kifejezve. Pl. az otthoni hálózatokban gyakori a 192.168.1.1 címspecifikáció.

a) Generálja a bináris számformátumát ennek az IP címnek!

Megoldás:

A számrendszerek közötti konverzióra két módszert alkalmazhatunk. Az egyik az új számrendszer hatványaival történő összehasonlítás és folyamatos kivonás, a másik a maradékos osztás.

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	256	128	64	32	16	8	4	2	1
	0	1	1	0	0	0	0	0	0
192		-128							
		64	-64						
			0						

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	256	128	64	32	16	8	4	2	1
	0	1	0	1	0	1	0	0	0
168		-128							
		40		-32					
				8		-8			
						0			

$$192_{10} = 128 + 64 = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 11000000_2$$

$$168_{10} = 128 + 32 + 8 = 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 10101000_2$$

$$1_{10} = 1 = 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 00000001_2$$

$$\text{Tehát } 192.168.1.1_{10} = 11000000.10101000.00000001.00000001_2$$

b) A BME MIT kuka.mit.bme.hu nevű szerverének IP címe 152.66.252.1. Generáljuk ezt a címet is, most a maradékos osztás algoritmust használva!

Megoldás:

$$152:2 = 76, \quad \text{maradék } 0 \text{ (Ez egyúttal az LSB, a legkisebb helyiértékű bit)}$$

$$76:2 = 38, \quad \text{maradék } 0$$

$$38:2 = 19, \quad \text{maradék } 0$$

$$19:2 = 9, \quad \text{maradék } 1$$

$$9:2 = 4, \quad \text{maradék } 1$$

$$4:2 = 2, \quad \text{maradék } 0$$

$$2:2 = 1, \quad \text{maradék } 0$$

$$1:2 = 0, \quad \text{maradék } 1, \text{ ez az MSB, a legnagyobb helyiértékű bit}$$

$$152_{10} = 10011000_2$$

$$\text{Hasonlóan számolva } 66_{10} = 01000010_2, \quad 252_{10} = 11111100_2,$$

$$\text{Tehát } 152.66.252.1_{10} = 10011000.01000010.11111100.00000001_2$$

F2. Adja meg az előző értékeket hexadecimális formában!

Megoldás:

A bináris formátumban, az egyes bájtok 4-4- bitből álló csoportjai alkotnak egy hexadecimális digitet.

A hexadecimális karakterek szimbólumai a 0...9 után a 10..15 értékekre az A, B, C, D, E, és F.

Tehát $192.168.1.1_{10} = 1100_0000.1010_1000.0000_0001.0000_0001_2 = C0.A8.01.01_{16}$

Tehát $152.66.252.1_{10} = 1001_1000.0100_0010.1111_1100.0000_0001_2 = 98.42.FC.01_{16}$

F2. Előjeles számábrázolás és műveletek 8 biten az alább megadott értékekkel.

a) Töltse ki a táblázatot!

Decimális értékek	Bináris értékek	2's C negatív
a = 20	a = 00010100	-a = 11101100
b = 14	b = 00001110	-b = 11110010
c = 42	c = 00101010	-c = 11010110
d = 111	d = 01101111	-d = 10010001

A kettes komplement képzést a bitek invertálásával és 1 hozzáadásával végeztük.

Pl. az „a” értékre: $00010100 \rightarrow \text{inv} \rightarrow 11101011 + 1 \rightarrow 11101100$

b) Végezze el az **a + b**, az **a + d**, a **d - c** és az **a + d - c** műveleteket bináris aritmetikával!

Megoldás:

a + b	a + d	d - c = d + (-c)
00010100	00010100	01101111
<u>+00001110</u>	<u>+01101111</u>	<u>+11010110</u>
00100010	10000011	01000101
$2^5 + 2^1$	$-2^7 + 2^1 + 2^0$	$2^6 + 2^2 + 2^0$
34	-125	69

Az **a + d** összeadás eredménye már nem ábrázolható a 8 bites kettes komplement formátumban, mert ennek értéktartománya $-2^{(8-1)} \dots 0 \dots +2^{(8-1)} - 1$, azaz $-128 \dots 0 \dots +127$. Ezért a 8 bites kódot kiolvassva egy negatív számot kapunk, hiszen az első bit (MSB) értéke 1.

Ennek tudatában az **a + d - c** művelet elvégzése előtt, óvatos stratégiát alkalmazva a sorrendet átrendezzük, hogy elkerüljük az esetleges túlcsofordulást. (Egész típusú számábrázolás esetén az összeadás kommutatív.) Tehát a biztonságosabbnak tűnő műveleti sorrendek a következők lehetnének: **(a - c) + d**, vagy még inkább **(d - c) + a**. Ezek elvégzését most nem ismertetjük, könnyen reprodukálható.

Mint érdekesség, megemlítjük, hogy a kettes komplement számábrázolás modulo tulajdonságai miatt az eredeti sorrendben végzett számítás is helyes eredményt ad, annak ellenére, hogy egy esetben a belső eredmény túlcsofordulást mutat.

a + d	(a + d) - c = (a + d) + (-c)
00010100	10000011
<u>+01101111</u>	<u>+11010110</u>
10000011	101011001
$-2^7 + 2^1 + 2^0$	$2^6 + 2^4 + 2^3 + 2^0$

-125

69

A sárgával jelölt 9. bit, az aritmetika átvitel a 9. bit pozícióban, már nem része a 8 bites kettes komplement eredménynek, tehát az eredmény numerikus értékének meghatározásában elhagyható.

SZORGALMI FELADAT

A szorgalmi feladatok csupán az érdeklődés felkeltésére, az ismeretek bővítésére szolgálnak. Ez a feladat az mutatja be, hogy egy egyszerű félvezetőtechnológiájú hőmérsékletérzékelő alkatrész 16 bites soroskimenetű adatformátuma előjeles, valós számformátumot alkalmaz, melynek megfelelő használata feltételezi az első előadáson megismert fixpontos előjeles kettes komplement valós számbábrázolás értelmezését.

F4. Valós számok. A TMP121 hőmérsékletmérő szenzor adatlapjának részletei az alábbiakban láthatók. A hőmérőt a félév végén használni fogjuk, a 12. labormérésen a MiniRISC CPU-val a mért értéket Az SPI szabványú kommunikációt szoftveresen megvalósítva kiolvassuk és decimálisan kijelzük egy 4 digites numerikus kijelzőn. A teljes adatlap elérhető a segédletben vagy a <http://www.ti.com/lit/ds/symlink/tmp121.pdf> linken. A feladat szövege az ábrák után folytatódik.



TMP121
TMP123

SBOS273C – JUNE 2003 – REVISED FEBRUARY 2005

1.5°C Accurate Digital Temperature Sensor with SPI™ Interface

FEATURES

- **DIGITAL OUTPUT:** SPI-Compatible Interface
- **RESOLUTION:** 12-Bit + Sign, 0.0625°C
- **ACCURACY:**
±1.5°C from -25°C to +85°C (max)
- **LOW QUIESCENT CURRENT:** 50µA (max)
- **WIDE SUPPLY RANGE:** 2.7V to 5.5V
- **TINY SOT23-6 PACKAGE**
- **OPERATION TO 150°C**

DESCRIPTION

The TMP121 and TMP123 are SPI-compatible temperature sensors available in the tiny SOT23-6 package. Requiring no external components, the TMP121 and TMP123 are capable of measuring temperatures within 2°C of accuracy over a temperature range of -40°C to +125°C. Low supply current, and a supply range from 2.7V to 5.5V, make the TMP121 and TMP123 excellent candidates for low-power applications.

The TMP121 and TMP123 are ideal for extended thermal measurement in a variety of communication computer

D15	D14	D13	D12	D11	D10	D9	D8
T12	T11	T10	T9	T8	T7	T6	T5
D7	D6	D5	D4	D3	D2	D1	D0
T4	T3	T2	T1	T0	0	Z	Z

Table 1. Temperature Register

TEMPERATURE (°C)	DIGITAL OUTPUT(1) (BINARY)	HEX
150	0100 1011 0000 0000	4B00
125	0011 1110 1000 0000	3E80
25	0000 1100 1000 0000	0C80
0.0625	0000 0000 0000 1000	0008
0	0000 0000 0000 0000	0000
-0.0625	1111 1111 1111 1000	FFF8
-25	1111 0011 1000 0000	F380
-55	1110 0100 1000 0000	E480

(1) The last two bits are high impedance and are shown as 00 in the table.

Table 2. Temperature Data Format

a) Mi az adatformátum? Kiolvasható-e az angol nyelvű dokumentációból?

Megoldás:

A JELLEMZŐK felsorolás szerint a mért adat felbontás 12 bit + előjel, tehát összesen 13 bit. A digitális kimeneti adat viszont D15...D0 16 bites bitsorozat, de ebből ezek szerint csak 13 érvényes adatbit, az utolsó 3 érvénytelen. Mivel a felbontás alapján a legkisebb kifejezhető hőmérsékletkülönbség 0,0625 °C, ez éppen 1/16 °C, tehát a 13 adatbitből 4 db törtértékű jegy, így 9 egészértékű bit marad.

Az adatformátum fixpontos 13 bites előjeles kettes komplementes valós (nem egész) formátum, azaz (13.4). A kettes pont a 2^0 helyiérték után van, a törtjegyeket megelőzően.

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
-256	128	64	32	16	8	4	2	1	0,5	0,25	0,125	0,0625

b) Mekkora a maximális és minimális megadható értékek az adott formátumban?

Megoldás:

A (13.4) formátumban a maximális és a minimális értékek:

$$\begin{aligned} +\text{MAX} &= +255,9375 \text{ °C,} \\ -\text{MAX} &= -256,0000 \text{ °C.} \end{aligned}$$

Részletesen:

$+\text{MAX} = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 + 0,5 + 0,25 + 0,125 + 0,0625 = 255,9375$, azaz az előjelet kivéve minden bit aktív 1 értékű.

$-\text{MAX} = -256$, azaz csak az előjel bit aktív (negatív súllyal, az össze többi bit értéke nulla).

A hőmérő ezt a tartományt nem tudja átfogni, az adatlap specifikációja szerint csak -40°C és +150°C között működőképes (szilícium félvezető alapú elektronikai eszköz).

c) Értelmezzük a 0xFE8 és a 0x0540 16 bites kiolvasásokat! Mekkora hőmérsékletet jelentenek?

Megoldás:

A két adat az eredeti 16 bites kiolvasás, amiből, mint tudjuk az utolsó 3 bit értéktelen. Tehát a valódi (13.4) –es formátumú adatot ezek elhagyásával (vagy 3 bittel jobbra léptetve, tehát 8-cal osztva) kapjuk meg. (A bináris formátumban a _ jel az olvashatóságot segíti, a „.” a kettes pont pozíciója.)

$0xFE8 = 1111_1110_1010_1000_2$, alsó 3 bitet elhagyva $1111_1110_1010_1_2$, beillesztve a „kettes” pontot $1111_1110_1.010_1_2$, átrendezve az _ elválasztókat a könnyebb olvashatóság érdekében: $1_1111_1101.0101_2$,

$$-256 + 128 + 64 + 32 + 16 + 8 + 4 + 1 + 0,25 + 0,0625 = -2,6875 \text{ °C.}$$

$$0x0540 = 0000_0101_0100_0000_2 \rightarrow 0000_0101_0100_0_2 \rightarrow 0000_0101_0.100_0_2 \rightarrow$$

0_0000_1010.1000₂, ennek értéke

$$8 + 2 + 0,5 = +10,5 \text{ } ^\circ\text{C}.$$

Ezek az értékek persze csak $\pm 1,5^\circ\text{C}$, azaz korlátozott pontosságúak. A valódi hőmérséklet értéke valahol a -4.1875°C és -1.1875°C illetve $+9^\circ\text{C}$ és $+12^\circ\text{C}$ között van. Sajnos a digitális szenzoroktól kapott nagy felbontású adatok gyakran félrevezetően pontosnak is tűnnek, tehát a leolvasásnál, értékelésnél ezt mindenképpen figyelembe kell venni.