

20.....év ...hó ...nap

NÉV:..... Neptun kód:..... gyak/lab kurzus: .....

A feladatokat önállóan, meg nem engedett segédeszközök használata nélkül oldottam meg:

Olvasható aláírás:.....

Kedves Kolléga! A kitöltést a dátum, név és aláírás rovatokkal kezdje! Az alábbi kérdésekre a válaszokat - ahol lehet - mindig a feladatlapon oldja meg! A feladatok megoldása során a részletes kidolgozást, ha az esetleg nem fér rá a ZH lapra, külön papíron végezze, (egyértelműen jelölje, hogy melyik lap melyik feladathoz tartozik, a papírra már a kezdetkor írja rá a nevét és Neptun kódját) és ezeket a papírokat is adja be a dolgozatával! A kérdésekre a táblázatok vagy a pontozott vonalak értelemszerű kitöltésével válaszoljon, hacsak külön másként nem kérjük. **Mindenütt a legegyszerűbb megoldás éri a legtöbb pontot.** Jó munkát!

F1:  
F2:  
F3:  
Σ :  
IMSC:

**F1 feladatok (15p)**

**F1/1.** (3p) Végezze el a megadott Boole algebrai kifejezésen az egyszerűsítést vagy átalakítást csak a feladathoz megadott tétel (akár többszöri) alkalmazásával!

- a.  $(/AB + C) + (CD + E) / (AB + C) = .. /AB + C....$  Alkalmazza az elnyelési tételt!
- b.  $(A + /CD)B/E + /(A + /CD)B/E + BE = .. B/E + BE = B.....$  Alkalmazza az egyszerűsítési
- c.  $/(A + B + /C) = ..... /A/BC.....$  Alkalmazza De' Morgan tételt!

**F1/2. a.** (3p) Írja fel az alább megadott azonosság duálisát! (Alkalmazza a dualitás tételt az azonosság mindkét oldalára!)

$(B * C) + (/B * D) + (C * D) = B * C + (/B * D)$                        $(B + C) / (B + D) (C + D) = (B + C) / (B + D)$

**b.** Adja értékül a fenti **eredeti** Boole algebrai azonosság **bal oldalát** az f változónak Verilog-ban! Az első sorban definiálja az f változót!  
...**wire**.. f;    // f változó definiálása

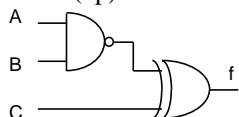
...**assign**..f = ..**(B & C) | (~B & D) | (C & D);**.....                      // értékadás az f változónak

**F1/3.** (1p) Számítsa ki, az alábbi Boole algebrai alakban megadott f(A,B,C) logikai függvény **értékét** a megadott bemeneti kombináció esetén!

Bemeneti kombináció: **A,B,C = 1,0,1**

A logikai függvény és értéke:                       $f = A \oplus /(B + /C) = ...0....$

**F1/4.** (4p) Töltse ki az alábbi kapcsolás rajzzal megadott f logikai függvény igazságtábláját!



A	B	C	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0

A	B	C	f
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

**F1/5.** (4p) Igazságtáblájával adott az f logikai függvény.

**a.** Adja meg a függvény diszjunktív normál alakját (DNF tehát ne egyszerűsítsen)!

A	B	C	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0

A	B	C	f
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$f_{DNF} = ... /A/B/C + /A/BC + A/B/C + ABC .....$

**F2 feladatok (15p)**

**F2/1.** (2p) Alakítsa át az alábbi **decimális** számot **bináris**sá! A részletszámításokat is írja le, anélkül nem kap pontot! A szám: **61**

Részletszámítások:

Az eredmény:..**111101**.....

$61/2=30$  marad **1** (LSB)     $30/2=15$  marad **0**     $15/2=7$  marad **1**  
 $7/2=3$  marad **1**     $3/2=1$  marad **1**     $1/2=0$  marad **1**

**F2/2.** (1p) Alakítsa át az alábbi **bináris** számot **hexadecimálissá**! A részletszámításokat is írja le, anélkül nem kap pontot! A szám: 1011000101

Részletszámítások: **0010\_1100\_0101**  
                           **2      C      5**

Az eredmény:.....**2C5**.....

**F2/3.** (1p) Alakítsa át az alábbi **hexadecimális** számot **decimálissá**! A szám: 2D

Az eredmény: **...45**.....

**F2/4.** (2p) Végezze el az alábbi összeadást a 7 bites bináris előjel nélküli számokon (op1 + op2)! A legfelső sor **(i+1)-edik bitjén** tüntesse fel az i-edik bitek összeadásakor keletkező átvitelt! (Amit az i+1-edik bitek összeadásakor figyelembe vesz.) A szürke cellákba ne írjon semmit!

átvitel:	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	-
op1:	-	0	1	1	0	1	1	1
op2:	-	1	0	0	1	0	1	1
összeg:	-	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>

Ellenőrizze, hogy az összeg helyes-e? **..nem helyes**.....  
 Ha nem, mi az oka? **.....túlsordulás, nem fér el 7 biten.**

**55+75 =130 >127**

**F2/5.** (2p) Végezze el a bináris szorzást a megadott előjel nélküli bináris számokkal. A **szorzást a szorzó LSB-jével kezdje!** A részletszámításokat is írja le, anélkül nem kap pontot! Az eredményt binárisan külön is írja le!

Részletszámítások: 10110\*1010

Az eredmény: **11011100**

**00000**  
**10110**  
**00000**  
**10110**  
**11011100**

**F2/6.** (3p) **a.** Képezze a megadott 7 bites **2-es komplementes ábrázolású szám 2-es komplementét!** A részletszámításokat is írja le, anélkül nem kap pontot! A szám: **1110010**

Részletszámítások:

Az eredmény (7 biten):

0	0	0	1	1	1	0
---	---	---	---	---	---	---

**1110010 inv->0001101+1->0001110**

**b.** Írja fel **az eredeti 2-es komplementes kódolású számot** számot 9 biten (előjel kiterjesztés)!

1	1	1	1	1	0	0	1	0
---	---	---	---	---	---	---	---	---

**c.** Adja meg **az eredeti 2-es komplementes számot előjeles decimális számként!** **.....-14**.....

**F2/7.** (1p) Mekkora a **legkisebb** ábrázolható szám 9 bites 2-es komplementes kódban? **...-256**.....

**F2/8.** (1p) Alakítsa **decimálissá** a következő **BCD kódban megadott** számot! 01010111 = **...57**.....

**F2/9.** (2p) Konvertálja át a megadott **2-es komplementes fixpontos** ábrázolású számot decimálissá!

A szám 3 db 2-eses tört bitet tartalmaz. A részletszámításokat is írja le, anélkül nem kap pontot!

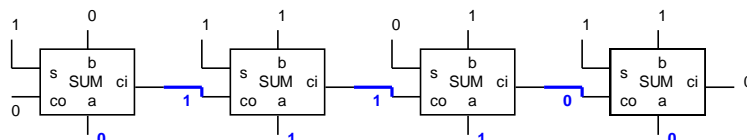
A szám: **1110 100** Részletszámítások:

Az eredmény:.....**-1.5**.....

**1110 100 inv->0001011+1->0001100-> egész.tört 0001.100 -> 1+0.5**

**F3 feladatok (20p)**

**F3/1.** (2p) Kaszkádosítsa az alábbi 1 bites összeadókat! Írja be az ábrába a kaszkádosított összeadók a bemeneteinek és **co** kimeneteinek logikai értékeit a többi be- és kimenetre írt adat alapján!



**F3/2.** (3p) **a.** Töltsd ki a teljes összeadó **co** kimenetének igazságtábláját!

a	b	ci	co
0	0	0	<b>0</b>
0	0	1	<b>0</b>
0	1	0	<b>0</b>
0	1	1	<b>1</b>

a	b	ci	co
1	0	0	<b>0</b>
1	0	1	<b>1</b>
1	1	0	<b>1</b>
1	1	1	<b>1</b>

**b.** Adja meg a **co** függvényt legegyszerűbb SOP alakban!

**co = ....a\*b + a\*ci + b\*ci**.....

**F3/3.** (1p) A következő Verilog leírással egy **kivonót** adtunk meg: {co, s} = a + (~b + 1);

Mi lesz **a** értéke, ha **a** és **b** előjel nélküli számábrázolású és **a < b**? **co = ..0**...

(Az **a**, **b** és **s** azonos méretű bitvektorok, **co** 1 bites.)

**F3/4. a.** (2p) Az alábbi Verilog leírás egy ismert funkcionális elemet ír le. Adja meg a funkcionális elem *nevét* és jeleinek *funkcióját*!

```
wire e;  
wire [1:0] s;  
wire [3:0] out;  
assign out[0] = e & (s==2'b00);  
assign out[1] = e & (s==2'b01);  
assign out[2] = e & (s==2'b10);  
assign out[3] = e & (s==2'b11);
```

**b.** Adja meg az out[3:0] kimenet értékét *binárisan* az alább megadott bemenetek esetén:  
e: 0, s: 10 out[3:0] = **.0000..**      e: 1, s: 00 out[3:0] = **.0001..**

neve: **...2/4-es dekóder...** e: **engedélyezés** s: **select...** out[3:0]: **kimenetek...**

**F3/5.** (6p) **a.** Adja meg egy engedélyezhető 2/4-es 4 bites *busz multiplexer* Verilog viselkedési leírását! Elkezdjük, folytassa!

```
wire en;  
wire [1:0] sel;  
wire [3:0] in0, in1, in2, in3;  
reg [3:0] out;  
always@(*)  
begin  
if(en)  
  case(sel)  
    2'd1: out <= ...in1.....;  
    2'd2: out <= ...in2.....;  
    2'd3: out <= ...in3.....;  
  default: out <= ...in0.....;  
  endcase  
else.  
    out <= 4'b0000;  
end
```

**b.** Adja meg a fenti multiplexer kimenetének *értékét* az alábbi bemeneti kombinációk esetén!

en = 1, sel = 10; in0 = 4'h0, in1 = 4'h1, in2 = 4'h2, in3 = 4'h3      out = **.....4'h2.....**  
en = 0, sel = 10; az in bemenetek értéke ugyanaz mint fent.      out = **.....4'h0.....**

**F3/6.** (6p) **a.** Adja meg egy 4 bemenetű *enkóder* Verilog viselkedési leírását! Elkezdjük, folytassa!

```
wire en;  
wire [3:0] in;  
reg [1:0] out;  
always@(*)  
begin  
if(en)  
  case(in)  
    4'b0001: out <= 2'h0.....;  
    4'b0010: out <= 2'h1.....;  
    4'b0100: out <= 2'h2.....;  
    4'b1000: out <= 2'h3.....;  
  default: out <= 2'h0.....;  
  endcase  
else  
    out <= 2'h0.....;  
end
```

**b.** Adja meg a fenti enkóder kimenetének *értékét* az alábbi bemeneti kombinációk esetén!

en = 1, in = 0100; out = **.....2'h2.....**      en = 0, in = 0001; out = **.....2'h0.....**

**IMSC1.** (3p) Adja meg egy olyan **ADD4** nevű összeadó modul Verilog leírását, amely két **4 bites** bementére (**a**, **b**) érkező 2-es komplementes kódolású számot ad össze és az eredmény előjel helyesen jelenik meg 2-es komplementes kódban az **5 bites out** kimenetén. Röviden magyarázza el a megoldásának működését!

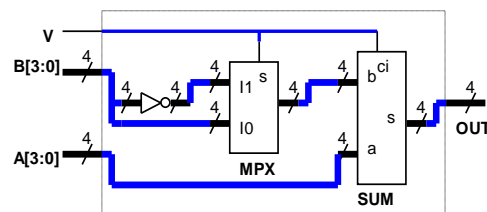
```
modul ADD4(input [3:0] a,b, output [4:0] out)
```

```
    assign out = {a[3],a} + {b[3],b};
```

```
endmodule
```

**5 bites összeadó bementeire 5 bites előjel kiterjesztéssel adjuk a 4 bites összeadandókat. Így túlsordulás nélkül 5 biten keletkezik az eredmény.**

**IMSC2.** (4p) Készítsen 4 bites 2-es komplementes számokkal működő összeadó/kivonót a berajzolt egységek összekötésével! Az egység összead (**OUT=A+B**), ha **V = 0** és kivon (**OUT=A-B**), ha **V = 1**. (Az esetleges túlsordulással most nem törődünk.)



Adja meg egy ugyanezt a funkciót megvalósító modul legegyszerűbb leírását Verilogban!

```
module ADD_SUB4(input V, input [3:0] A,B, output [3:0] OUT)
```

```
// a modul logikájának leírása Verilogban:
```

```
    assign OUT = V ? (A-B) : (A+B);
```

```
endmodule
```

Maximális pontszám: 50 pont (IMSC: 7p)    Rendelkezésre álló idő: 100 perc