

20.....év ...hó ...nap

NÉV:.....neptun kód:..... 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 nagyfeladatonként 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!

E:
F1:
F2:
F3:
Σ :

Ellenőrző kérdések (25p)

E1. Konvertálja az alábbi számokat a kért formátumra! (2p)

kiinduló formátum	konvertálandó szám	kért formátum	konvertált szám
decimális	20	6 bites előjel nélküli bináris	
8 bites 2-es komplement	11111101	decimális	

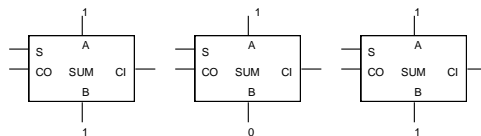
E2. Mi az előnye a 2-es komplement számábrázolásnak? Húzza alá a legkedvezőbb választ! (1p)

- a. A előjeles számok viszonyának (kisebb/nagyobb) eldöntésénél előnyös.
- b. Előjeles számok összeadásánál előnyös.
- c. Előjeles számok 2 hatvánnyal való szorzásánál előnyös.

E3. Egyszerűsítse az alábbi Boole algebrai kifejezést, Boole algebrai átalakításokkal! A részeredményeket is írja le! (1p)

$A + /A.B = \dots\dots\dots$

E4. Kaszkádosítsa az alábbi, 1 bites összeadókat a rajz kiegészítésével úgy, hogy azok 2 db 3 bites számot helyesen adjanak össze. (CI értékét ennek megfelelően írja be). **Adja meg a kimeneteiken megjelenő logikai értékeket is a beírt adatok esetére!** (2p)



E5. Megadtuk a fenti összeadó S kimenetének igazságtábláját Karnaugh táblás alakban. (3p)

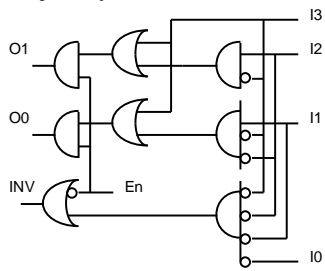
a. Írja le a függvényt megvalósító kombinációs hálózatot **Verilog nyelven**, egyszerűsítetlen **SOP** alakban!

BCi	S			
A	00	01	11	10
0	0	1	0	1
1	1	0	1	0

.....

b. Lehet-e egyszerűsíteni a függvényt, ha a megvalósításhoz csak AND, OR, INV logikai operátorokat használhatunk? Húzza alá a megfelelő választ! igen nem

E6. Az alábbi kapcsolási rajz egy ismert funkcionális elem belső felépítését mutatja. Adja meg a nevét, majd folytassa az **O kimenetének** Verilog leírását! (3p)

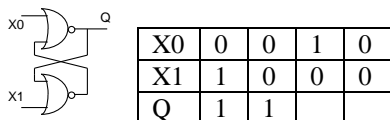


neve:.....

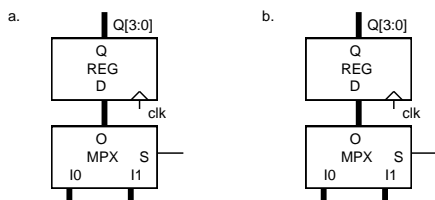
```
always (*)
if(En)
caseX (I)
4'b0001: O = 2'b00;
```

E7. a. Milyen ismert áramkör látható az alábbi kapcsolási rajzon? (1p)

b. Adja meg, milyen válaszosorozatot ad az alábbi bemeneti sorozatra az áramkör! A megoldást elkezdjük, folytassa! (2p)



E8. Egészítse ki az alábbi rajzot, az kérdéseknek megfelelően!



a. Egy engedélyezhető regisztert valósítson meg! ($L_d = 1$ tölt) (1p)

b. Egy jobbra shiftelő tölthető shiftregisztert valósítson meg! ($L_d = 1$ tölt, egyébként shiftel) (1p)

E9. Milyen funkcionális elemet adtunk meg az alábbi Verilog leírással? (1p)

wire [3:0] in;

assign ou = ((({3'b000,en} << sel) & in);

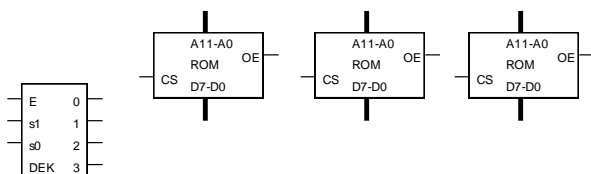
funkcionális elem neve:

E6. a. Készítsen az alábbi ROM-okból 3-szoros kapacitású egységet! (Egészítse kia a rajzot!) (1p)

Az elkészítendő egység jelei: A13-A0, D7-D0, OE, CS

b. Mekkora egy ROM modul kapacitása? (1p)

.....kibyte

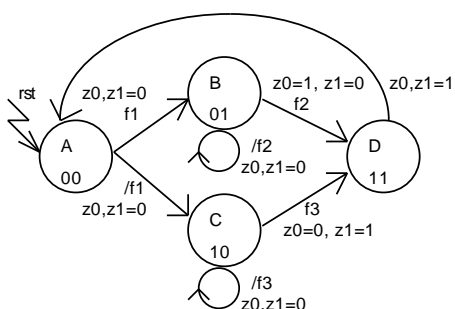


E7. Mely állítások igazak és melyek hamisak? Jelölje +-al az igaz, --al a hamis állításokat! (5p)

1.	Az always blokkban wire és reg típusú változóknak lehet értéket adni.	
2.	A bináris számlálók bármely adat kimenetén a magas és alacsony szint névlegesen ugyanannyi ideig tart.	
3.	Az EXNOR függvény páratlan számú 1-es esetén ad 1-et.	
4.	Minden szinkron sorrendi hálózat kimenete csak az órajel aktív élének hatására változhat meg.	
5.	Ha két ugyanazon órajelre működő regiszter között egy kombinációs hálózat van, akkor az ideális órajel periódusidejére (T_{clk}) igaz, hogy $T_{clk} > t_{cqmax} + t_{pmax} + t_{su}$	

Feladatok:

F1. (11p) Adott egy FSM az alábbi állapotgráffal. Állapotok: A,B,C,D,E. Bemenetek: rst, Start, f1, f2, f3. Kimenetek: z0,z1. Az állapotkódolást megadtuk. Készítse le a Verilog leírását külön az **állapotregiszter**, külön a **next_state logika** és külön a **kimeneti logika** megadásával! A next_state logikánál az always-t a kimeneti logikánál pedig assign-t használja! A leírást elkezdjük, fejezze be!



Milyen modell szerint működik? Húzza alá a megfelelőt! (1p)

Mealy Moore

//Konstans és változó deklarációk, minhárom egységhez tartozó deklarációkat ide írja! (2p):

```
localparam [1:0] A =
```

```
reg [1:0] state;
```

//Állapotregiszter (2p):

```
always@(
```

//Next_state logika (4p):

```
always@(
```

//Kimeneti logika (2p):

```
assign z0 =
```

```
assign z1 =
```

F2. Decimális számláló modul tervezés, kaszkádosítás, időmultiplexált kijelzés. (12p)

- a.** Adja meg egy decimális számláló (`dec_cnt`) Verilog modulját. A számláló tölthető (`ld`, `d[3:0]`) kétányú (`dir` = 1 fel), engedélyezhető (`en`), kimenetei: `q[3:0]` és `tc`. (6p)

- b.** Kaszkádosítson a fenti `dec_cnt` modulból 2 példányt (`dec_cntH`, `dec_cntL`). Utóbbi adja az LSB-t! A kaszkádosítást annak Verilog leírásával végezze! (4p)

- c.** *Egészítse ki a b-pontbeli leírást (alább vagy külön lapon), hogy a decimális számlálók kimenete egy 2 digités időmultiplexált kijelzőn jelenjen meg!* A 7 szegmenses kijelző digitjeinek engedélyezését a `dig0`, `dig1` jelek végzik, az adat bemenete `dat[3:0]`. A 7 szegmenses kijelző BCD kódú adatokat vár (a `BCD_7` szegmenses dekóder bele van építve). Az időmultiplexálás vezérléséhez rendelkezésére áll egy 2 bites gyűrűs számláló modul: `ring(input rst, input clk, output [1:0] Ou)`. (3p)

F3. Egy 256x8-as ROM-ban meg kell számolni a külső din bemeneten lévő adattal egyező adott adatok számát! A számolást a *start* bemenetre adott impulzus indítja. A feladat elkészültét az *rdy* kimenetén egy pulzussal jelzi, ekkor *data_count* kimenetén az adatok száma található. (12p)

a. Röviden fogalmazza meg az algoritmust! (2p)

b. Rajzolja le az adatstruktúrát vezérlő HLSM állapotdiagramját! (4p)

c. Rajzolja le a feladat adatstruktúrájának blokkvázlatát. A feladat elkészítéséhez az alábbi előre megadott modulokat használja fel (akár több példányban is). (6p)

```
ROM256x8(input [7:0] addr, output [7:0] data);           //ROM
counter256(input clk, input rst, input en, output [7:0] q, output tc); //szinkron rst-ű felfele számláló
comp(input [7:0] a, input [7:0] b, output eq);          // egyenlőség komparátor
```

Maximális pontszám: 60 pont
Rendelkezésre álló idő: 100 perc