

1.

a.) Készítsen hálózatot, amely **X,Y** 6 bites pozitív számokon a következő műveleteket végzi:

$$Z = 2 * X + Y, \text{ ha } X > Y$$

$$Z = 2 * X - Y, \text{ ha } X < Y$$

$$Z = 2 * Y + X, \text{ ha } X = Y$$

**Z** nyolc bites kettes komplement érték.

A hálózat jelezze, ha a művelet elvégzése **túlcsordulást** eredményezett!

A megvalósításhoz használja az alábbi áramköröket: 74LS85(komparátor), 74283 (összeadó)

A megvalósítás során a lehető legkevesebb áramköri elemet használjon fel!

b.) Készítsen hálózatot, amely **X,Y** 8 bites BCD számokon BCD kódban összeadást végez. Az összeadás eredménye **Z** 8 bites BCD szám és **CO** átvitel.

A megvalósításhoz használja a 74283 (összeadó) áramkört. A megvalósítás során a lehető legkevesebb áramköri elemet használjon fel!

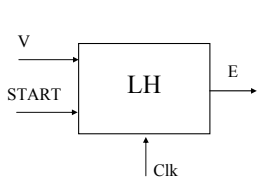
c.) Tervezze meg azt a számláló áramkört, amelynek kimenetén a bemenetre kötött órajellel ütemezve a 2, 7, 12, ..., 87 → 112, 117, ..., 202, 207 → 2, 7, ... (  $n * 5 + 2$  ) számsorozat jelenik meg.

a. Készítse el az áramkör blokkvázlatát, röviden határozza meg az egyes blokkok feladatát.

b. Tervezze meg a specifikált áramköri blokkokat.

A feladat megoldása során 4 bites párhuzamos összegzőt (74283), 4 bites szinkron **LOAD** és aszinkron **CLEAR** bemenettel rendelkező számláló áramkört (74LS161) valamint szükséges számú kapuáramkört és flip-flopot használjon fel. Törekedjen minimális áramköri megoldásra.

d.) Készítsen hálózatot, amely **SZA** és **SZB** reverzibilis számlálók működését vezérli.



A számlálók működése az órajelhez szinkronizált **START** jel megjelenésére indul. A kezdeti értékek: **SZA=4, SZB=10**.

A **V** jel (amely a **START** jellel mintavételezhető) határozza meg a számlálók számlálási irányát:

**V=0** : **SZA** felfelé, **SZB** lefelé számol

**V=1** : **SZA** lefelé, **SZB** felfelé számol

A számlálók kimeneti értékének egyezése esetén a számlálás leáll, amelyet az **E=1** kimenet jelez. Az újraindítás a **START** jellel történik.

A megvalósításhoz használja az alábbi áramköröket:

74169 reverzibilis számláló (szinkron *load*) , 74LS85 komparátor, D flip-flop.

e.) Készítsen bináris számláló áramkört, mely 4 biten **N2-től N1-ig** számol lefelé. **N1** és **N2** értékét kívülről állíthatjuk be. A megvalósítás során ügyeljen arra, hogy a megkezdett számlálási ciklus alatt **N1** és/vagy **N2** megváltozása csak a ciklus végén, azaz az új ciklusban jusson érvényre.

A megvalósításhoz használja az alábbi áramköröket: 74LS163 (számláló), 74LS374 (regiszter), 74LS85(komparátor), 7404 (inverter)

f.) **Készítsen bináris** számláló áramkört minimális külső áramkör felhasználásával, mely **4 biten (N<sub>3</sub>, N<sub>2</sub>, N<sub>1</sub>, N<sub>0</sub>; N<sub>0</sub>= 2<sup>0</sup>)** ciklikusan számol 0-tól 7-ig felfelé, majd onnan 0-ig lefelé. (A számláló kimenetén periódikusan a következő sorozat látszik: 0,1,2,3,4,5,6,7,6,5,4,3,2,1,0,1,...)

Oldja meg, hogy a számláló egy külső **RESET=1** jel segítségével bármikor nullázható legyen. A megvalósítás során ügyeljen arra, hogy minden érték csak egyetlen órajel periódusig jelenjen meg a számláló kimenetén.

A megvalósításhoz használja a mellékelt 4 bites bináris számláló áramkört, mely szinkron betöltés (**/LOAD**) és aszinkron törlés (**/CLEAR**) bemenetekkel rendelkezik.

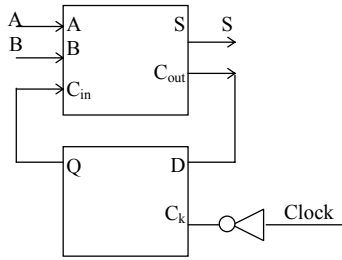
Adja meg, hogy **mikor** és **milyen** bináris értékeket kell a számlálóba tölteni?

Megoldható lenne-e szinkron törlésű (**/CLEAR**) bemenetű számlálóval a tetszőleges időpontban történő nullába állítás? Röviden indokolja a választát!

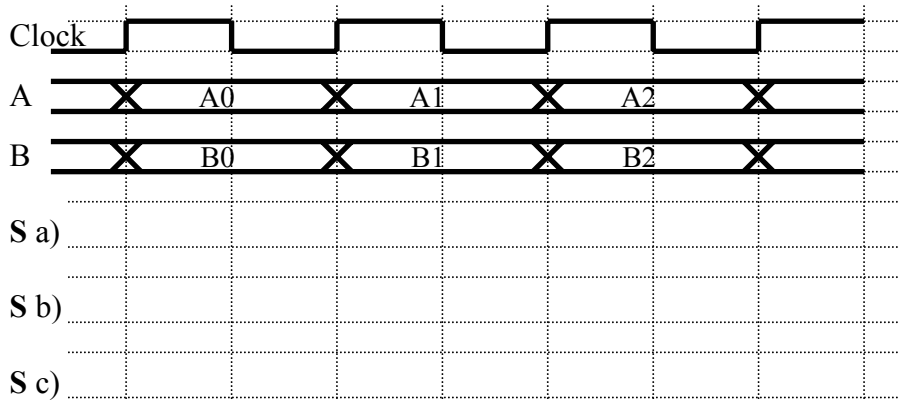
**Rajzolja fel** a teljes áramkört. Ügyeljen arra, hogy a számlálón kívül minimális számú kiegészítő áramkört használjon!

2.

- a.) Egy teljesösszeadó bemenetére a mellékelt idődiagram szerint az órajellel ütemezve érkeznek az összeadandó bitek a legkisebb helyértéktől kezdve. A következő helyértékre vonatkozó átvitelt a D-flip-flop tárolja. A bemenetekhez hasonló jelöléssel **jelölje meg** az ábrán az S0, S1, S2 eredménybitek **érvényességének időtartamát**, ha a flip-flop:



- a) felfutó élvezérelt  
 b) master-slave  
 c) data-lock-out ütemezéssű



- b.) Négybites bináris számláló áramkör (szinkron /LOAD, aszinkron /CLEAR) felhasználásával tervezzen számláló egységet, amelynek kimenetén (N0...N8, ahol N0 a legkisebb helyérték) az órajellel ütemezve a következő számsorozat jelenik meg:

$$8, 10, 12, \dots, 92, 94 \rightarrow 97, 99, \dots, 259, 261 \rightarrow 8, 10, \dots, 92, 94 \rightarrow 97, \dots$$

A számlálót egy külső RESTART=1 jellel szinkron módon 8-tól lehessen újraindítani.

**Tervezze meg** a számláló egység **legkisebb helyértékű kimenetét** (N0) előállító áramkört felfutó élvezérelt JK flip-flop felhasználásával.

**Tervezze meg** a **számláló egység** N1, ..., N8 kimeneteit előállító áramkört.

- c.) Készítsen hálózatot, amely  $X_{0..5}, Y_{0..5}$  6 bites **kettes komplementben** ábrázolt számokon a következő műveleteket végzi:

$$Z = 2 \cdot X + Y, \text{ ha } X > Y$$

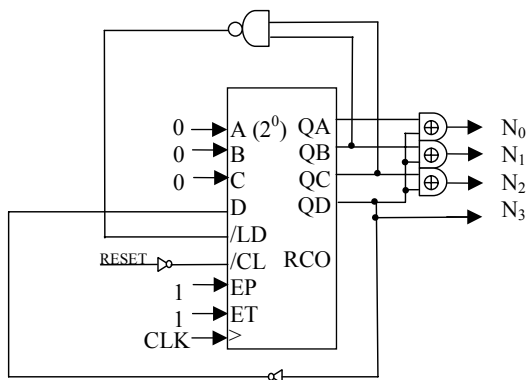
$$Z = 2 \cdot X - Y, \text{ ha } X < Y$$

$$Z = 2 \cdot Y + X, \text{ ha } X = Y$$

$Z_{0..7}$  nyolc bites **kettes komplement** érték.

**Rajzolja fel** a teljes áramkört. A megvalósításhoz használja az alábbi áramköröket: 4 bites teljes összeadó, 8 bites kettes komplementes komparátor.

- d.) 4 bites bináris számlálóval (aszinkron /CLEAR, szinkron /LOAD) az ábrán látható áramkört építették meg.



Adja meg, hogy mikor és milyen bináris értékek töltődnek a számlálóba?

Adja meg decimálisan, hogy milyen kimeneti számsorozatot állít elő az áramkör az N0...N3 kimenetein (N0 a legalacsonyabb helyiérték) egy alaphelyzetbe állító RESET pulzust követően.

- e.) Tervezzon 4 bites teljes összeadó és minimális kiegészítő áramkör felhasználásával olyan összeadó/kivonó áramkört, amely  $X(x_2, x_1, x_0)$ ,  $Y(y_2, y_1, y_0)$  3 bites előjel nélküli pozitív számokon (ahol  $x_0$ ,  $y_0$  a legkisebb helyiérték) hajtja végre az alábbi műveleteket az M vezérlő bemenet értékétől függően:

$$\text{Ha } M=0, \text{ akkor } Z = X+Y.$$

$$\text{Ha } M=1, \text{ akkor } Z = X-Y.$$

$Z(z_3, z_2, z_1, z_0)$  négybites kettes komplement kódban ábrázolt szám ( $z_0$  a legkisebb helyiérték).

2.

- f.) **Egészítsen ki** minimális kiegészítő hálózat felhasználásával egy 4 bites bináris számlálót oly módon, hogy az a 0...9 tartományban számláljon ciklikusan. A számláló szinkron /LD és aszinkron /CL bemenettel rendelkezik.
- g.) **Egészítsen ki** minimális kiegészítő hálózat felhasználásával egy 4 bites bináris számlálót oly módon, hogy az a 0...12 tartományban számláljon ciklikusan. A számláló **szinkron** /LD és **aszinkron** /CL bemenettel rendelkezik.
- h.) **Egészítsen ki** minimális kiegészítő hálózat felhasználásával egy 4 bites bináris felfele számlálót oly módon, hogy az a 12...0 tartományban számláljon ciklikusan **lefelé!** A számláló **szinkron** /LD és **aszinkron** /CL bemenettel rendelkezik.
- i.) 74283-as teljes összeadó és minimális kiegészítő áramkör felhasználásával rajzoljon fel egy összeadó/kivonó áramkört, amely  $X(x_2, x_1, x_0)$ ,  $Y(y_2, y_1, y_0)$  3 bites pozitív számokon (ahol  $x_0$ ,  $y_0$  a legkisebb helyérték) hajtja végre a műveletet. A műveletvégzést az M vezérlő bemenet értéke határozza meg.  
 Ha  $M=0$ , akkor  $Z = X+Y$ .  
 Ha  $M=1$ , akkor  $Z = X-Y$ .  
 $Z(z_3, z_2, z_1, z_0)$  négybités kettes komplementes kódban ábrázolt szám ( $z_0$  a legkisebb helyérték).
- j.) A és B két négybités 2-es komplementes kódban ábrázolt szám. Rajolja fel az  $A=B$ ,  $A < B$ ,  $A > B$  kimeneteket előállító áramkört 74LS85 (4 bites komparátor) komparátor felhasználásával.
- k.) **Rajzoljon fel** a 4 bites komparátorok felhasználásával 8 bites **kettes komplementes** kódban ábrázolt számok ( $P_{0..7}$ ,  $Q_{0..7}$ ) összehasonlítására alkalmas kapcsolást.
- l.) **Készítsen** aritmetikai áramkört, mely 7 bites kettes komplementesben ábrázolt számokkal ( $X, Y$ ) a következő műveleteket hajtja végre:  
 $Z = 2 * X - Y$  ha  $X > Y$   
 $Z = 2 * Y - X$  ha  $X < Y$   
 $Z = X$  ha  $X = Y$   
 A megvalósításhoz használja az alábbi áramköröket: 2 db 4 bites teljes összeadó, 1 db 8 bites komparátor, 2 db 8 bites 2/1 multiplexer. **Rajolja fel** a teljes áramkört!

3.

- a.) Jelölje meg, hogy az alábbi állítások mely memória típusokra jellemzők!

EPROM	STATIKUS RAM	DINAMIKUS RAM	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1 bitcella megvalósítása általában flip-floppal történik
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Csak olvasható
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Maszkprogramozott
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Periodikusan frissíteni kell
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tápfeszültség ráadása után tartalma véletlenszerű
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Utraibolya fényvel törölhető

- b.) Ismertesse a XILINX gyártmányú SPARTAN 3E áramkör IOB (input/output block) felépítését és fontosabb feladatait!
- c.) Ismertesse a XILINX gyártmányú Spartan 3 áramkör CLB egységének alapkiépítését, illetve határozza meg az ezen egységgel megvalósítható logikai hálózat korlátait.
- d.) Adja meg egy 2 bites carry-look-ahead típusú párhuzamos összegző legmagasabb helyértékén létrejövő átvitel-kimenet (C3OUT) logikai függvényét.
- e.) Rajzoljon fel 74163 **bináris** számláló (**szinkron load** és **szinkron clear**) felhasználásával számlálót, amely a **0 ... 11** tartományon ciklikusan számlál! A számlálót egy legalább 1 órajel periódusig fennálló START pulzussal **7-ről** lehessen újra indítani!
- f.) Rajzoljon fel 74161 **bináris** számláló (**szinkron load** és **aszinkron clear**) felhasználásával számlálót, amely a **0 ... 11** tartományon ciklikusan számlál! A számlálót egy legalább 1 órajel periódusig fennálló START pulzussal **5-ről** lehessen újra indítani!
- g.) 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!
- h.) Sorolja fel a legfontosabb, integrált áramkörben megvalósított memóriatípusokat és 1-1 sorban jellemezze őket.

4. a.) Illesszen 8085-ös mikroprocesszor alapú sínre (A0...A15, D0..D7,  $\overline{RD}, \overline{WR}, \overline{IO/M}, S_0, S_1, \overline{RESET/OUT}, \overline{READY}$ ) 1db **27C32** típ. (4k) EPROM és 2db **5532** típ. (4k) RAM memóriákat úgy, hogy egy K kapcsoló értékétől függően az alábbi címtartományokat fedjék le:
- K=0: (K zárt)  
 EPROM: 0000-0FFFh  
 RAM1: nincs  
 RAM2: 2800-37FFh
- K=1: (K nyitott)  
 EPROM: nincs  
 RAM1: 0000-0FFFh  
 RAM2: 2800-37FFh
- **Rajzolja fel** a memória modul memória térképét K=0 és K=1 esetben.
  - **Rajzolja fel** a memóriamodul címdekóder egységét a **K** kapcsolóval együtt **egyetlen** 74LS138 felhasználásával. A tervezés során vegye figyelembe, hogy a rendszerben más memória egység is lehet (**teljes dekódolás**)!
  - **Rajzolja fel** a READY logikát, ha az EPROM olvasáskor 0 WAIT állapotot, a RAM olvasáskor és íráskor 1 WAIT állapotot igényel! EPROM írásakor a READY logika nem ad /READY jelet
  - **Helyettesítse** a K kapcsolót egy D flip-floppal, amely 00h címmel rendelkező OUT utasítással, a D<sub>0</sub> vezeték értékére állítható! Dekódoláshoz kapu áramkört használjon! (Ne feledkezzen meg az indulási helyzetről és arról, hogy a rendszerben minden eszköznek /READY-t kell adnia!)
- b.) Illesszen 8085-ös mikroprocesszor alapú sínre (A0...A15, D0..D7,  $\overline{RD}, \overline{WR}, \overline{IO/M}, S_0, S_1, \overline{RESET/OUT}, \overline{READY}$ ) 1db **27C64** típ. EPROM és 1db **5565** típ. RAM memóriákat úgy, hogy az alábbi címtartományokat fedjék le:
1. **0000-1FFFh** EPROM
  2. **5000-6FFFh** RAM
- **Rajzolja fel** a memória memóriatérképet (azaz a memória chip-ek elhelyezkedését a processzor memória-címtartományában)!
  - **Rajzolja fel** a memóriamodul címdekóder egységét **egyetlen** 74LS138 felhasználásával. A tervezés során vegye figyelembe, hogy a rendszerben más memória egység is lehet (**teljes dekódolás**)!
  - **Rajzolja fel** az adatbusz meghajtó áramkör-vezérlő logikát!
  - **Adja meg** a RAM memória-áramkör bekötését! Ügyeljen a címvezetékek helyes bekötésére!
  - **Rajzolja fel** a **legegyszerűbb** **READY** logikát a következő paraméterek figyelembevételével: A RAM és az EPROM READY logikája 1 WAIT állapottal legyen olvasható. A RAM írásakor 0 WAIT állapot legyen.
  - **Rajzolja fel** a memóriamodul címdekóder egységének **legegyszerűbb** megvalósítását, ha tudjuk, hogy a rendszerben nincs és nem is lesz semmilyen más memória áramkör (**nem teljes dekódolás**).
  - **Rajzolja fel** az előző esetre azt a **legegyszerűbb** **READY** logikát, ami 0 WAIT állapottal működteti a memóriát.
- c.) Illesszen 8085-ös mikroprocesszor alapú sínre (A0...A15, D0..D7,  $\overline{RD}, \overline{WR}, \overline{IO/M}, S_0, S_1, \overline{RESET/OUT}, \overline{READY}$ ) 1db **27C64** típ (8KB). EPROM és 1db **5532** típ.(4KB) RAM memóriákat úgy, hogy az alábbi címtartományokat fedjék le:
- EPROM:  
 0000-07FFh  
 1400-17FFh
- RAM, egy **K** kapcsolóval állítható módon:  
 8000-8FFFh K=0 (zárt kapcsoló) esetén  
 C000-CFFFh K=1 (nyitott kapcsoló) esetén
- **Rajzolja fel** a memória modul memória térképét.
  - **Rajzolja fel** a RAM memória címdekóder egységét a **K** kapcsolóval együtt egy 74LS138 felhasználásával. A tervezés során vegye figyelembe, hogy a rendszerben más memória egység is lehet (**teljes dekódolás**)!
  - **Rajzolja fel** az adatbusz meghajtó áramkör-**vezérlő logikát** és jelölje be a meghajtón az A és B oldalt!
  - **Rajzolja fel** az EPROM memória címdekóder egységét egy 74LS138 felhasználásával. A tervezés során vegye figyelembe, hogy a rendszerben más memória egység is lehet (**teljes dekódolás**)!
  - **Rajzolja fel** a READY logikát a következő feltételek figyelembevételével  
 EPROM: olvasáskor 1 WAIT állapot, íráskor nincs READY adás;  
 RAM: olvasáskor és íráskor 2 WAIT állapot.

4.

- d.) Illesszen 8085-ös mikroprocesszoron alapuló sínre ( $\overline{RD}, \overline{WR}, IO/\overline{M}, S0, S1, \overline{INTA}, \overline{READY}, A15..A0, D7..D0, CLK$ ) i2732 típusú EPROM, illetve TC5565 típusú RAM memóriák felhasználásával memóriamodult, mely 4KB EPROM-ot és 16KB RAM-ot tartalmaz. A memóriák a következő címtartományokat foglalják el:

**EPROM:** 0C00H - 1BFFH,  
**RAM:** A000H - DFFFH.

A modul az EPROM olvasásakor 0 wait állapotot, a RAM írása/olvasása-kor 1 wait állapotot kérjen. A tervezéskor tétellezze fel, hogy a rendszerben a 0000-7FFFH címtartományban a fenti EPROM-on kívül nincs és nem is lesz más memória.

- **Adja meg** a RAM címdekódoló áramkörét a szükséges helyen 74LS138 felhasználásával (teljes címdekódolás) és az EPROM címdekódoló áramkörének legegyszerűbb realizációját (teljes címdekódolás).
- **Rajzolja fel** a READY áramkört.
- **Rajzolja fel** a memóriák buszmeghajtó áramköreit és vezérlésüket. **Adja meg** a memória áramkörök bekötését!

- e.) Illesszen 8085-ös mikroprocesszoron alapuló sínre ( $\overline{RD}, \overline{WR}, IO/\overline{M}, S0, S1, \overline{INTA}, \overline{READY}, A15..A0, D7..D0, CLK$ ) i2732 típusú EPROM, illetve TC5565 típusú RAM memóriák felhasználásával memóriamodult, mely 4KB EPROM-ot és 16KB RAM-ot tartalmaz. A memóriák a következő címtartományokat foglalják el:

**EPROM:** BC00H - CBFFH,  
**RAM:** 4000H - 7FFFH.

A modul az EPROM olvasásakor 0 wait állapotot, a RAM írása/olvasása-kor 1 wait állapotot kérjen. A tervezéskor tétellezze fel, hogy a rendszerben a 0000-7FFFH címtartományban a fenti RAM-on kívül nincs és nem is lesz más memória.

- **Adja meg** a RAM címdekódoló áramkörét a szükséges helyen 74LS138 felhasználásával (teljes címdekódolás) és az EPROM címdekódoló áramkörének legegyszerűbb realizációját (teljes címdekódolás).
- **Rajzolja fel** a READY áramkört.
- **Rajzolja fel** a memóriák buszmeghajtó áramköreit és vezérlésüket. **Adja meg** a memória áramkörök bekötését!

- f.) Illesszen 8085-ös mikroprocesszoron alapuló sínre ( $\overline{RD}, \overline{WR}, IO/\overline{M}, S0, S1, \overline{INTA}, \overline{READY}, A15..A0, D7..D0, CLK$ ) i2732 típusú EPROM, illetve TC5565 típusú RAM memóriák felhasználásával memóriamodult, mely 4KB EPROM-ot és 16KB RAM-ot tartalmaz. A memóriák a következő címtartományokat foglalják el:

**EPROM:** 3C00H - 4BFFH,  
**RAM:** B000H - EFFFH.

A modul az EPROM olvasásakor 1 wait állapotot, a RAM írása/olvasása-kor 0 wait állapotot kérjen. A tervezéskor tétellezze fel, hogy a rendszerben a 0000-7FFFH címtartományban a fenti EPROM-on kívül nincs és nem is lesz más memória.

- **Adja meg** a RAM címdekódoló áramkörét a szükséges helyen 74LS138 felhasználásával (teljes címdekódolás) és az EPROM címdekódoló áramkörének legegyszerűbb realizációját (nem teljes címdekódolás).
- **Rajzolja fel** a READY áramkört.
- **Rajzolja fel** a memóriák buszmeghajtó áramköreit és vezérlésüket. **Adja meg** a memória áramkörök bekötését!

- g.) Illesszen 8085-ös mikroprocesszoron alapuló ( $\overline{RD}, \overline{WR}, IO/\overline{M}, S0, S1, \overline{INTA}, \overline{READY}, A15..A0, D7..D0, CLK$ ) sínre i2716 (2 KB) típusú EPROM, illetve TC5565 típusú RAM (8 KB) memóriák felhasználásával memóriamodult, mely 4 KB EPROM-ot és 4 KB RAM-ot tartalmaz. A memóriák a következő címtartományokat foglalják el:

**EPROM:** 0000H - 0FFFH,  
**RAM:** 8000H - 8FFFH.

- **Adja meg** az EPROM címdekódoló áramkörét a szükséges helyen 74LS138 felhasználásával (teljes címdekódolás)!
- **Adja meg** a RAM címdekódoló áramkörének legegyszerűbb realizációját, ha tudjuk, hogy a rendszerben a 8000-FFFFH címtartományban a fenti RAM-on kívül nincs és nem is lesz más memória. (nem teljes címdekódolás)!
- **Rajzolja fel** a READY áramkört 74LS74 felhasználásával úgy, hogy a modul az EPROM **olvasásakor** 1 wait állapotot, a RAM **írása/olvasása-kor** 0 wait állapotot kérjen.
- **Rajzolja fel** a memóriák buszmeghajtó áramkörének vezérlését. Adja meg a memória áramkörök bekötését! Ügyeljen az egyes jelek elnevezésére! (az azonos nevű jelek összekötöttnek tekinthetők)

4.

**h.)** Illesszen 8085-ös mikroprocesszoron alapuló ( $\overline{RD}, \overline{WR}, IO/\overline{M}, S0, S1, \overline{INTA}, \overline{READY}, A15..A0, D7..D0, CLK$ ) sínre i2716 (2 KB) típusú EPROM, i2764 (8kb) EPROM, illetve TC5565 típusú RAM (8 KB) memóriák felhasználásával memóriamodult, mely összesen 10 KB EPROM-ot és 8 KB RAM-ot tartalmaz. A memóriák a következő címtartományokat foglalják el:

**EPROM1:** 0000H-07FFH,  
**RAM:** E000H - FFFFH,  
**EPROM2:** 8000H-9FFFH

- **Adja meg** az EPROM1 címdekódoló áramkörének legegyszerűbb realizációját, ha tudjuk, hogy a 0000h-7FFFh tartományban nincs és nem is lesz más memória áramkör (nem teljes címdekódolás).
- **Adja meg** a RAM és az EPROM2 közös címdekódoló áramkörének legegyszerűbb realizációját a szükséges helyen 74LS138 felhasználásával (teljes címdekódolás)!
- **Rajzolja fel** a READY áramkört 2db 74LS74 és minimális kiegészítő hálózat felhasználásával úgy, hogy a modul az EPROM1 **olvasásakor** 1 wait állapotot, EPROM2 **olvasásakor** 2 wait állapotot, a RAM **írása/olvasása-kor** 0 wait állapotot kérjen.
- **Rajzolja fel** a memóriák buszmeghajtó áramkörének vezérlését. Adja meg a memória áramkörök bekötését! Ügyeljen az egyes jelek elnevezésére! (az azonos nevű jelek összekötöttnek tekinthetők)

**i.)** Illesszen 8085-ös mikroprocesszoron alapuló sínre ( $\overline{RD}, \overline{WR}, IO/\overline{M}, S0, S1, \overline{INTA}, \overline{READY}, A15..A0, D7..D0, CLK$ ) i2732 (4 KB) típusú EPROM, i2764 (8KB) EPROM, illetve TC5565 típusú RAM (8 KB) memóriák felhasználásával memóriamodult, mely összesen 12 KB EPROM-ot és 4 KB RAM-ot tartalmaz. A memóriák a következő címtartományokat foglalják el:

**EPROM1:** 0000H - 0FFFH,  
**RAM:** E000H - EFFFH,  
**EPROM2:** 6000H - 7FFFH

- **Adja meg** az RAM címdekódoló áramkörének legegyszerűbb realizációját, ha tudjuk, hogy a 8000h-FFFFh tartományban nincs és nem is lesz más memória áramkör (nem teljes címdekódolás).
- **Adja meg** a EPROM1 és az EPROM2 közös címdekódoló áramkörének legegyszerűbb realizációját 74LS138 felhasználásával (teljes címdekódolás)!
- **Rajzolja fel** a READY áramkört 2db 74LS74 és minimális kiegészítő hálózat felhasználásával úgy, hogy a modul az EPROM1 **olvasásakor** 1 wait állapotot, EPROM2 **olvasásakor** 0 wait állapotot, a RAM **írása/olvasása-kor** 2 wait állapotot kérjen.
- **Rajzolja fel** a memóriák buszmeghajtó áramkörének vezérlését. Adja meg a memória áramkörök bekötését! Ügyeljen az egyes jelek elnevezésére! (az azonos nevű jelek összekötöttnek tekinthetők)

**j.)** Illesszen 8085-ös mikroprocesszoron alapuló sínre ( $\overline{RD}, \overline{WR}, IO/\overline{M}, S0, S1, \overline{INTA}, \overline{READY}, A15..A0, D7..D0, CLK$ ) i27256 (32 KB) típusú EPROM, illetve TC5565 típusú RAM (8 KB) memóriák felhasználásával memóriamodult, mely összesen 24KB EPROM-ot és 16 KB RAM-ot tartalmaz. A memóriák a következő címtartományokat foglalják el:

**EPROM:** 0000H - 5FFFH,  
**RAM1:** 8000H - 9FFFH,  
**RAM2:** C000H - D7FFFH

- **Adja meg** az RAM1 és RAM2 címdekódoló áramkörének legegyszerűbb, kapukból kialakított realizációját, ha tudjuk, hogy a 8000h-FFFFh tartományban nincs és nem is lesz más memória áramkör (nem teljes címdekódolás).
- **Adja meg** a EPROM1 címdekódoló áramkörének legegyszerűbb realizációját 74LS138 felhasználásával (teljes címdekódolás)!
- **Rajzolja fel** a READY áramkört 2db 74LS74 és minimális kiegészítő hálózat felhasználásával úgy, hogy a modul az EPROM **olvasásakor** 2 WAIT állapotot, a RAM1 és RAM2 **írásakor** 0 WAIT állapotot, és **olvasásakor** 2 WAIT állapotot kérjen.
- **Rajzolja fel** a memóriák buszmeghajtó áramkörének vezérlését. Adja meg a memória áramkörök bekötését! Ügyeljen az egyes jelek elnevezésére! (az azonos nevű jelek összekötöttnek tekinthetők)

4.

k.) Illesszen 8085-ös mikroprocesszoron alapuló ( $\overline{RD}, \overline{WR}, IO/\overline{M}, S0, S1, \overline{INTA}, \overline{READY}, A15..A0, D7..D0, CLK$ ) sínre i2764 (8 KB) típusú EPROM, illetve TC5565 típusú RAM (8 KB) memóriák felhasználásával memóriamodult, mely 5 KB EPROM-ot és 4 KB RAM-ot tartalmaz. A memóriák a következő címtartományokat foglalják el:

**EPROM:** 0000h - 13FFh,

**RAM:** 9800h - A7FFh.

- Adja meg az EPROM címdekódoló áramkörét a szükséges helyen 74LS138 felhasználásával (teljes címdekódolás)!
- Adja meg a RAM címdekódoló áramkörének legegyszerűbb realizációját, ha tudjuk, hogy a rendszerben a 8000-FFFFh címtartományban a fenti RAM-on kívül nincs és nem is lesz más memória. (nem teljes címdekódolás)!
- Rajzolja fel a **legegyszerűbb** READY áramkört 74LS74 felhasználásával úgy, hogy a modul az EPROM **olvasásakor** 1 wait állapotot, a RAM **írásakor** 0 wait és **olvasásakor** 1 wait állapotot kérjen.
- Rajzolja fel a memóriák buszmeghajtó áramkörének vezérlését. Adja meg a memória áramkörök bekötését! Ügyeljen az egyes jelek elnevezésére! (az azonos nevű jelek összekötöttnek tekinthetők)

l.) Készítse el egy Intel 8085-ös típusú mikroprocesszor ( $\overline{RD}, \overline{WR}, IO/\overline{M}, S0, S1, \overline{INTA}, \overline{READY}, A15..A0, D7..D0, CLK$ ) sínjére csatlakozó, teljes címdekódolást igénylő 2x8 kbyte-os kapacitású memória-blokk áramköri terveit. A memória működése: RESET hatására (pl. bekapcsoláskor) a 0000h - 1FFFh memóriatartományban egy 27C64-es típusú EPROM válaszol az olvasási műveletekre. Egy 48h I/O címre végrehajtott írási művelet után egy 5565 típusú RAM memória kerül az EPROM memória helyére. A következő RESET-ig a RAM lesz a kiválasztott memória.

- Tervezze meg a memória és at I/O címdekódoló áramköröket, a memória-kiválasztást tároló JK flip-flopot és ezen áramkörök belső kapcsolatát!
- Készítse el a memóriablokk áramköri terveit, feltételezve, hogy az EPROM esetében 0, a RAM esetében 1 WAIT állapot szükséges!

m.) Illesszen 8085-ös típusú mikroprocesszoron alapuló ( $\overline{RD}, \overline{WR}, IO/\overline{M}, S0, S1, \overline{INTA}, \overline{READY}, A15..A0, D7..D0, CLK$ ) sínre i27256 típusú EPROM, illetve TC5565 típusú RAM memóriák felhasználásával memóriamodult.

A memóriák részben szólnak meg:

**EPROM:** 0000h - 5BFFh, 6C00h - 7FFFh

**RAM:** 5C00h - 6BFFh címen.

A modul nem kér WAIT állapotot.

- Adja meg a modul legegyszerűbb címdekódoló áramkörét a  $\overline{READY}$  jel pontos meghajtásával együtt!
- Rajzolja fel a memóriák buszmeghajtó áramköreit és vezérlésüket. Adja meg a memória áramkörök bekötését!

n.) Illesszen 8085-ös mikroprocesszor alapú sínre

( $A0..A15, D0..D7, \overline{RD}, \overline{WR}, IO/\overline{M}, S0, S1, \overline{READY}$ )

1db 27C64 típ. EPROM és 1db 5565 típ. RAM memóriákat úgy, hogy az alábbi címtartományokat fedjék le:

**EPROM:** 0000-1FFFh

**RAM:** 4000-4FFFh

**RAM:** 6000-6FFFh

- Rajzolja fel a memória memóriatérképet (azaz a memória chip-ek elhelyezkedését a processzor memória-címtartományában)!
- **Rajzolja fel** a memóriamodul címdekóder egységét **egyetlen** 74LS138 felhasználásával. A tervezés során vegye figyelembe, hogy a rendszerben más memória egység is lehet (**teljes dekódolás**)!
- **Rajzolja fel** az adatbusz meghajtó áramkör-**vezérlő logikát**.
- Adja meg a RAM memória-áramkör bekötését! Ügyeljen a címvezetékek helyes bekötésére!
- Rajzolja fel a  $\overline{READY}$  logikát a következő paraméterek figyelembevételével: A RAM és az EPROM READY logikája 1 WAIT állapottal legyen írható/olvasható.
- Rajzolja fel a memóriamodul címdekóder egységének **legegyszerűbb** megvalósítását, ha tudjuk, hogy a rendszerben nincs és nem is lesz semmilyen más memória áramkör (nem teljes dekódolás).
- Rajzolja fel az előző esetre azt a **legegyszerűbb**  $\overline{READY}$  logikát, ami 0 WAIT állapottal működteti a memóriát.

4. o.) Illesszen 8085-ös mikroprocesszor alapú sínre (A0...A15, D0..D7,  $\overline{RD}$ ,  $\overline{WR}$ ,  $IO/\overline{M}$ , S0, S1,  $\overline{READY}$ ) 1db **27C64** típ. EPROM és 1db **5565** típ. RAM memóriákat úgy, hogy az alábbi címtartományokat fedjék le:  
 EPROM:  
**0000-0FFFh**  
**3000-3FFFh**  
 RAM, egy **K** kapcsolóval állítható módon:  
**8000-9FFFh** K=0 (zárt kapcsoló) esetén  
**A000-BFFFh** K=1 (nyitott kapcsoló) esetén
- **Rajzolja fel** a memória modul memória térképét.
  - **Rajzolja fel** a memóriamodul címdekóder egységét a **K** kapcsolóval együtt **egyetlen** 74LS138 felhasználásával. A tervezés során vegye figyelembe, hogy a rendszerben más memória egység is lehet (**teljes dekódolás**)!
  - **Rajzolja fel** az adatbusz meghajtó áramkör-**vezérlő logikát** és jelölje be a meghajtón az A és B oldalt!
  - **Adja meg** a RAM memória-áramkör bekötését! Ügyeljen a címvezetékek helyes bekötésére!
  - **Rajzolja fel** a READY logikát a következő feltételek figyelembevételével:  
 EPROM: olvasáskor 1 WAIT állapot, íráskor nincs READY adás;  
 RAM: WK kapcsolóval állíthatóan: WK kapcsoló 0 állásakor 0 WAIT állapot, WK kapcsoló 1 állásakor 1 WAIT állapot.
- p.) Illesszen 8085-ös mikroprocesszor alapú sínre (A0...A15, D0..D7,  $\overline{RD}$ ,  $\overline{WR}$ ,  $IO/\overline{M}$ , S0, S1,  $\overline{READY}$ ) 1db **27C128** típ. (16k) EPROM és 2db **5532** típ. (4k) RAM memóriákat úgy, hogy egy K kapcsoló értékétől függően az alábbi címtartományokat fedjék le:  
 K=0: (K zárt)  
 EPROM: **0000-3FFFh**  
 RAM: -  
 K=1: (K nyitott)  
 EPROM: **1000-1FFFh**  
**3000-3FFFh**  
 RAM: **0000-0FFFh**  
**2000-2FFFh**
- **Rajzolja fel** a memória modul memória térképét K=0 és K=1 esetben.
  - **Rajzolja fel** a memóriamodul címdekóder egységét a **K** kapcsolóval együtt **egyetlen** 74LS138 felhasználásával. A tervezés során vegye figyelembe, hogy a rendszerben más memória egység is lehet (**teljes dekódolás**)!
  - **Rajzolja fel** a READY logikát, ha az EPROM 0 WAIT állapotot, a RAM 1 WAIT állapotot igényel!
  - **Helyettesítse** a K kapcsolót egy D flip-floppal, amely 0F0h - 0FFh I/O címek közötti bármelyik címre történő OUT utasítással a D<sub>0</sub> vezeték értékére állítható! (Ne feledkezzen meg az indulási tranzienstről és arról, hogy a rendszerben minden eszköznek READY-t kell adnia!)

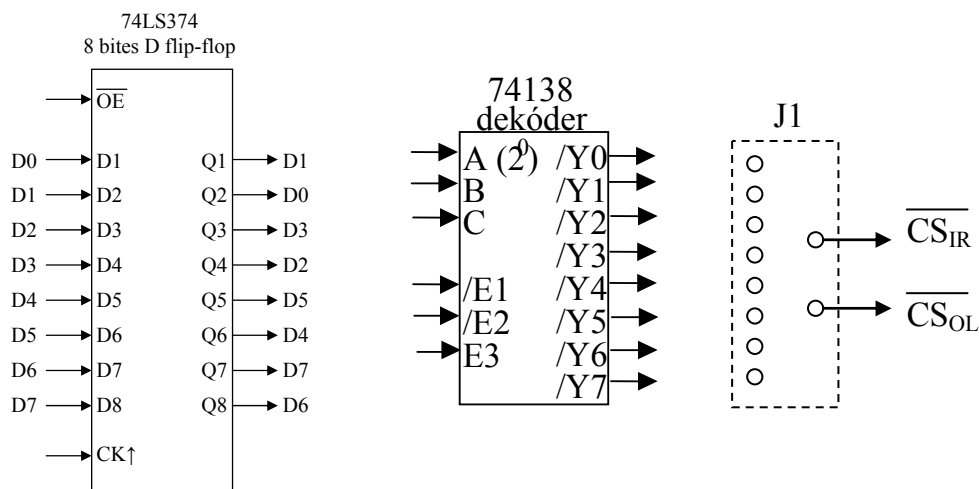


5. a.) Készítsen megszakítási vonalakat kezelő áramkört, amely egy 8085-ös processzoron alapuló sín ( $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{IO/M}$ ,  $\overline{INTA}$ ,  $\overline{READY}$ ,  $A_{15..A_0}$ ,  $D_{7..D_0}$ ,  $CLK$ ,  $ResetOut$ ,  $RST_{5.5}$ ,  $RST_{6.5}$ ,  $RST_{7.5}$ )  $RST_{5.5}$ -ös és  $RST_{6.5}$ -ös megszakítási vonalaira csatlakozva 2 db külső megszakítás (IT1 és IT2) fogadására alkalmas.

A külső megszakítások kezelését külön-külön áramkörök végzik, amelyek a külső megszakítás felfutó élére kérnek megszakítást és alaphelyzetbe OUT utasítással állíthatók (vagy a  $ResetOut$  jel hatására). Egy megszakítási impulzus csak egyszer kérjen megszakítást. Az IT1 megszakítás hatására a C regiszter 55h, IT2 hatására a C regiszter AAh értéket vesz fel.

- Készítse el a külső IT-k fogadására alkalmas áramköröket (dekódoló,  $\overline{READY}$ , megszakítási FF) A megszakítási flip-flopok 11h (IT1) és 17h (IT2) címen érhetőek el.
- Készítse el a memória első 64 byte-jában lévő ugrási táblát!
- Készítse el a rendszer **inicializálás**át végző főprogramot (40h címtől kezdődően)!
- Készítse el a megszakítási szubrutinokat!

b.) Illesszen i8085-ös mikroprocesszoros rendszersínre ( $A_{0..A_7}$ ,  $D_{0..D_7}$ ,  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{IO/M}$ ,  $S_0$ ,  $S_1$ ,  $\overline{READY}$ ) egy a megadott ábra szerinti bekötésű visszaolvasható 8 bites kimeneti regisztert.



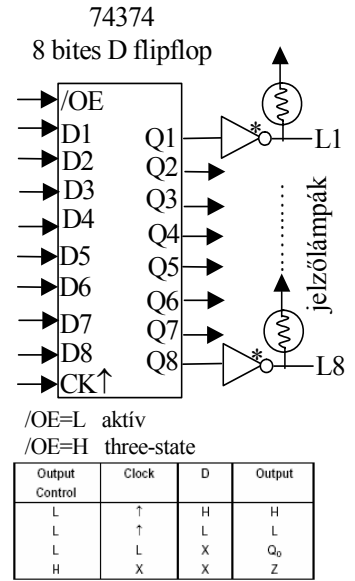
- Rajzoljon címdekódoló egységet 1db 74138-as áramkör és a J1 átkötési mező felhasználásával, úgy hogy a kimeneti regiszter 0FFh I/O címen írható és 1Fh címen olvasható.
- Készítse el az egység **legegyszerűbb**  $\overline{READY}$  áramkörét, ha a rendszerben nincs és nem is lesz másik I/O címzésű áramkör!
- Rajolja fel a 74LS374 áramkör (regiszter) ki/bemeneti illesztésének vezérlését!
- Írjon assembly **szubrutint (KONV)**, amely az A regiszterben megkapott byte-ot átírja a 74LS374-es regiszter bekötésének megfelelően. A szubrutin kimenete az átkonvertált byte az A regiszterben. (A szubrutin más regiszterek tartalmát nem ronthatja el!)
- Írjon ellenőrző 8085-ös assembly **szubrutint (ELO)**, amely az összes lehetséges kombinációval ellenőrzi a kimeneti regiszter bekötését. A szubrutin lefutása után a D,E regiszterpárban legyen a felismert hibák száma! (A szubrutin a D,E regiszterpár kivételével a regiszterek tartalmát nem ronthatja el!) A megoldásban használja az előző feladatban definiált KONV szubrutint!

5.

- c.) Illesszen i8085-ös mikroprocesszoros rendszersínre ( $A0..A7, D0..D7, \overline{RD}, \overline{WR}, IO/\overline{M}, \overline{AEN}, \overline{READY}$ ) egy visszaolvasható 8 bites kimeneti regisztert! A kimeneti regiszter minden második bitje (K0,K2,K4,K6) negáltan legyen kivezelve.
- Adja meg az egység címdekóderének logikai rajzát 1db 74138 felhasználásával, ha a kimeneti regiszter (74374) a 0AAh IO címen írható. A kimeneti egység K1...K8 jelei a 0ACh IO címen olvashatók vissza.
  - Készítse el az egység  $\overline{READY}$  logikáját, ha tudjuk, hogy a regiszterek működéséhez 0 WAIT állapot szükséges!
  - Rajzolja fel a négy inverterrel kiegészített kimeneti regisztert és a bemenet illesztését 74374 és 74244-es áramkörök felhasználásával!
  - Írja meg a VIZS 8085 assembly szubrutint, amely az A regiszterben kapott értéket beírja az előző feladatban kialakított kimeneti regiszterbe, majd a kimenet értékét visszaolvassa és ellenőrzi, hogy a beérkezett karakter helyes-e? Hiba esetén a szubrutinból való visszatéréskor Z=1, különben Z=0 legyen!
- d.) Illesszen i8085-ös mikroprocesszoros rendszersínre ( $A0..A7, D0..D7, \overline{RD}, \overline{WR}, IO/\overline{M}, \overline{AEN}, \overline{READY}$ ) két 8251-es soros periféria illesztő áramkört. Az „A” áramkör csak kimenetre, a „B” áramkör csak bemenetre van felprogramozva. Az „A” áramkör TxD kimenetét a „B” áramkör RxD bemenetére kell kötni. Mindkét áramkör TxC és RxC órajel bemeneteire CLK96 órajel van kötve. Inicializáláskor mindkét áramkör programozása: aszinkron üzemmód, 8 bites karakter, páros paritás és 2 stop bit. A „B” áramkör az /RST 5,5 sínen kér megszakítást. Az „A” áramkör báziscíme 0AAh, a „B” áramkör báziscíme 0AEh.
- Adja meg az „A” és „B” egységek címdekóderének logikai rajzát 1db 74138 felhasználásával
  - Rajzolja fel a periféria oldali illesztést, azaz kösse be az adat, a modem, az órajel, és a megszakítás jeleket. Az  $\overline{RST5.5}$  jelnél vegye figyelembe, hogy ugyan ezen a vezetéken más eszköz is kérhet megszakítást!
  - Írja meg azt a két assembly programrészletet, amely a két 8251-est felprogramozza a fenti konfigurációnak megfelelően!
  - Rajzolja fel a READY logikát egy WK kapcsolóval állíthatóan! WK kapcsoló 0 állásakor 1 WAIT állapot, WK kapcsoló 1 állásakor 2 WAIT állapot legyen. Lehet használni morze kapcsolót is.
- e.) Illesszen a 8085 mikroprocesszor sínrendszerére ( $\overline{RD}, \overline{WR}, IO/\overline{M}, \overline{INTA}, \overline{READY}, A15..A0, D7..D0, CLK$ ) 8255-ös típusú PPIO áramkört, amely a 94H, 95H, 96H, 97H portcímeket foglalja el. A PPIO áramkör B portjára egy 8 bites párhuzamos adatkimenettel rendelkező periféria csatlakozik, (tehát a B portot bemenetként kell felprogramozni) amely hand-shaking (kézfogásos) jelekkel ütemezi az adatátvitelt. Adat beolvasása esetén az áramkör kérjen megszakítást a CPU RST5.5 bemenetén. A periféria 8 adatvezetéken kívül az adattal egyidőben megjeleníti a PEVEN páros paritás jelet is. A PEVEN jel csak a hand-shaking folyamat közben áll rendelkezésre.
- Készítse el a hardware terveket (címdekodoló, adatsín meghajtás, PPIO áramkör bekötése). Vegye figyelembe, hogy a megszakítási szubrutinban megvalósított adatbeolvasás időpontjában nem biztos, hogy a PEVEN paritás rendelkezésre áll, viszont a feladathoz feltétlen szükséges. Ezért javasolt a paritás jel mintavételezése (a megfelelő hand-shaking jellel) és letárolása a PPIO áramkörön kívül, majd beolvasása az A porton keresztül (pl. PA0 bemeneten) a megszakítási szubrutinban.
  - Írjon INIC55 szubrutint, amely elvégzi a PPIO áramkör fentiek szerinti inicializálást, és az SP beállítását, ha a STACK 8F00h és 8FFFh memóriatarományban van
  - Írjon RST5.5 megszakítási szubrutint, amely beolvassa az adatot és a paritást PPIO áramkorról, ellenőrzi paritást, majd az adatot elhelyezi a 3456H című memória byte-ban, illetve paritáshiba esetén megnöveli a 3457H című memória byte tartalmát.

5. f.) Illesszen i8085-ös mikroprocesszoros rendszersínre (A0..A7,D0...D7,  $\overline{RD}$ ,  $\overline{WR}$ ,  $IO/\overline{M}$ ,  $\overline{AEN}$ ,  $\overline{READY}$ ) egy visszaolvasható 8 bites kimeneti regisztert, mely az alábbi ábrán látható lámpa vezérlő egységet képes működtetni.

- Adja meg az egység címdekóderének logikai rajzát 1db 74138 felhasználásával, ha a kimeneti regisztert (74374) a 20h IO címen írható és a kimeneti egység L1...L8 jelei a 20H IO címen visszaolvashatók. A megvalósítás során ügyeljen arra, hogy az  $\overline{AEN}$  jel L értéke esetén a DMA vezérlő hajtja a sínjeleket!
- Készítse el az egység  $\overline{READY}$  logikáját, ha tudjuk, hogy a regiszterek működéséhez 0 wait állapot szükséges.
- Rajzolja fel a kimeneti regisztert és a bemenet illesztését 74374 és 74244-es áramkörök felhasználásával
- Írja meg a CHECK 8085 assembly szubrutint, amely az A regiszterben kapott értéket beírja az előző feladatban kialakított kimeneti regiszterbe, majd a kimenet értékét visszaolvassa és ellenőrzi. Hiba esetén visszatéréskor CY=1, különben CY=0 legyen.



g.) Illesszen i8085-ös mikroprocesszoros rendszersínre (A0..A7,D0...D7,  $\overline{RD}$ ,  $\overline{WR}$ ,  $IO/\overline{M}$ ,  $\overline{AEN}$ ,  $\overline{READY}$ ) egy 8255-ös párhuzamos periféria illesztő áramkört, amely az A portján egy kimeneti perifériával (KP), a B portján egy bemeneti perifériával (BP) tud 1-es üzemmódban kommunikálni. A kimeneti periféria handshake jelei: **ODR - adat kész**, **OAC - adat elfogadva**; a bemeneti periféria handshake jelei: **IDR - input adat kész**, **IAC - input adat elfogadva** (minden jel ponált logikájú jel). A kimeneti perifériát programozott lekérdezéssel, a bemeneti perifériát pedig megszakítással szeretnénk kezelni az  $\overline{RST5.5}$  vonalon keresztül.

- Adja meg az egység címdekóderének logikai rajzát 1db 74138 felhasználásával, ha az áramkör báziscíme a 80h. A megvalósítás során ügyeljen arra, hogy az  $\overline{AEN}$  jel L értéke esetén a DMA vezérlő hajtja a sínjeleket!
- Rajzolja fel a periféria oldali illesztést, azaz kösse be az adat, a handshake, és a megszakítás jeleket. Az  $\overline{RST5.5}$  jelnél vegye figyelembe, hogy ugyan ezen a vezetéken más eszköz is kérhet megszakítást!
- Írja meg azt az assembly programrészletet, amely a 8255-öst felprogramozza a fenti konfigurációnak megfelelően! (A C port nem használt bitjeit bemenetként programozza! Ne feledkezzen meg a megszakítások tiltásáról sem!)
- Írja meg azt az  $\overline{RST5.5}$  IT rutint (**IT55**), amely beolvasson egy byte adatot a bemeneti perifériáról, elhelyezi azt a 8000h memóriacímre, és a 8001h memóriacímen elhelyezkedő byte legmagasabb helyértékű bitjét 1-re állítja. A rutin írásakor vegye figyelembe, hogy ugyan ezen a vezetéken más eszköz is kérhet megszakítást. Ezen eszközök lekezeléséhez szükséges műveleteket az **IT5KEZ** szubrutin tartalmazza! Ne feledje el visszatérés előtt az IT-t a 8255-ben letiltani!

h.) Tervezzen i8085 alapú buszrendszerhez (az ábra szerinti jelekkel) memória modul **1 db i2764-es EPROM** és **1 db TC5565-ös statikus RAM** felhasználásával. Az EPROM lassú és olvasáskor **1 WAIT**, a RAM gyors és **0 WAIT** állapotot igényel. Az egyszerűség kedvéért a sín **cím-** és **vezérlőjeleit NEM** kell leválasztani! A memória modul adatbusz meghajtóval csatlakozzon a sínre (74LS245)! Memória kiosztás:

EPROM: **0000h - 07FFh**  
**2000h - 0FFFh**  
 RAM: **0800h - 1FFFh**

Adja meg a EPROM memória-áramkör bekötését!  
**Ügyeljen a b feladatnak megfelelő (helyes) címvezetékek bekötésére**

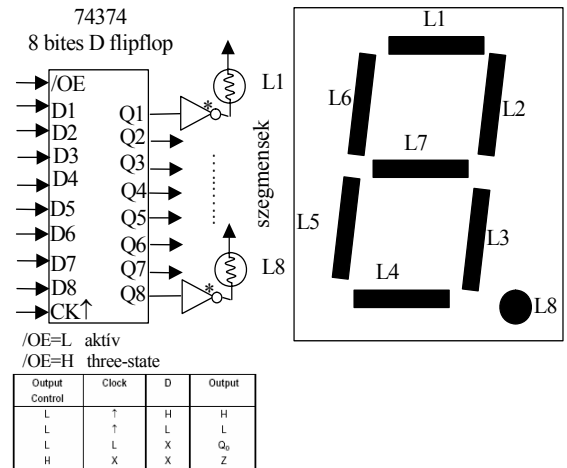
Töltse ki a memóriatérképet megadó táblázatot!

Adja meg (2 kbyte-os blokkokban), hogy a rendszerben a **0000h-07FFh**, illetve a **2000h-2FFFh** címtartományba lefordított programot hogyan kell az EPROM-ba beégetni (hogyan helyezkedik el a program az EPROM-ban)?

Memória cím						Memória
A15	A14	A13	A12	A11	A10--A0	
0	0	0	0	0	x	EPROM

**Rajzolja fel** a memóriamodul címdekóder egységét (/CE-ket előállító) **egyetlen** 74LS138 felhasználásával. A tervezés során vegye figyelembe, hogy a rendszerben más memória egység is lehet (**teljes dekódolás**)!

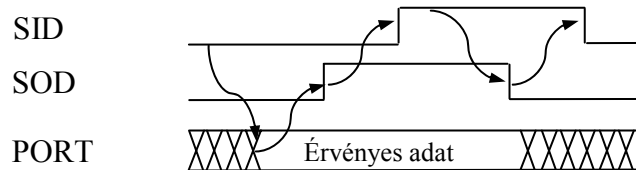
5. i.) **Illesszen** i8085-ös mikroprocesszoros rendszersínre (SA0..SA15,SD0...SD7,  $\overline{SRD}$ ,  $\overline{SWR}$ ,  $SIO / M$ ,  $\overline{SAEN}$ ,  $\overline{SRST5.5}$ ,  $\overline{SREADY}$ ,  $\overline{SRESETOUT}$ ) egy 8 bites latch-elt bemeneti perifériát 74LS374 felhasználásával. A bemenetekre 8db kapcsoló jele csatlakozik, amelyek zárt állapotban 0 (low) nyitott állapotban 1(high) értékűek. A kapcsoló értékeit egy nyomógomb megnyomásával lehet a latch-be beírni. Ugyanezen jel hatására a rendszer felé az  $\overline{RST5.5}$  vonalon egy megszakításkérés generálódik. A latch a **80h I/O** címen olvasható be. A beolvasás egyben törli az  $\overline{RST5.5}$  IT kérés jelet is, amelyet az  $\overline{SRESETOUT}$  jel is töröl. A **80H I/O** címen egy 8 bites kimeneti latch (74LS374) legyen írható. Ezek a kimenetek egy hétszegmenses kijelzőt hajtanak meg az alábbi ábrán látható módon. A kimeneti latch áramkört és vezérlését nem kell megterveznie.



**Adja meg** az egység címdekóderének **legegyszerűbb** logikai rajzát 1db 74138 felhasználásával úgy, hogy feltételezheti, hogy a **80h-87h** tartományban **nincs és nem is lesz** más I/O egység. A megvalósítás során ügyeljen arra, hogy az  $\overline{AEN}$  jel L értéke esetén a DMA vezérlő hajtja a sinjeleket! Rajzolja fel a  $\overline{SREADY}$  áramkört, amely nem kér wait állapotot. **Rajzolja fel** a nyomógomb pergesmentesítését NAND kapuk felhasználásával. **Rajzolja fel** az  $\overline{RST5.5}$  kérést megvalósító hálózatot és vonalmeghajtót. **Rajzolja fel** a bemeneti latch D bemeneteire csatlakozó kapcsolókat, a hozzá tartozó felhúzó elemmel! Rajzolja fel az /OE bemenetekre csatlakozó áramkört is!

6. a.) Írjon 8085-ös assembly programot, amely a processzor SOD kimenetén tetszőleges frekvenciájú és kitöltésű órajelet állít elő! A processzor egy fázisának ideje kb. 320 nsec. A SOD kimenet csak SID=1 értéke esetén változik. A feles megszakításokat induláskor maszkolni kell!
- b.) Írjon 8085 assembly szubrutint, amely a következő specifikáció szerint működik:  
 A szubrutin a SID jel felfutó élére egy adatkivitel, a lefutó élére egy adatbeolvasást végezzen a következők szerint:  
 - a szubrutin induláskor megvárja a SID jel 0=>1 átmenetét és a B regiszter tartalmát kiírja a 70h-es port címre,  
 - majd a SID jel 1=>0 átmenetére 71h port címről beolvas egy adatbajtót, amit a C regiszterbe tesz, ezután kilép a szubrutinból.  
**A szubrutin bemenete:** a kiírandó adat a B regiszterben.  
**A szubrutin kimenete:** a beolvasott adat a C regiszterben  
 A megoldás során ügyeljen arra, hogy a szubrutin a C regiszter kivételével a regiszterek és a flag-ek értékét ne változtassa meg!
- c.) **Írjon** 8085 assembly szubrutint, amely megvárja a SID jel változását (felfutó és lefutó él). A szubrutin kimenete: Z=1 felfutó él volt, Z=0 lefutó él volt! A szubrutin ne változtassa meg a regiszterpárok értékét!
- d.) Írjon 8085 assembly szubrutint, amely a HL regiszterpárban kapott kezdőcímmű és 32 byte hosszúságú adatblokkhoz 1 byte-os ellenőrző összeget számol ki, és elhelyezi a blokkot követő rekeszben. A szubrutin végén az akkumulátor az ellenőrző összeget tartalmazza! (A rutin bármely regiszter értékét elronthatja!)  
 Az ellenőrző összeg számítási algoritmus  $(\sum_{i=1..32} a_i) \text{ mod } 256$
- e.) Írjon assembly programot amely kivonja két memóriában elhelyezkedő 32 szavas memóriatömbök egyes szavait és az eredményeket visszahelyezi egy újabb memóriaterületre! A 2 byte-os szavak felső 8 bitjét a kisebb című (páros) byte, az alsó 8 bitet a következő (páratlan) cím tárolja!  
 (9000H,9001H) = (9100H,9101H) - (9200H,9201H) ... stb.  
 ■ Írja meg a kivonó programot!  
 ■ A kivonások elvégzése után a 9000H-n kezdődő 32 szavas memória terület után helyezzen el egy olyan byte-ot, amely megadja, hogy a 32 eredmény szó közül mennyi a pozitív szám feltételezve, hogy az operandusok (és az eredmény) 16 bites kettes komplementes kódban van ábrázolva.

6. f.) Írjon assembly **szubrutint** amely kivonja két, a memóriában elhelyezkedő 64 byte-os memóriatömb egyes byte-jait és az eredményeket visszahelyezi az első operandus helyére! Az operandusok kezdőcímeit a szubrutin a HL és DE regiszter-párokban kapja. A kivonások elvégzése után a rutin az BC regiszter-párban kapott címtől kezdődően helyezzen el egy olyan byte-ot, amely megadja, hogy a 64 eredmény byte közül mennyi a pozitív szám (>0), egy másik byte-ot, hogy mennyi a negatív szám (<0), és egy harmadik byte-ot, hogy hány érték volt 0 (=0). (Az operandusok és az eredmény 8 bites kettes komplement kódban vannak ábrázolva.)
- Írja meg a szubrutint!
  - Írja meg azt a programrészletet, amely meghívja a szubrutint a HL=8000h, DE=8800h és BC=9000h paraméterekkel, majd ellenőrzi, hogy a 9000h címtől kezdődő 3 byte összege valóban 64.
- g.) Írjon **assembly szubrutint**, amely a HL regiszterpárban kapott címen elhelyezkedő 00h végjelre végződő **karakterstringben** megkeresi az első '1' számjegyet. (az '1' számjegy ASCII kódja 31h). Találat esetén a szubrutinból visszatéréskor a HL regiszterpár mutasson a megtalált karakter címére és CY = 0 legyen. Ha a keresett karakter nem található, akkor a szubrutin CY=1 értékkel térjen vissza. (Ilyenkor a HL regiszterpár a 00h végjelre mutasson).
- h.) Írjon **assembly szubrutint**, amely a HL regiszterpárban kapott címtől kezdődően átmásol 32 bájtot a DE regiszterpárban megadott kezdőcímen kezdődő memóriaterületre. A megoldás során ügyeljen arra, hogy a rutin ne változtassa meg a flag-ek állapotát és a BC regiszterpár értékét!
- Írjon **assembly programrészletet**, amely SID=1 esetén várákozzon SID=0-ra, majd SID=0 esetén a fenti szubrutin meghívásával másolja át 32 bájtot a 9000h memóriacímről a 9150h címre.
- i.) Írjon **assembly szubrutint**, amely a DE regiszterpárban kapott címtől kezdődő 16 bájtot a blokk végétől kezdve **visszafelé átmásol** a HL regiszterpárban megadott címen kezdődő memóriaterületre (A forrás-blokk utolsó byte-ja van a cél-blokk első helyén). A megoldás során ügyeljen arra, hogy a rutin ne változtassa meg a flag-ek állapotát és a regiszterpárok értékét!
- j.) Írjon **assembly szubrutint**, amely a DE regiszterpárban kapott címtől kezdődő 16 bájtban megszámlolja a **negatív számokat** (feltételezve, hogy az adatok 8 bites kettes komplementben ábrázolt előjeles számok), és a kapott eredményt elhelyezi a 8000h memória rekeszbe. A megoldás során ügyeljen arra, hogy a rutin ne változtassa meg a flag-ek állapotát és a regiszterpárok értékét!
- k.) Írjon **assembly szubrutint**, amely a HL regiszterpárban kapott címtől kezdődő 254 byte-os adatblokk végére kiszámít egy 2 byte-os ellenőrző szót (a blokk byte-jainak összeadásával), és elhelyezi az adatokat követően. A megoldás során ügyeljen arra, hogy a rutin ne változtassa meg a flag-ek állapotát és a regiszterpárok értékét!
- l.) Írjon **assembly szubrutint**, amely 15 byte adatot és egy byte ellenőrző karaktert tartalmazó üzenet átvitelét ellenőrzi. A szubrutin bemenete az üzenet kezdőcíme (HL-ben), kimenete legyen hibátlan átvitel esetén Z=1, hibás átvitelkor Z=0. A megoldás során ügyeljen arra, hogy a rutin ne változtassa meg a regiszterpárok (BC,DE,HL) értékét!
- m.) Írjon Intel 8085 mikroprocesszor assembly nyelvén a PORTOLV szubrutint, amely a CPU SID jelének 0 → 1 átmenetére beolvassa 64H I/O című portot és az adatot az akkumulátorban adja vissza a szubrutint meghívó programrésznek.  
Írjon a OLV32 szubrutint, amely a PORTOLV szubrutin felhasználásával beolvas 32 byte-ot és elhelyezi a memória 2000H-201FH memóriaterületében. Az OLV32 szubrutin változatlan tartalommal adja vissza a meghívó programnak a CPU regisztereket.
- n.) Írja meg a **SENDBYTE** 8085 assembly szubrutint, amely a következő specifikáció szerint működik:  
A szubrutin feladata a mellékelt ábrán specifikált hand-shake szekvencia alapján a **B** regiszterben kapott adat byte továbbítása a 20H I/O címre. A megoldás során ügyeljen arra, hogy a szubrutin a flag-ek értékét ne változtassa meg! Rajzolja fel a szubrutin mellé a rutin folyamatábráját is!



Írjon 8085 assembly kódrészletet, mely a 2345h címtől kezdődő 34 byte adatot a **SENDBYTE** szubrutin segítségével továbbítja.

6

o.) Írjon 8085 assembly szubrutint, amely a következő specifikáció szerint működik:

A szubrutin (**GETBLOCK**) feladata a 80H I/O címen elhelyezkedő 8255 párhuzamos periféria illesztő A portján 1-es üzemmódban az RST 5.5 IT-vel érkező 32 byte-os adatblokk és a hozzá tartozó 1 byte-os ellenőrző összeg beolvasása. A rutin a beolvasott adatokat a 9000h címtől kezdődően helyezze el. Az ellenőrző összeg számítási algoritmus:  $(\sum_{i=1..32} a_i) \bmod 256$ . A rutin Z=1-gyel jelezze, ha az ellenőrző összeg helyes, Z=0-val, ha hibás volt. Mindkét esetben az

akkumulátor a helyes (a számított) ellenőrző összeget tartalmazza! (A rutin bármely regiszter értékét elronthatja!)

- **Írja meg** a 8255-öst felprogramozó szubrutint (A port: 1-es üzemmód, bemenet; B port: 0-ás üzemmód, kimenet; C port: bemenet)
- **Írja meg** azt az RST 5.5-ös IT rutint, amely a 8001h címen elhelyezkedő byte legmagasabb helyértékű bitjének bebillentésével jelzi, ha beolvasott egy adatot. Az adat ilyenkor a 8000h memória címen található.
- **Írja meg** a **GETBLK** szubrutint, az RST 5.5-ös IT rutin használatával Feltételezheti, hogy a rutin hívásakor az RST5.5 IT már engedélyezve van, de ne feledkezzen meg arról, hogy inicializáláskor a megszakítást a 8255-ösben letiltottuk!

p.) Készítse el egy 8251 típusú USART-ot tartalmazó soros adó/vevő áramkör terveit és az aszinkron vételi folyamatot megvalósító szubrutin programját.

- Készítse el az áramköri terveket! Az USART áramkör a 80h és 81h port címtérületet foglalja el. Adja meg a címdekódoló hálózat (teljes címdekódolás), az USART áramkör, valamint az adatsín adó/vevő áramkör kapcsolási rajzát, feltételezve, hogy a 8251-es áramkör 0 WAIT állapotot igényel, és a soros átvitelhez a HZ9600 jelű órajelet használja fel, amelynek előállítására nem része a feladatnak!
- Írja meg az áramkört inicializáló és a vételi folyamatot engedélyező programrészt! Az átvitel paraméterei:
  - aszinkron adás és vétel
  - egyszeres Baud-rate
  - 8 adatbit, páros paritás, 1 STOP bit
  - adás letiltva, vétel engedélyezve
- Írja meg a 8251-es áramkörtől keresztül 32 adatbyte-ot ciklikus lekérdezéssel beolvasó szubrutint, amely az adatot a memória 8000h - 801Fh tartományába írja be! A szubrutinban feltételezheti, hogy a vétel során nem lép fel hiba!

r.) Írja meg azt a 0000h címen induló program részletet, amely:

- **szabadon hagyja** az RST szubrutinhívó és az RST megszakításkezelő vonalakhoz rendelt programterületet. (2p)
- **Töltse ki a 2.i. feladatban megvalósított** bemeneti periféria-kezelő áramkörhöz szükséges megszakítás (RST5.5) címét úgy, **hogy** a megszakítás érvényre jutásakor a program végrehajtás a **KBEOL** megszakítás-kezelő szubrutinra kerüljön. (2p)
- **Állítsa be** a stack pointer értékét úgy, hogy **az első** elmentendő adat 1FFFh-ra kerüljön. (1p)
- **Engedélyezze** az RST5.5 érvényre jutását, majd **végtelen ciklusban hívja meg a KIVITEL szubrutint. Tétélezze fel**, hogy a **KIVITEL** szubrutint másik forrásfájlban írták meg és használja a szintén ott definiált **DISP** változót (használja a megfelelő direktívá(ka)t)!

s.) **Írjon egy hexadecimális – 7szegmens** konverziós szubrutint (**CONV**), mely az **C** regiszterben kapott bájt alsó 4 bitje által meghatározott hexadecimális számhoz hozzárendel egy-egy bájtot amely a 2.i feladat szerinti kijelző egység meghajtásához szükséges. Az egyes hexadecimális számokhoz tartozó, kijelzendő 7-szegmens értékeket a **TABL** (0800h) memóriacímtől kezdődő 16 byte-os táblázatban találja. A szubrutin a visszatérési értéket az **C** regiszterbe tegye. A rutin a **C** regiszteren kívül más regiszter értékét ne változtassa meg!

t.) **Írja meg a KBEOL** IT szubrutint, amely a 2.i. feladat szerinti bemeneti perifériából beolvassa 8 bites bináris értéket (8 kapcsoló, K1 a legkisebb helyérték). A beolvasott érték alsó 4 bitjének hexadecimális értékét átalakítja 7-szegmens formátumra a **CONV** szubrutin segítségével. Az eredményt adja át a **DISP** változónak! A megszakítás rutin ne rombolja a regiszterek értékeit.

u.) **Írjon** assembly programot, amely esemény figyelést végez. A program a SOD vonalon egy impulzussal jelzi, az esemény bekövetkezését (min. 5 $\mu$ s – max. 8 $\mu$ s), a B regiszterben pedig számolja az eseményeket. A mikroszámítógéphez a SID és a lemaszkolt RST5.5 vonalakon egymástól függetlenül egymástól tetszőleges távolságra tetszőleges szélességű impulzusok érkeznek. Az impulzusok távolsága és szélessége milliszekundumos nagyságrendbe esik. A programnak mintavételezni kell a SID és RST5.5 vonalakon érkező jeleket. Ha két mintavétel között a **SID** vonalon egy **felfutó-** és az **RST5.5** vonalon egy **lefutó** él érkezett, akkor az eseményt jelent. Az előbbieken alapján ezt jelezni és számlálni kell. A B regiszterben az esemény számlálás 0-ról indul és a regiszter FFh-ra való feltöltődése után a program engedélyezi a megszakítást majd egy HLT utasításon megáll. A 8085 rendszerben az RST6.5-ös megszakítás maszkolatlan. Ehhez a vonalhoz egy pergésmentes nyomógomb csatlakozik, amely hatására megszakítás generálódik. A megszakítás törli a B regisztert és újraindítja az eseményfigyelést!

Feladat: **INIC**, **ESFIGY** és **IMP** szubrutinok megírása. **IT6RU** megszakítási szubrutin megírásakor figyelembe kell venni, hogy a megszakítás külső flip-flopját a 80h IO címre kiadott bármely adattal törölni kell.

7.

a.) Mi látható az i8085-ös mikroszámítógép buszán, ha a CPU az alábbi programrészletet hajtja végre egy RESET esemény után? Adja meg a megváltozott regiszterek és a Z flag értékeit is!

```

ORG    0000h
LXI    SP,0h
RST    7
HLT
ORG    38h
XRA    A
LXI    H,0F000h
MOV    M,A
ORA    M
JNZ    38h
RET
    
```

b.) Analizálja a mellékelt assembly programot! Adja meg, hogy a program milyen paraméterű gépi ciklusokat hajt végre a HLT utasítás eléréséig. A gépi ciklus paraméterei: címsín, adatsín tartalma, a címkombináció forrása, az adatátvitel iránya (írás/olvasás).

```

ORG    0000h
LXI    H,9100H
SPHL
LXI    D,0FFH
MVI    A,10H
CALL   RUTIN
HLT
RUTIN: OUT    30H
        DAD    D
        RET
    
```

c.) Analizálja a mellékelt assembly programot! Adja meg, hogy a program milyen paraméterű gépi ciklusokat hajt végre a HLT utasítás eléréséig. A gépi ciklus paraméterei: címsín, adatsín tartalma, a címkombináció forrása, az adatátvitel iránya (írás/olvasás).

```

ORG    0000h
LXI    H,9FFFH
SPHL
PUSH   H
CALL   RUTIN
HLT
RUTIN: LDA    0A001H
        OUT    4FH
        RET
    
```

d.) Analizálja a mellékelt assembly programot! Adja meg, hogy a program milyen paraméterű gépi ciklusokat hajt végre a HLT utasítás eléréséig. Minden gépi ciklusra adja meg a CPU által megjelenített címet, az adatsín tartalmát, az adat átviteli irányát és típusát (READ vagy WRITE művelet, I/O vagy memória művelet), valamint az A,D,E,H,L,SP,PC regiszterek tartalmát.

```

ORG    0000h
LXI    H,9100H
SPHL
LXI    D,080H
MVI    A,2H
CIKLUS: CALL  RUTIN
        DCR    A
        JNZ    CIKLUS
        HLT
RUTIN: OUT    30H
        DAD    D
        RET
    
```

e.) Mi látható az i8085-ös mikroszámítógép buszán, és mi lesz a megváltozott regiszterek értéke az egyes utasítások végrehajtása után, ha a CPU az alábbi programrészletet hajtja végre a 0000h címtől kezdődően?

```

KEZD:  ORG    0000h
        LXI    SP,0h
        LXI    H,PR
        XRA    A          PCHL
        ORG    38h
        JMP    RUTIN
PR:    LXI    D,50h
        DAD    D
        RST    7
        JMP    PR
RUTIN: MOV    M,A
        RET
    
```

7.

f.) Mi látható az i8085-ös mikroszámítógép buszán, ha a CPU az alábbi programrészletet hajtja végre?

```
ORG 0000h
LXI H,9000h
SPHL
LXI H,0100h
PCHL
HLT
CALL 0102h
```

```
ORG 0100h
MVI A,45h
LXI B,1234h
XRA C
PUSH B
XRA B
POP D
HLT
JMP 0008h
HLT
```

g.) Mi látható az i8085-ös mikroszámítógép buszán a *CALL 0102h utasítás végrehajtása után*, ha a CPU az alábbi programrészletet hajtja végre a 0000h címtől kezdődően? Adja meg a regiszterek értékeit is!

```
ORG 0000h
LXI SP,9000h
LXI H,0001h
LXI D,0203h
XRA A
CALL 0102h
HLT
```

A regiszterek értékei a 0102h cím elérésekor:

SP= . . . HL= . . . DE= . . .

A= . . . Z<sub>flag</sub>= . . .

```
ORG 0100h
MVI A,45h
LXI B,000Dh
ANA M
RZ
PUSH B
RET
XRA B
POP D
HLT
JMP 0008h
HLT
```

h.) Mi látható az i8085-ös mikroszámítógép buszán *csak a RUT1 szubrutin futása alatt*, ha a CPU az alábbi programrészletet hajtja végre a 0000h címtől kezdődően? Adja meg a regiszterek értékeit is!

```
ORG 0000h
LXI SP,0h
LXI H,8000h
XRA A
MVI M,0AAh
MOV B,A
INR B
CALL RUT1
HLT
```

A regiszterek értékei (hexadecimálisan) a 0200h cím elérésekor:

SP= . . . HL= . . . BC= . . .

A= . . . F= . . . (S,Z,0,AC,0,P,1,CY)

RUT1:  
CIKL:

UGR:

```
ORG 0200h
PUSH PSW
MOV A,M
ANI 80h
JNZ UGR
RZ
INX H
DCR B
JNZ CIKL
POP PSW
RET
```



7.

i.) Mi látható az i8085-ös mikroszámítógép buszán, ha a CPU az alábbi programrészletet hajtja végre?

```

ORG    0000h
LXI    SP,5000h
LXI    H,0500h
PCHL
NOP
HLT

ORG    0500h
MVI    A,0AAh
LXI    B,55AAh
XRA    C
PUSH   B
XRA    B
RST    1
    
```

j.) Végezze el az alábbi Intel 8085 assembly nyelvű program analízisét a HLT utasítás eléréséig: bontsa gépi ciklusokra az alábbi utasítások végrehajtását. Minden gépi ciklusra adja meg a CPU által megjelenített címet, az adatsín tartalmát, az adat átviteli irányát (READ vagy WRITE művelet), valamint a B,C,D,E,H,L,SP,PC regiszterek tartalmát.

```

RESET:  ORG    0000h
        JMP    TOVA

TOVA:   ORG    1000h
        LXI    H,8000h
        SPHL
        LXI    H,VEGE
        LXI    B,1234h
        PUSH   B
        POP    D
        CALL  RUTIN

VEGE:   HLT
RUTIN:  PCHL
    
```

k.) Végezze el az alábbi Intel 8085 assembly nyelvű program analízisét a HLT utasítás eléréséig: bontsa gépi ciklusokra az alábbi utasítások végrehajtását. Minden gépi ciklusra adja meg a CPU által megjelenített címet, az adatsín tartalmát, az adat átviteli irányát (READ vagy WRITE művelet). Az utasításokat RESET után a végrehajtás sorrendjében analizálja!

```

ORG    0000h
JMP    FOPR

ORG    38h
JMP    RUTIN

FOPR:   ORG    40H
        LXI    H,0h
        SPHL
        ANI    0
        PUSH  PSW
        POP   B
        RST   7

VEGE:   HLT
RUTIN:  LXI    H,VEGE
        LXI    B,0
        DAD   B
        PCHL
        RET
    
```

l.) Mi látható az i8085-ös mikroszámítógép buszán, ha a CPU az alábbi programrészletet hajtja végre egy RESET esemény után. Adja meg a megváltozott regiszterek értékeit is.

```

ORG    0000h
LXI    SP,8000h
XRA    A
MOV    D,A
MVI    E,11
PUSH   D
RET
MVI    A,0Fh
XRA    E
OUT    0B6h
HLT
    
```

m.) Az Intel 8085-ös mikroprocesszor az alábbi assembly programrészletet hajtja végre. Az első utasítás végrehajtása előtt a CPU regisztereinek tartalma: A=01, B=02, C=03, D=04, E=05, H=06, L=07, PC=1234H, SP=6789H, CY=Z=S=P=AC=0.

**Adja meg** a CPU regisztereinek tartalmát a HLT utasítás végrehajtásakor!

**Adja meg** a memória 6783H – 6788H címek közötti byte-jainak tartalmát a HLT utasítás végrehajtásakor

```

ORG      1234H
KEZD:   PUSH   H
        CALL   SR
VEGE:   HLT
SR:     POP    D
        POP    PSW
        PUSH   B
        PUSH   PSW
        LXI   H, 6783H
        MOV   C, M
        INX   H
        INX   H
        MOV   D, M
        POP   PSW
        POP   B
        RET
    
```

8.

a.) Az alábbi kódrészlettel a 8000h címen elhelyezkedő 500 byte-os memóriaterület modulo 256-os ellenőrző összegét szeretnénk kiszámolni, és a memória területet követő byte-ban eltárolni. A program hibásan működik. Miért?

```

LXI     H,8000h
LXI     D,500
XRA     A
ciklus: ADD   M
        INX   H
        DCX   D
        JNZ   ciklus
        MOV   M,A
    
```

b.) Egy i8085-ös mikroszámítógép megszakítási rendszerének megvalósításakor i8259A megszakítás-vezérlőt alkalmaztunk, amely báziscíme 20h. Az alábbi megszakítást úgy szeretnénk megvalósítani, hogy bármelyik másik megszakítás-bemenetre érkező kérés azonnal érvényre juthasson mielőtt maga az IRUT rutin befejeződik. A program hibásan van megírva. Miért? Mit kellene kijavítani?

```

IRUT:   PUSH   PSW
        PUSH   B
        EI
        OUT   41h
        IN    40h
        ADI   1
        OUT   40h
        MVI   B,3

        CIKL: IN    40h
        ACI   0
        OUT   40h
        DCR   B
        JNZ   CIKL
        POP   B
        MVI   A,20h
        OUT   20h
        POP   PSW
        RET
    
```

c.) Az alábbi kódrészlettel a 8000h címen elhelyezkedő 500 byte-os memóriaterület modulo 256-os ellenőrző összegét szeretnénk kiszámolni, és a memória területet követő byte-ban eltárolni. A program hibásan működik. Miért?

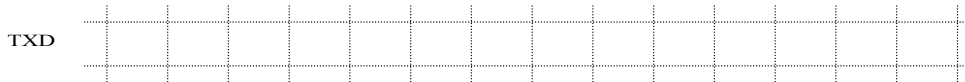
```

LXI     H,8000h
MVI     D,500
XRA     A
ciklus: ADD   M
        INX   H
        DCR   D
        JNZ   ciklus
        MOV   M,A
    
```

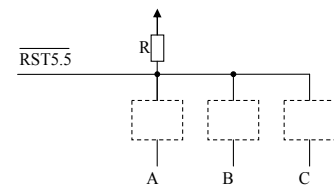
8. d.) Az alábbi kódrészlettel a 8000h címen elhelyezkedő szót (16bit) szeretnénk hozzáadni a 8150h címen elhelyezkedő szóhoz úgy, hogy az eredmény a 8000h címre kerüljön (A szó alacsonyabb helyértékű bájtja (LSB) mindig a kisebb címen helyezkedik el.). A program hibásan működik. Miért? Mit kellene kijavítani?

```
Osszead:  LXI    H,8000h
           LDA    8150h
           ADD    M
           MOV    M,A
           INX    H
           LDA    8151h
           ADD    M
           MOV    M,A
           RET
```

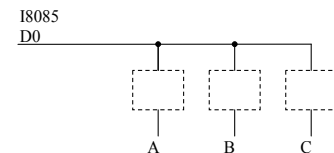
9. a.) Egy kaszkádosított megszakításkezelő rendszerben, milyen esetben kell parancsbyte-ban megadni a SLAVE megszakításkezelőknek, hogy slave áramkörök? Ebben az esetben hány ICW parancsbyte-ot kell kiadni és miért? Miért van szükség ICW3 parancsbyte-ra? Miért van szükség ICW4 parancsbyte-ra?
- b.) Inicializáljon egy 90h báziscímű i8255-ös párhuzamos periféria áramkör: A portját 1-es üzemmódban kimenetre, B portját szintén 1 üzemmódban bemenetre, C port alsó és felső felét bemenetre kell állítani. Adja meg a vezérlő szó értékét! Állítsa be a 8255 INTE ff-jainak értékét is!
- c.) Egy vegyesen kaszkádosított 8259-es megszakítási rendszerben a MASTER egység IR2 bemenetére SLAVE áramkör kapcsolódik. A MASTER egység ugrási táblájának kezdőcíme 8000h, a SLAVE ugrási táblájának kezdőcíme 9000h, mindkettő nyolcas osztásban.
- Adja meg, hogy milyen ICW3 parancsot kell küldeni a MASTER illetve a SLAVE egységeknek!
  - Adja meg, mennyi lesz a MASTER IR6 bemenetére érkező megszakítás szubrutinjának kezdőcíme!
  - Adja meg, mennyi lesz a SLAVE IR6 bemenetére érkező megszakítás szubrutinjának kezdőcíme!
- d.) Egy aszinkron üzemmódba felprogramozott 8251-es soros áramkör hibásan működik. A program státusz beolvasáskor olyan hibákat olvas, amelyek adat fogadásakor lépnek fel. Melyek lehetnek ezek, mi lehet a hibák oka ?
- e.) Mit jelent a BREAK üzemmód a 8251-es áramkörnél? Mire alkalmazható? Hogyan lehet beállítani?
- f.) Rajzolja fel egy aszinkron üzemmódban felprogramozott USART kimeneti jelalakját (TxD) 8 bites, 0Fh értékű adat elküldésekor, ha páros paritást és 1 stop bitet alkalmazunk.



- g.) Egy i8085-ös mikroprocesszoros rendszerben az A, B, C jelek külön kártyákon állnak elő, változásuk egymástól független, de mindannyian szeretnénk időszakosan megszakítást kérni a processzor RST5.5 vonalán. Milyen kimenetű kapukat kell a szaggatottal jelölt dobozokba tervezni, hogy a rendszer helyesen működjön, miért?



- h.) Egy i8085-ös mikroprocesszoros rendszerben az A, B, C jelek külön kártyákon állnak elő, változásuk egymástól független, de mindegyiket szeretnénk időszakosan címezhetően külön-külön beolvasni a processzor D0 adatvonalán. Milyen kimenetű kapukat kell a szaggatottal jelölt dobozokba tervezni, hogy a rendszer helyesen működjön, miért?



- i.) Egy 8085 mikroprocesszoros rendszerben 1 master és 4 slave 8259-es IT vezérlőt használunk. Hány megszakítást tudunk összesen lekezelni a 8259-esekkel?
- A fenti konfigurációt úgy kötötték be, hogy a slave egységek a master 0,2,4 és 6 sorszámu bemenetere kapcsolodnak. A rendszer hibásan működik. Miért? Indokolja a választ!
  - A fenti konfigurációban az összes vezérlő egy közös meghajtón (74LS245) keresztül csatlakozik a rendszersínre. Hogyan lehet ilyen esetben az egyes vezérlőknek megmondani, hogy master vagy slave szerepet játszanak?

- 9.
- j.) Egy DMA vezérlő mind a 4 csatornájára egy-egy periféria csatlakozik. Használható-e ilyenkor az 1-es csatorna AUTOLOAD üzemmódban? Indokolja a választ!
    - Programozza fel a DMA vezérlőt AUTOLOAD üzemmódba úgy, hogy a 9000h memóriacímtől kezdődő címre periódikusan 128 byte adatot legyen képes betölteni a perifériából. A 8257-es báziscíme: 70H.
  - k.) Mi különbség adatátvitel szinkronizálása szempontjából a 8251-es soros áramkör aszinkron és szinkron soros adatátvitel között?
  - l.) Programozzon fel egy 8253-as timer áramkört, amely 3,072MHz-ről 4800Hz frekvenciára osztja le jelet, 50-50%-os kitöltésűre! A 8253-as áramkör binárisan számol, és 90h báziscíme van. Használja a 0-s számlálót!
  - m.) Egy kaszkádított 8259-es rendszerben a MASTER egység IR7 és IR6 bemenetére SLAVE áramkörök kapcsolódnak.
    - Adja meg, hogy milyen ICW3 parancsot kell küldeni a Master egységnek
    - Adja meg, hogy milyen ICW3 parancsot kell küldeni Az IR6-ra csatlakozó SLAVE egységnek
    - Hogyan tudjuk kijelölni, hogy lesz ICW3 parancs?
- 10.
- a.) Sorolja fel az i8085 állapotait!
  - b.) Szerkesszen gráfot, amely az i8085-ös állapotainak kapcsolatát ábrázolja! A nyilazott ágakra az állapot-változások okait kell felírni.
  - c.) Milyen esetekben kerül a 8085-ös mikroprocesszor HALT állapotból RUN állapotba?
  - d.) Sorolja fel azokat az eseteket, amikor az INTE FF = 0 értékű lesz!
  - e.) Sorolja fel azokat az eseteket, amikor az RST7.5 FF = 0 értékű lesz!
  - f.) Az i8085-ös processzor reszettelésekor milyen értékű lesz az INTEFF, és mi lesz a PC tartalma?
  - g.) Milyen kezdőcímeiken kezdődnek az RST 2 és RST 5 utasításokkal meghívott szubrutinok, illetve az RST 5,5 és az RST 7,5 megszakítási szubrutinok?
  - h.) Mit kell kezdőértéknek az SP-be betölteni ha a programozó úgy kívánja beállítani a STACK tárat, hogy az első értékes beírt bajt 8FFFh-ra íródjon? Miért?
  - i.) Hogyan jelezzük a fordítóprogramnak, hogy a generált kódot az 1698h címtől kezdődően helyezze el?
  - j.) Hogyan jelezzük a fordítóprogramnak, hogy az 1712h címtől kezdődően helyezze el a „digit” stringet?
  - k.) Hogyan jelezzük a fordítóprogramnak, hogy inicializált vagy inicializálatlan helyfoglalás történik?
  - l.) Mi a különbség az EQU 1200h és a DW 1200h direktívák között?
  - m.) Sorolja fel az i8085 mikroprocesszor megszakítással kapcsolatos jeleit!
  - n.) Milyen típusú kimenettel rendelkező áramkörökkel lehet busz rendszerű vezetékeket meghajtani? Húzza alá, hogy melyik típusú kimenet alkalmas arra, hogy egy vezeték (busz) több kimenet egyidejűleg is meghajthasson?
  - o.) Miért nem lehet két totem-pole kimenetű áramkör kimeneteit összekötni? Indokolja a választ?
  - p.) i8255-ös párhuzamos periféria áramkör A portját 1-es üzemmódban kimenetre, B-portját 0 üzemmódban bemenetre, C port alsó felét bemenetre, felső felét kimenetre kell állítani.
    - Adja meg a vezérlő szó értékét!
    - Adja meg, melyik regiszterbe és milyen értéket kell írni, hogy a fenti üzemmód esetén a PC4 portbit 0, illetve 1 értékű legyen
  - q.) Egy 3 gépi cikusból álló utasítás (pl.: IN 83h) végrehajtása során az első gépi ciklus T1 fázisában egyidőben High(1) értékűek lesznek az INT és a HOLD bemenetek.
    - Mikor adja át a DMA vezérlőnek a 8085-ös a busz vezérlési jogot? Indokolja a választ!
    - Mikor jut érvényre a megszakítás (megszakításkérés engedélyezett állapotban van és nincs más kérés) Indokolja a választ!
  - r.) Sorolja fel azokat az eseményeket és beavatkozásokat, amelyek tiltó, illetve engedélyező állapotba billentik a 8085-ös mikroprocesszor INTE FF-ját!

**BME-IIT Digitális technika II. (VIII106) - Ellenőrző feladatok V1.3.**

10. s.) Karikázza **be** az alábbi felsorolásban **azokat** a memória-áramkör típusokat, amelyek lehetnek egy 8085-ös alap rendszerben a 0000H címmel kezdődő memóriatartományban közvetlenül a bekapcsolást követő RESET jel után? Miért?
- D-RAM      EEPROM      EPROM      FLASH      OTP-ROM      RAM      ROM      S-RAM
- t.) Sorolja fel azokat az eseményeket és utasításokat, amelyek az SP megfelelő beállítása után hajthatók végre! Hova érdemes állítani az SP értékét?
- u.) Milyen módon állítható be a 8251-es áramkör soros kiviteli sebessége aszinkron átvitelt feltételezve?
- v.) Milyen feltétel nélküli vezérlés átadó utasítások találhatók az i8085 mikroprocesszorban, amelyek hatására módosul a programszámláló (PC) értéke?

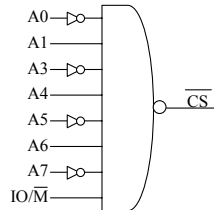
11. a.) A 8085-ös mikroprocesszorra ASM85 assembly nyelven írt alábbi programrész fordítása után adja meg a memória tartalmát!
- ```

ORG 1000H
ADAT1: DW 12
ADAT2: DB 12H
    
```
- b.) Milyen programkód keletkezik az alábbi program lefordítása után?  
Jelölje **vízszintes vonallal**, ahol nem keletkezik **programkód**.
- ```

ORG 2000h
A1h: DS 4
C1h: DB 67
      END
B59  EQU 020H
      PUBLIC B59
    
```
- c.) A 8085-ös mikroprocesszorra ASM85 assembly nyelven írt alábbi programrésznek mi a jelentése?
- d.) Mi a kezdőcíme az RST2 utasítással hívott szubrutinnak és az RST6.5-ös vezetéken érkező megszakítási szubrutinnak?
- e.) Az alábbiak közül a 8085 processzor melyik állapot-átmeneteket valósíthatja meg külső reset nélkül?
- |                |                          |                          |
|----------------|--------------------------|--------------------------|
| RUN-WAIT-RUN   | Igen                     | Nem                      |
| WAIT-HOLD-WAIT | <input type="checkbox"/> | <input type="checkbox"/> |
| HALT-HOLD-HALT | <input type="checkbox"/> | <input type="checkbox"/> |
| RUN-HALT-WAIT  | <input type="checkbox"/> | <input type="checkbox"/> |
- f.) Milyen módon **kell** egy 8255-ös PPIO áramkör MODE1-be felprogramozott PA portjához tartozó belső INTE FF-ot törölni?
- |                            |                          |                          |
|----------------------------|--------------------------|--------------------------|
| PA portra történő írással  | Igen                     | Nem                      |
| PC portra történő írással  | <input type="checkbox"/> | <input type="checkbox"/> |
| BIT RESET paranccsal       | <input type="checkbox"/> | <input type="checkbox"/> |
| BIT SET parancs kiadásával | <input type="checkbox"/> | <input type="checkbox"/> |
- g.) Az USART (8251) három adatátviteli hibát tud felismerni. Melyik az a hiba, amely **csak aszinkron** üzemmódban lehetséges?
- |                    |                          |                          |
|--------------------|--------------------------|--------------------------|
| FE (kerethiba)     | Csak aszinkron           | Szinkron és aszinkron    |
| OE (túlfutás hiba) | <input type="checkbox"/> | <input type="checkbox"/> |
| PE (paritás hiba)  | <input type="checkbox"/> | <input type="checkbox"/> |
- h.) Egy 8085-ös processzorban Reset után milyen értéket vesznek fel az alábbi regiszterek, flag-ek és flipflopok? A határozatlan értékeket jelölje X-el.
- |          |              |
|----------|--------------|
| PC       | Érték (hexa) |
| SP       |              |
| INTE ff. |              |
| CY flag  |              |

12.

- a.) Egy memória-áramkör 13 címbemenettel rendelkezik (A12,...,A0) és nyolc adat be/ki menettel (D7..D0). Adja meg az áramkör kapacitását **kilobyte-ban** és az áramkör által lefedett címtartományt **hexadecimálisan**, ha az A15, A14, A13 alábbi értéke esetén kap engedélyező (CE) jelet: A15, A14, A13 = 1 0 0.
- b.) Egy **16 bites** (D15..D0) memória-áramkör 11 címbemenettel rendelkezik (A10,...,A0). Adja meg az áramkör kapacitását **kilobyte-ban** és az áramkör által lefedett címtartományt **hexadecimálisan**, ha a kezdőcím 6000h és az alkalmazott mikroprocesszoros rendszer bajtonkénti hozzáférést (címezést) is lehetővé tesz.
- c.) Az i8085-ös mikroprocesszor sínrendszerére illesztett periféria-áramkör engedélyező/kiválasztó (/CS) bemenetét az alábbi kombinációs hálózat engedélyezi. Adja meg az áramkörhöz rendelt hexadecimális címeket!



- d.) Adja meg az alábbi dekódoló áramkör bekötéséhez tartozó engedélyező jelek címtartományát!

