Beadáskor ezt a feladatlapot a megoldáshoz csatolni kell. A feladatokat külön lapon, kézírással oldja meg. Nem fogadható el olvashatatlan, javításokat tartalmazó megoldás! Hibás megoldás javítására a pótbeadás alkalmával van lehetőség.

1. Illesszen 8085-ös mikroprocesszor alapú sínre **2716** típusú EPROM és **5565** típusú RAM memóriákat úgy, hogy az alábbi címtartományokat fedjék le:
   1. `0000h-07FFh` EPROM
   2. `2000h-27FFh` EPROM
   3. `2800h-3FFFh` RAM

   A megoldás során 1db 74LS138 dekóder áramkört használjon minimális kiegészítő hálózattal A 84h I/O címre írt adattal a RAM memória írásvédettsége változtatható legyen (ha a kiírt adat 1, akkor a RAM írásvédett, ha 0 akkor nem). Gondoskodjon róla, hogy RESET után a RAM ne legyen írásvédett.

   A sín jelei:

   $SA0...SA15, SD0...SD7, \overline{SMRD}, \overline{SMWR}, \overline{SIORD}, \overline{SIOWR}, SIO/\overline{M}, \overline{SREADY}, SS0, SS1, SClkOut, SresetOut$

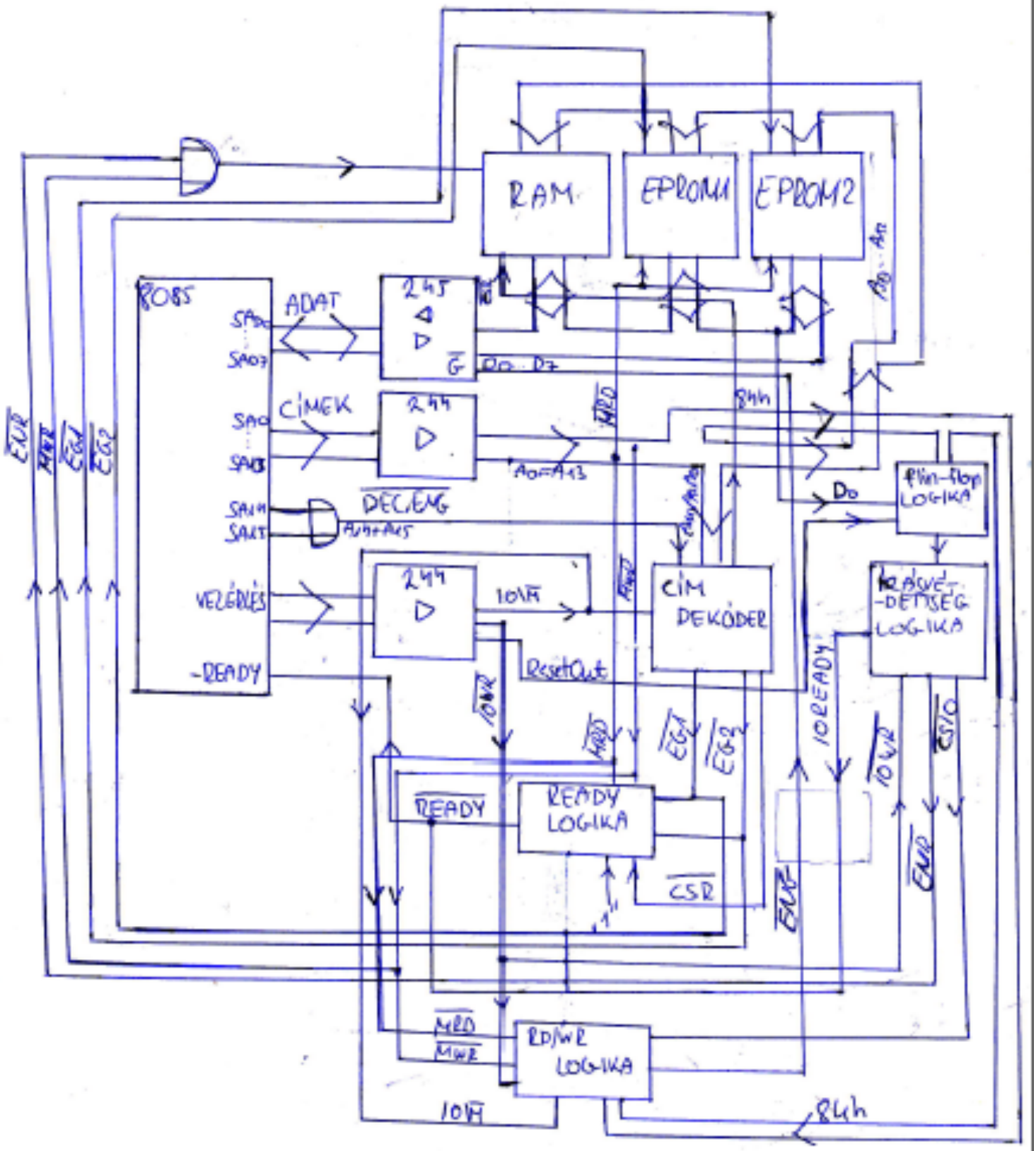
   a. Rajzolja fel a memória modul blokkvázlatát. (Figyeljen a jelek konzisztens elnevezésére!)
   b. Rajzolja fel a memóriamodul címtérképét és a címdekóder egységét.
   c. Rajzolja fel az adatbusz meghajtó áramkör-vezérlő logikát.
   d. Adja meg a memória-áramkörök bekötését!
   e. Rajzolja fel a READY logikát a következő paraméterek figyelembevételével:
      a RAM memóriák READY logikája **1** WAIT állapotot,
      az EPROM memóriák READY logikája kizárólag olvasásra **0** WAIT állapotot
      iktasson közbe a műveletvégzés közben!
   f. Tervezze meg a feladatban kért I/O egységet (dekódoló, flip-flop)!

2. Készítse el a következő assembly szubrutint, amellyel a RAM memória tesztelhető.

   Írjon ***ELLENOR*** szubrutint, amely a **DE** regiszterpárban egy kezdőcímet, a **BC** regiszterpárban egy hossz értéket kap, és első lépésben az így meghatározott memóriablokkot kitölti úgy, hogy minden byte a saját címének alsó 8 bitjét két bittel balra forgatva tartalmazza. A kitöltés után a szubrutin ellenőrizze, hogy a feltöltött memóriablokk rekeszei helyes értékeket tartalmaznak-e? A szubrutin **Z=0-val** jelezze, ha hibát talált. Ilyenkor a DE regiszterpár az első (legalacsonyabb memóriacímű) **megtalált hiba címét**, a HL regiszterpár pedig a hibásnak talált **byte-ok darabszámát** tartalmazza. Ha nincs hiba Z=1, HL=0 és DE a memóriablokk első elemére mutat. (A memóriablokk kitöltését különálló szubrutinban is megírhatja.)

   A szubrutint úgy írja meg, hogy a működéshez előírt regisztereken kívül más regiszterek értékét ne rontsa el! A szubrutint lássa el megjegyzésekkel és készítsen fejlécet is!

BLOKKVÁZLAT:



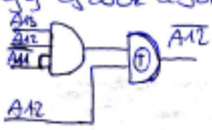
1/b)

MEMÓRIATÉRKÉP:

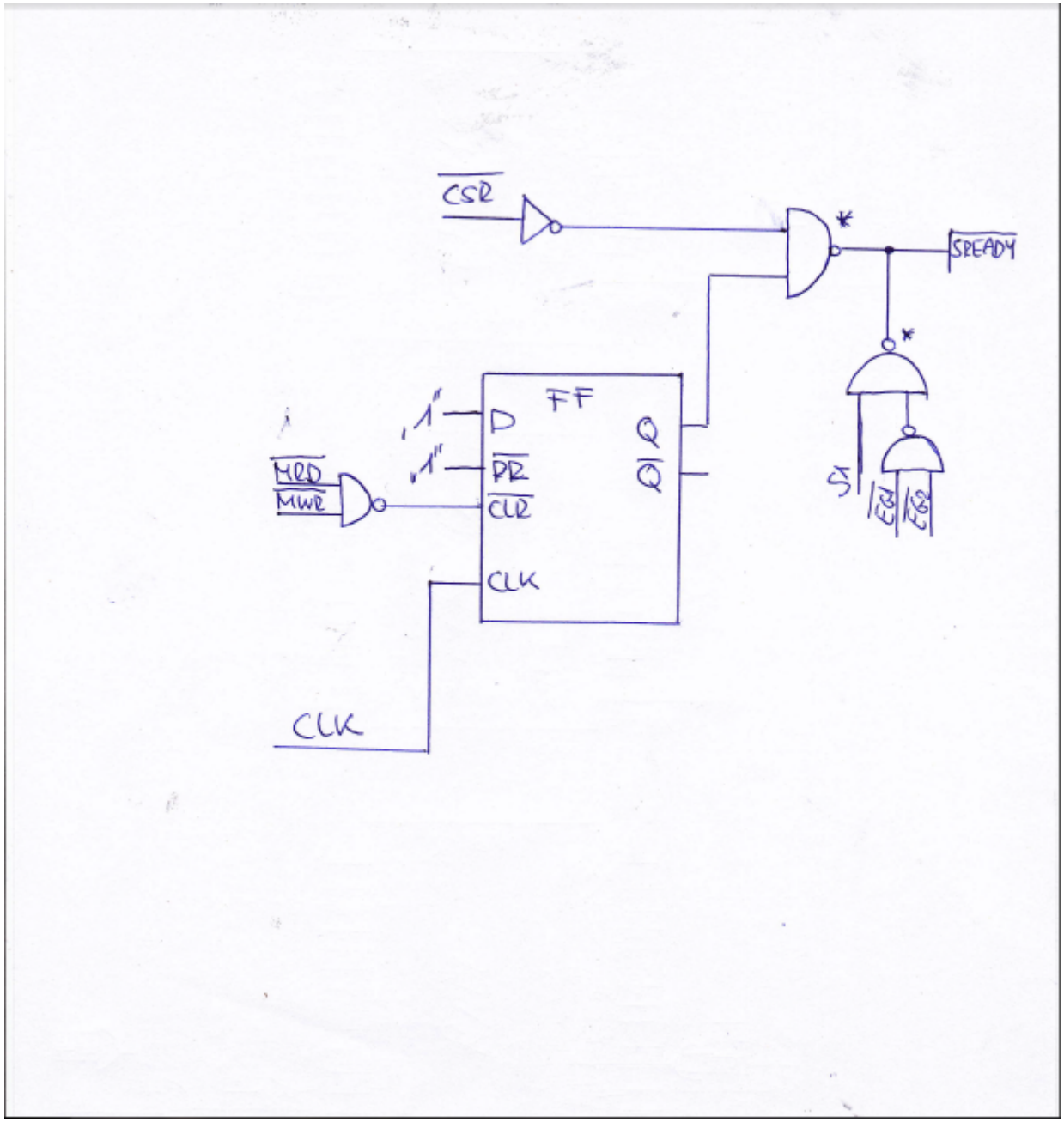
CPU	A15	A14	A13	A12	A11	A10	MEM.	
37FF 3800	0	0	1	1	$\overline{A11}$ 1	–	17FF [A0–A12] RAM 1000	RAM: 0000–07FF: $\overline{A11}$ 0800–0FFF: $\overline{A11}$, $\overline{A12}$ A0…A12 1000–17FF: $\overline{A11}$
37FF 3000	0	0	1	$\overline{A12}$ 1	$\overline{A11}$ 0	–	0FFF RAM 0800	
2FFF 2800	0	0	1	0	$\overline{A11}$ 1	–	07FF RAM 0000	
27FF 2000	0	0	1	0	0	–	7FF [A0–A10] EPROM2 000	EPROM2 A0…A10
1FFF 1800	0	0	0	1	1	–	1FFF ∅ 1800	KIHASZNÁLATLAN
17FF 1000	0	0	0	1	0	–	17FF ∅ 1000	
0FFF 0800	0	0	0	0	1	–	FFF ∅ 800	
07FF 0000	0	0	0	0	0	–	7FF [A0–A10] EPROM1 0000	EPROM1 A0…A10

Mivel a (RAM) memória címtartományának helyiértéki A0… A12 ig megy, az illesztés a fenti táblázatban akkor jó ha a RAM első és utolsó blokkcímzésénél az A11 es címvezeték negált, 0800–0FFF-ig viszont az A11 és A12 is negált.
Ez elérhető vezérelhető XOR kapuval, ha kikapuzzuk úgy hogy abban a tartományban negálja A12-őt.

Pl:

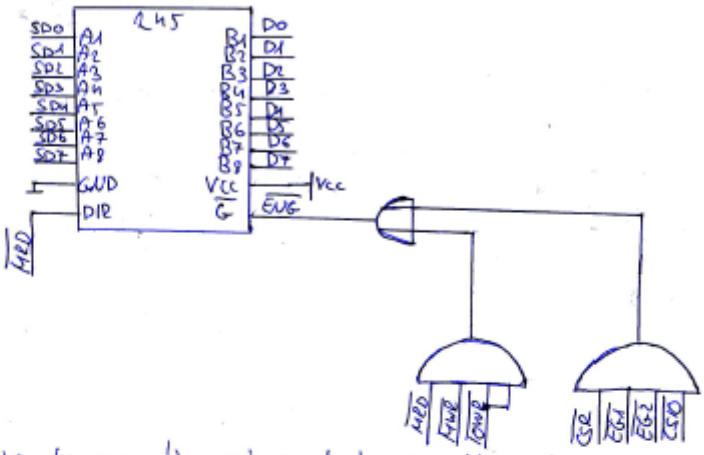


$\overline{A12}$ ha a CPU 3000–37FF ig címez

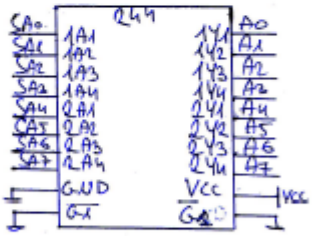

CSR
SREADY
FF
"1"
"1"
D
PR
CLR
CLK
Q
Q
MRD
MWR
CLK
S1
EQ1
EQ2

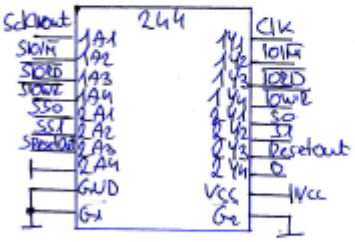
1/c)

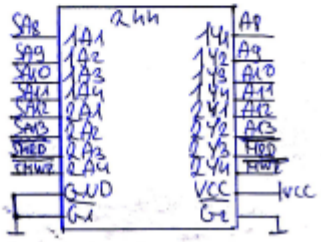
Adatleválasztás és adaterősítés:

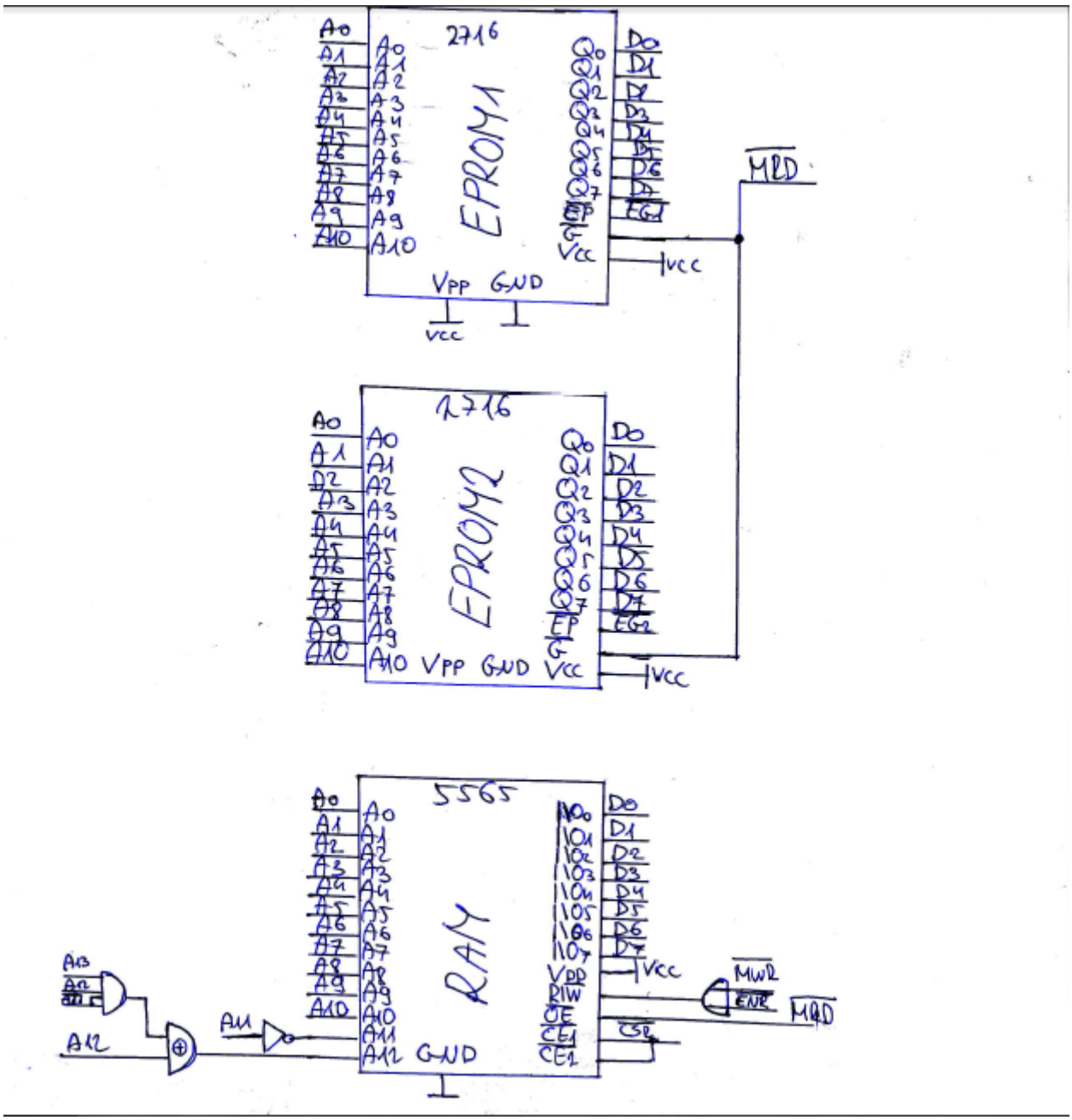


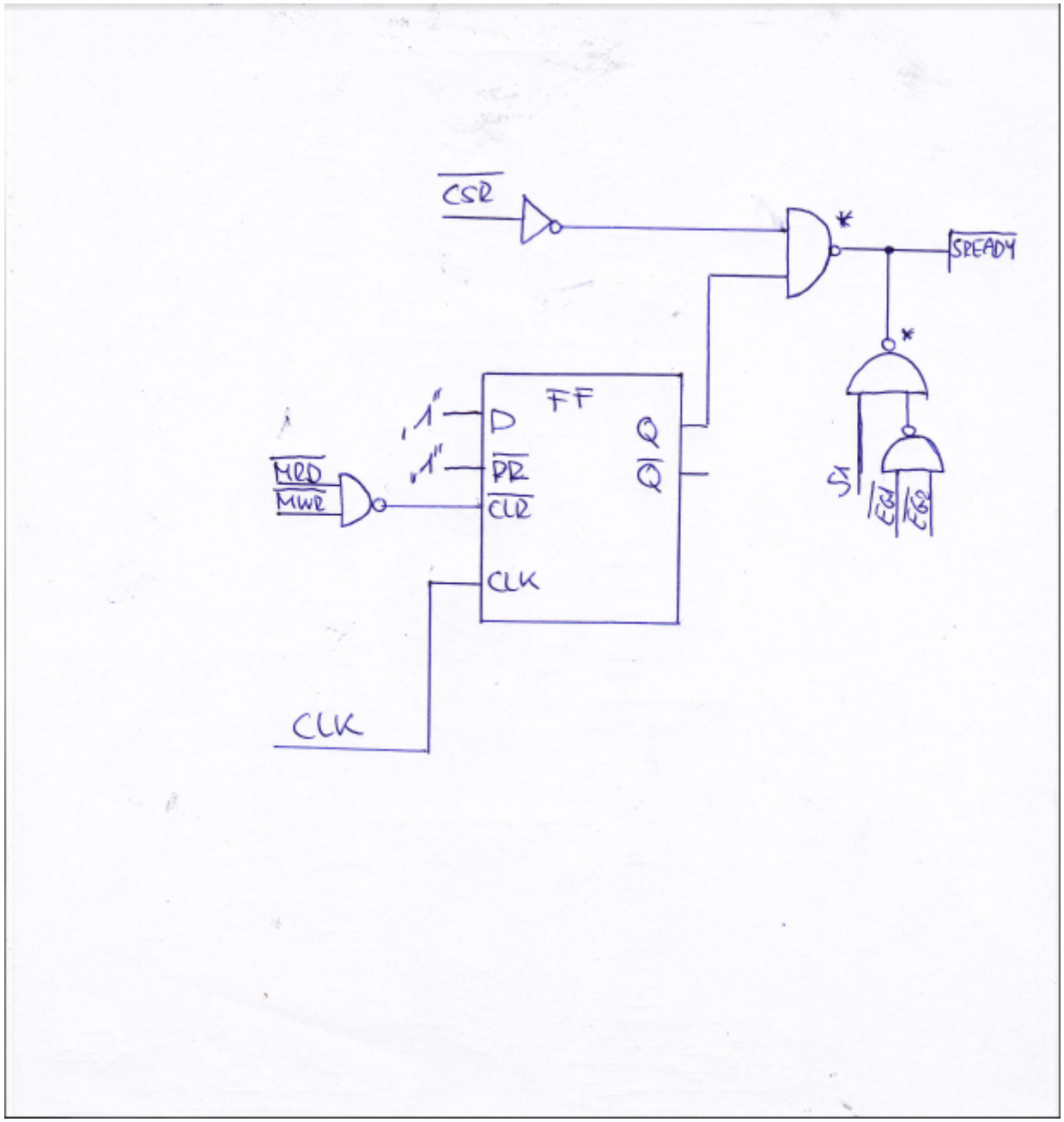
Címek és egyéb jelek leválasztásai:

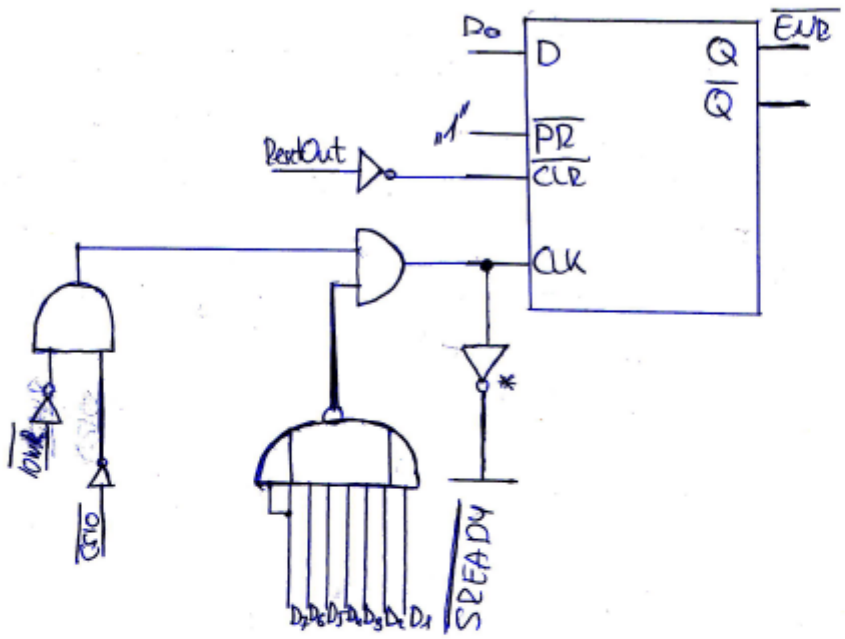




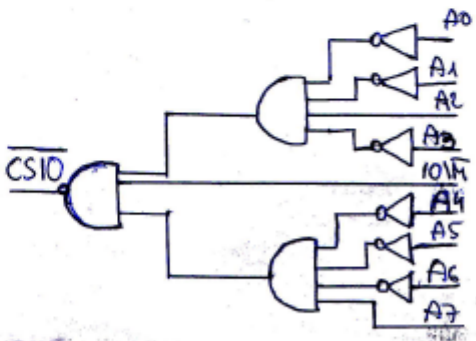



2716
EPROM1
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
EP
G
Vcc
VPP
GND
D0
D1
D2
D3
D4
D5
D6
D7
MRD
Vcc
2716
EPROM2
5565
RAM
A11
A12
GND
MWR
ENR
CS


CSR
SREADY
FF
"1"
"1"
D
PR
CLR
CLK
Q
Q
MRD
MWR
CLK
S1
EQ1
EQ2


D0
D
Q
Q
EVR
"1"
PR
CLR
ResetOut
CLK
*
CS10
D7 D6 D5 D4 D3 D2 D1
SREADY

CS10:


A0
A1
A2
A3
IOM
A4
A5
A6
A7
CS10

ASSEMBLY

2)
```
;XXXXXXXXXXXXX
; ELLENŐR SZUBRUTIN
; DE: kezdőcím
; BC: hossz
; Ha hincs hiba: Z=1, HL: hibaszám, DE: első hiba címe
; Ha minden jó: Z=0, HL:0, DE: kezdőcím
;XXXXXXXXXXXXX
CALL ELLENOR
  :
ELLENOR: PUSH B ; kezdőcím és hossz mentése
         PUSH D ; 16 bites adatok
      C1: MOV A,E ; alsó 8 bit kell
         RLC ; forgatás 1x
         RLC ; fogatás még 1x
         STAX D ; beírjuk a DE által kijelölt blokkba
         DCX B ; db--
         INX D ; cím++
         MOV A,C ; tesztelés, hogy vége-e?
         ORA B ; alsó és felső 8 bit or, ha 0, kész.
         JNZ C1 ; újra a ciklus, ha nincs vége
         MVI L,0 ; számláló rásítése, 0-ról indul
         MVI H,0 ; a hibákat fogja számolni
         POP D ; újra kell a hossz és a cím
         POP B
         PUSH B ; vissza le is mentjük
         PUSH D
         PUSH H
      C2: MOV A,E ; újratesztelünk; nagye alsó 8 bit kell
         RLC ; újra 2x forgatás
         RLC
         MOV L,E
         MOV H,D ; mutasson a HL a DE tartalmára
         CMP M ; A HL által mutatott értékből vonjuk ki az akkusit
                ; Ha 0 a Z, akkor rossz, ugrunk, kil.
                ; megyünk tovább
         INX NOVELES
```

```
+OLYT1: DCX B ;hossz -- mayd teszt
        INX D
        MOV A,C
        ORA B
        JNZ C2
        POP H
        MOV A,H
        ORA L
        JNZ HIBAVAN
        POP D
        POP B
        MVI A,1
        CPI 1 ;Z==1 lesz
        RET
NOVELES: POP H
        MOV A,L
        ORA H
        JZ ELSO
FOLYT2: INX H ;hibaszámláló++
        PUSH H ;mentése
        JMP FOLYT1 ;összehasonlítás folyt. köv.
HIBAVAN: POP D
        MOV C,E
        MOV B,D
        POP D
        MOV E,C ;azért kellett ez mert 2 DE van
        MOV D,B ;így megmaradt az adat/első rossz cime/
        POP B ;hossz vissza
        ORI 0 ;akkor orolsa(valami van benne), tehát Z==0 lesz
        RET
ELSO:   PUSH D ;első rossz címének mentése
        JMP FOLYT2
```