

4.c. Ismertesse a XILINX gyártmányú SPARTAN 3E áramkör IOB (input/output block) felépítését és fontosabb feladatait! (3p)

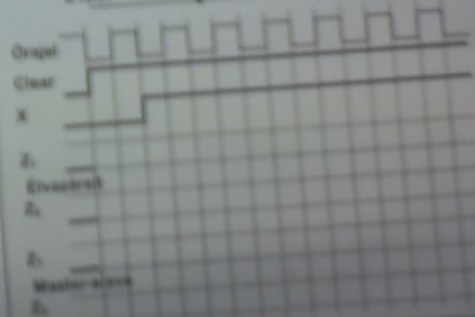
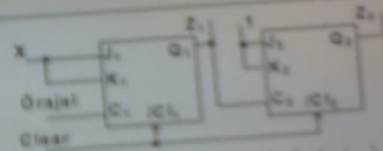
4.d. J-K flip-flopokból a mellékelt sorrendi hálózatot építették.

Jelölje meg, hogy X=1 esetén mit valósít meg a hálózat (2p)

- kétbites aszinkron számláló
- kétbites aszinkron számláló
- kétbites léptető regiszter
- egyik sem

Rajzolja be a mellékelt ábrába a Z1, Z2 kimeneti jelmezőket, ha a flip-flop felfutó él vezérlésű! (2p)

Rajzolja be a mellékelt ábrába a Z1, Z2 kimeneti jelmezőket, ha a flip-flop master-slave működésű! (2p)



2.a. Egy kétbemenetű (X_1, X_2), egy kimenetű (Z) sorrendi hálózat kimenete 0, ha X_1 bemenete 0.
 A kimenet 1-re változik, ha $X_1 = 1$ alatt X_2 bemenet 0-ról 1-re vált. Minden más esetben a kimenet változatlan.

Adja meg a fenti leírásnak megfelelően működő **aszinkron** sorrendi hálózat **előzetes** állapotábráját! (3 p)
 Adja meg a fenti leírásnak megfelelően működő **szinkron Mealy** sorrendi hálózat **előzetes** állapotábráját! (3 p)
 Adja meg a fenti leírásnak megfelelően működő **szinkron Moore** sorrendi hálózat **előzetes** állapotábráját! (3 p)

aszinkron					
y	X_1, X_2	00	01	11	10

szinkron Mealy					
y	X_1, X_2	00	01	11	10

szinkron Moore					
y	X_1, X_2	00	01	11	10

2.b. Adott egy szinkron sorrendi hálózat állapotábrája.
 Adja meg a hálózat lehetséges helyettesítési tulajdonságú particióit (önfüggő szekunder változó csoportok) beleértve a triviális particiókat is. (2p)

X_1, X_2	00	01	11	10
Y				
A	A,1	C,1	A,1	B,1
B	B,1	B,1	A,1	B,1
C	C,0	B,0	A,0	B,0

Kódolja az állapotokat a minimális számú szekunder változót igénylő, a triviálistól eltérő HT partició szerint (adja meg az állapot kódokat.), és jelölje meg, hogy melyik szekunder változót használta önfüggő szekunder változóként. (2p)

A választott állapotkóddal adja meg a kódolt állapotábrát (2p)

XILINX gyártmányú SPARTAN 3E áramkör IOB (input/output block) felépítését és fontosabb feladatait! (3p)

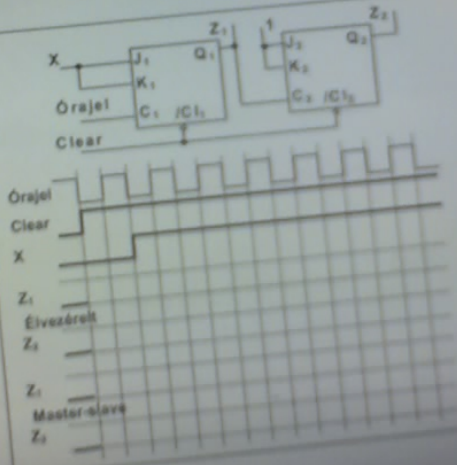
flopokból a mellékelt sorrendi hálózatot építettük.

meg, hogy $X=1$ esetén mit valósít meg a hálózat (1p):

- kétbites szinkron számláló
- kétbites aszinkron számláló
- kétbites léptető regiszter
- egyik sem

Írja be a mellékelt ábrába a Z_1 , Z_2 kimeneti jelsorozatokat, ip-flop felfutó él vezérlésű! (2p)

Írja be a mellékelt ábrába a Z_1 , Z_2 kimeneti jelsorozatokat, ip-flop master-slave működésű! (2p)



amely X, Y 6 bites pozitív számokon
 teket végzi:

$X > Y$

$X < Y$

$X = Y$

omplemens érték.

a művelet elvégzése túlesordulást

sználja az alábbi áramköröket:

or), 74283 (összeadó)

án a lehető legkevesebb áramkört
 !

l) a működés megértéséhez

s blokkvázlatot.

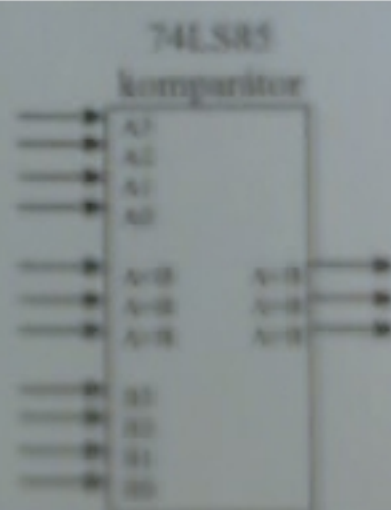
códést. (Hogyan valósítja meg a

ló szorzást. Miért nincs szükség az

ásakor multiplexert? Hogyan

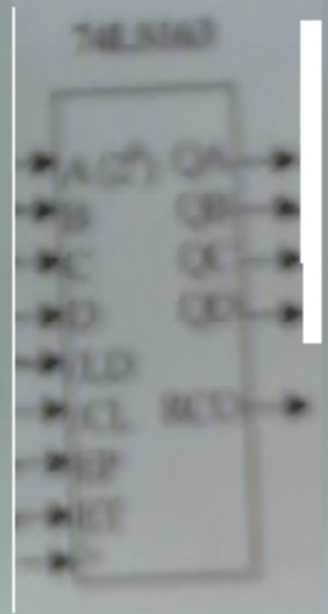
rdulást)

ramkört. (8p)



A > B				A = B		
A ₃ B ₃	A ₂ B ₂	A ₁ B ₁	A ₀ B ₀	A ₃ B ₃	A ₂ B ₂	A ₁ B ₁
00	00	00	00	0	0	0
00	00	00	01	0	0	0
00	00	00	10	0	0	0
00	00	00	11	0	0	0
00	00	01	00	0	0	0
00	00	01	01	0	0	0
00	00	01	10	0	0	0
00	00	01	11	0	0	0
00	01	00	00	0	0	0
00	01	00	01	0	0	0
00	01	00	10	0	0	0
00	01	00	11	0	0	0
00	01	01	00	0	0	0
00	01	01	01	0	0	0
00	01	01	10	0	0	0
00	01	01	11	0	0	0
00	10	00	00	0	0	0
00	10	00	01	0	0	0
00	10	00	10	0	0	0
00	10	00	11	0	0	0
00	10	01	00	0	0	0
00	10	01	01	0	0	0
00	10	01	10	0	0	0
00	10	01	11	0	0	0
00	11	00	00	0	0	0
00	11	00	01	0	0	0
00	11	00	10	0	0	0
00	11	00	11	0	0	0
00	11	01	00	0	0	0
00	11	01	01	0	0	0
00	11	01	10	0	0	0
00	11	01	11	0	0	0
00	11	10	00	0	0	0
00	11	10	01	0	0	0
00	11	10	10	0	0	0
00	11	10	11	0	0	0
00	11	11	00	0	0	0
00	11	11	01	0	0	0
00	11	11	10	0	0	0
00	11	11	11	0	0	0
01	00	00	00	1	0	0
01	00	00	01	1	0	0
01	00	00	10	1	0	0
01	00	00	11	1	0	0
01	00	01	00	1	0	0
01	00	01	01	1	0	0
01	00	01	10	1	0	0
01	00	01	11	1	0	0
01	00	10	00	1	0	0
01	00	10	01	1	0	0
01	00	10	10	1	0	0
01	00	10	11	1	0	0
01	00	11	00	1	0	0
01	00	11	01	1	0	0
01	00	11	10	1	0	0
01	00	11	11	1	0	0
01	01	00	00	1	0	0
01	01	00	01	1	0	0
01	01	00	10	1	0	0
01	01	00	11	1	0	0
01	01	01	00	1	0	0
01	01	01	01	1	0	0
01	01	01	10	1	0	0
01	01	01	11	1	0	0
01	01	10	00	1	0	0
01	01	10	01	1	0	0
01	01	10	10	1	0	0
01	01	10	11	1	0	0
01	01	11	00	1	0	0
01	01	11	01	1	0	0
01	01	11	10	1	0	0
01	01	11	11	1	0	0
01	10	00	00	1	0	0
01	10	00	01	1	0	0
01	10	00	10	1	0	0
01	10	00	11	1	0	0
01	10	01	00	1	0	0
01	10	01	01	1	0	0
01	10	01	10	1	0	0
01	10	01	11	1	0	0
01	10	10	00	1	0	0
01	10	10	01	1	0	0
01	10	10	10	1	0	0
01	10	10	11	1	0	0
01	10	11	00	1	0	0
01	10	11	01	1	0	0
01	10	11	10	1	0	0
01	10	11	11	1	0	0
01	11	00	00	1	0	0
01	11	00	01	1	0	0
01	11	00	10	1	0	0
01	11	00	11	1	0	0
01	11	01	00	1	0	0
01	11	01	01	1	0	0
01	11	01	10	1	0	0
01	11	01	11	1	0	0
01	11	10	00	1	0	0
01	11	10	01	1	0	0
01	11	10	10	1	0	0
01	11	10	11	1	0	0
01	11	11	00	1	0	0
01	11	11	01	1	0	0
01	11	11	10	1	0	0
01	11	11	11	1	0	0
01	11	11	11	1	0	0
10	00	00	00	0	1	0
10	00	00	01	0	1	0
10	00	00	10	0	1	0
10	00	00	11	0	1	0
10	00	01	00	0	1	0
10	00	01	01	0	1	0
10	00	01	10	0	1	0
10	00	01	11	0	1	0
10	00	10	00	0	1	0
10	00	10	01	0	1	0
10	00	10	10	0	1	0
10	00	10	11	0	1	0
10	00	11	00	0	1	0
10	00	11	01	0	1	0
10	00	11	10	0	1	0
10	00	11	11	0	1	0
10	01	00	00	0	1	0
10	01	00	01	0	1	0
10	01	00	10	0	1	0
10	01	00	11	0	1	0
10	01	01	00	0	1	0
10	01	01	01	0	1	0
10	01	01	10	0	1	0
10	01	01	11	0	1	0
10	01	10	00	0	1	0
10	01	10	01	0	1	0
10	01	10	10	0	1	0
10	01	10	11	0	1	0
10	01	11	00	0	1	0
10	01	11	01	0	1	0
10	01	11	10	0	1	0
10	01	11	11	0	1	0
10	01	11	11	0	1	0
10	10	00	00	0	1	0
10	10	00	01	0	1	0
10	10	00	10	0	1	0
10	10	00	11	0	1	0
10	10	01	00	0	1	0
10	10	01	01	0	1	0
10	10	01	10	0	1	0
10	10	01	11	0	1	0
10	10	10	00	0	1	0
10	10	10	01	0	1	0
10	10	10	10	0	1	0
10	10	10	11	0	1	0
10	10	11	00	0	1	0
10	10	11	01	0	1	0
10	10	11	10	0	1	0
10	10	11	11	0	1	0
10	10	11	11	0	1	0
10	11	00	00	0	1	0
10	11	00	01	0	1	0
10	11	00	10	0	1	0
10	11	00	11	0	1	0
10	11	01	00	0	1	0
10	11	01	01	0	1	0
10	11	01	10	0	1	0
10	11	01	11	0	1	0
10	11	10	00	0	1	0
10	11	10	01	0	1	0
10	11	10	10	0	1	0
10	11	10	11	0	1	0
10	11	11	00	0	1	0
10	11	11	01	0	1	0
10	11	11	10	0	1	0
10	11	11	11	0	1	0
10	11	11	11	0	1	0
11	00	00	00	0	0	1
11	00	00	01	0	0	1
11	00	00	10	0	0	1
11	00	00	11	0	0	1
11	00	01	00	0	0	1
11	00	01	01	0	0	1
11	00	01	10	0	0	1
11	00	01	11	0	0	1
11	00	10	00	0	0	1
11	00	10	01	0	0	1
11	00	10	10	0	0	1
11	00	10	11	0	0	1
11	00	11	00	0	0	1
11	00	11	01	0	0	1
11	00	11	10	0	0	1
11	00	11	11	0	0	1
11	00	11	11	0	0	1
11	01	00	00	0	0	1
11	01	00	01	0	0	1
11	01	00	10	0	0	1
11	01	00	11	0	0	1
11	01	01	00	0	0	1
11	01	01	01	0	0	1
11	01	01	10	0	0	1
11	01	01	11	0	0	1
11	01	10	00	0	0	1
11	01	10	01	0	0	1
11	01	10	10	0	0	1
11	01	10	11	0	0	1
11	01	11	00	0	0	1
11	01	11	01	0	0	1
11	01	11	10	0	0	1
11	01	11	11	0	0	1
11	01	11	11	0	0	1
11	10	00	00	0	0	1
11	10	00	01	0	0	1
11	10	00	10	0	0	1
11	10	00	11	0	0	1
11	10	01	00	0	0	1
11	10	01	01	0	0	1
11	10	01	10	0	0	1
11	10	01	11	0	0	1
11	10	10	00	0	0	1
11	10	10	01	0	0	1
11	10	10	10	0	0	1
11	10	10	11	0	0	1
11	10	11	00	0	0	1
11	10	11	01	0	0	

4.a. Rajzoljon fel 74163 bináris számláló (szinkron load és szinkron clear) felhasználásával számlálót, amely a 0 ... 9 tartományon ciklikusan számlál! A számlálót egy legalább 1 órajel periódusig fennálló START pulzussal 5-ről lehessen újra indítani! (2p)



4.b. Rajzoljon fel 74161 bináris számláló (szinkron load és aszinkron clear) felhasználásával számlálót, amely a 0 ... 9 tartományon ciklikusan számlál! A számlálót egy legalább 1 órajel periódusig fennálló START pulzussal 5-ről lehessen újra indítani! (2p)

