

| | | | |
|--|---------------------------|--|-------------|
| Név: Szabó Norbert | Gyak: Horváth Tamás (K2.) | 236 | Kód: AJD5YL |
| Feladat beadás: 2013.04.30 a gyakorlaton | | Pótbeadás: 2013.05.22. 9:30 - 10:30 - IB.310 | |

Beadáskor ezt a feladatlapot a megoldáshoz csatolni kell. A feladatokat külön lapon, kézírással oldja meg. A beadandó anyaghoz útmutatót a tárgy honlapján (www.iit.bme.hu/digit2) talál! Hibás megoldás javítására a pótbeadás alkalmával van lehetőség.

1. Illesszen 8085-ös mikroprocesszor alapú sínre 2764 típusú EPROM és 5565 típusú RAM memóriákat úgy, hogy az alábbi címtartományokat fedjék le:

1. 0000h-1FFFh EPROM
2. 2000h-3FFFh EPROM
3. 0000h-1FFFh vagy 8000h-9FFFh* RAM

* A 80h I/O címre írt adattal lehessen állítani, hogy a 0000h címen EPROM vagy a RAM memória látszódjon. (Ha a kiírt érték 0 az EPROM, ha 1 a RAM látszik) Gondoskodjon róla, hogy RESET után mindig az EPROM látszódjon! Ha a RAM a 0-ás címtől látszik, legyen írásvédett. Tervezze meg a szükséges IO egységet.

A sín jelei:

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

- 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.

- a. Írjon **KITOLT** szubrutint, amely a HL regiszterpárban egy kezdőcímet, a DE regiszterpárban egy hossz értéket kap, az így meghatározott memóriablokkot kitölti úgy, hogy minden páratlan címen elhelyezkedő byte az 0AAh, páros címen elhelyezkedő byte az 55h értéket tartalmazza.
- b. Írjon **ELLENOR** szubrutint, amely a HL regiszterpárban egy kezdőcímet, a DE regiszterpárban egy hossz értéket kap és ellenőrzi, hogy a memóriablokk rekeszei a **KITOLT** szubrutin által beírt értékeket tartalmazzák-e? A szubrutin **CY=0-val** jelezze, ha hibát talált. Ilyenkor a HL regiszterpár az utolsó (legmagasabb memóriacímű) **megtalált hiba címét**, a BC regiszterpár pedig a hibásnak talált **byte-ok darabszámát** tartalmazza. Ha nincs hiba **CY=1, BC=0** és HL a memóriablokk első elemére mutat.
- c. Írjon programrészletet, amely a processzor SID bemenetén fellépő 0→1 átmenet hatására a **KITOLT** és **ELLENOR** szubrutinok segítségével ellenőrzi a RAM területet. A teszt indulását és befejeződését a SOD kimeneten egy-egy 3 ms idejű impulzussal jelezze. Az időzítés meghatározásánál vegye figyelembe, hogy a program a feladatban meghatározott EPROM memóriában fut! Figyeljen arra, hogy a tesztelés alatt a RAM terület ne legyen írásvédett.

A szubrutinokat úgy írja meg, hogy a működéshez előírt regisztereken kívül más regiszterek értékét ne rontsák el! A szubrutinokat lássa el megjegyzésekkel és készítsen fejléct is!

Nyilatkozat:

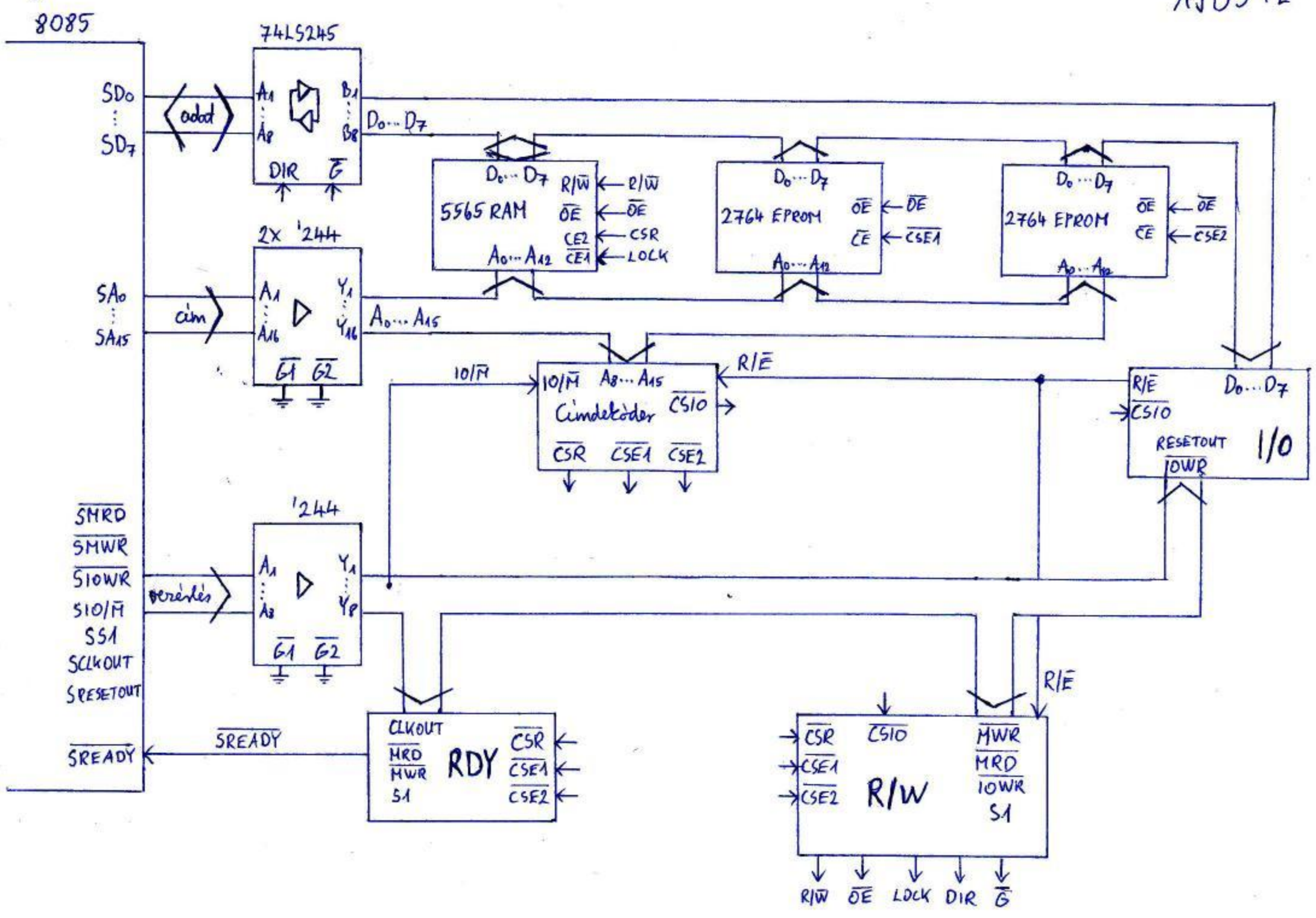
Alulírott Szabó Norbert nyilatkozom, hogy a házi feladatot meg nem engedett segédeszköz használata nélkül saját magam oldottam meg.

Dátum: 2013.

Aláírás:.....

1.9)

SZABÓ NORBERT
AJÓSYL

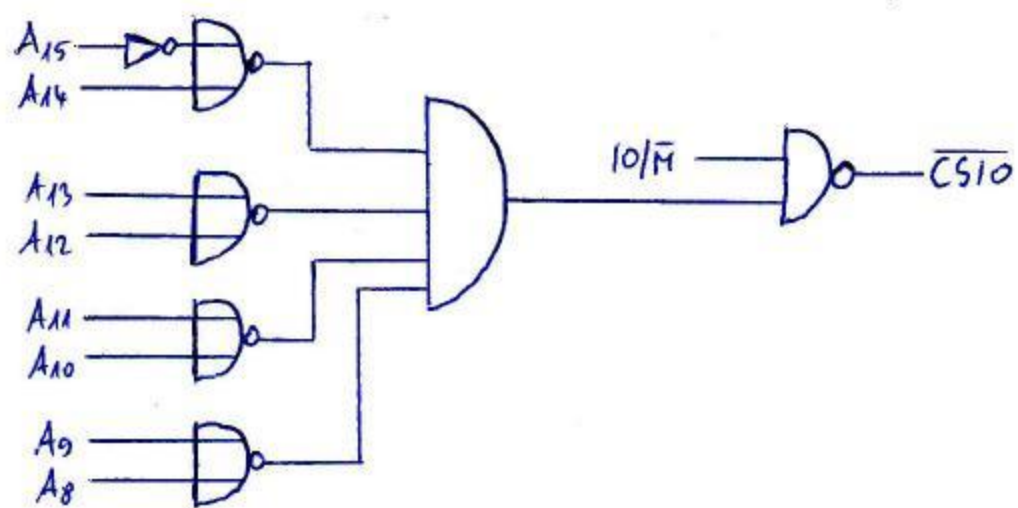
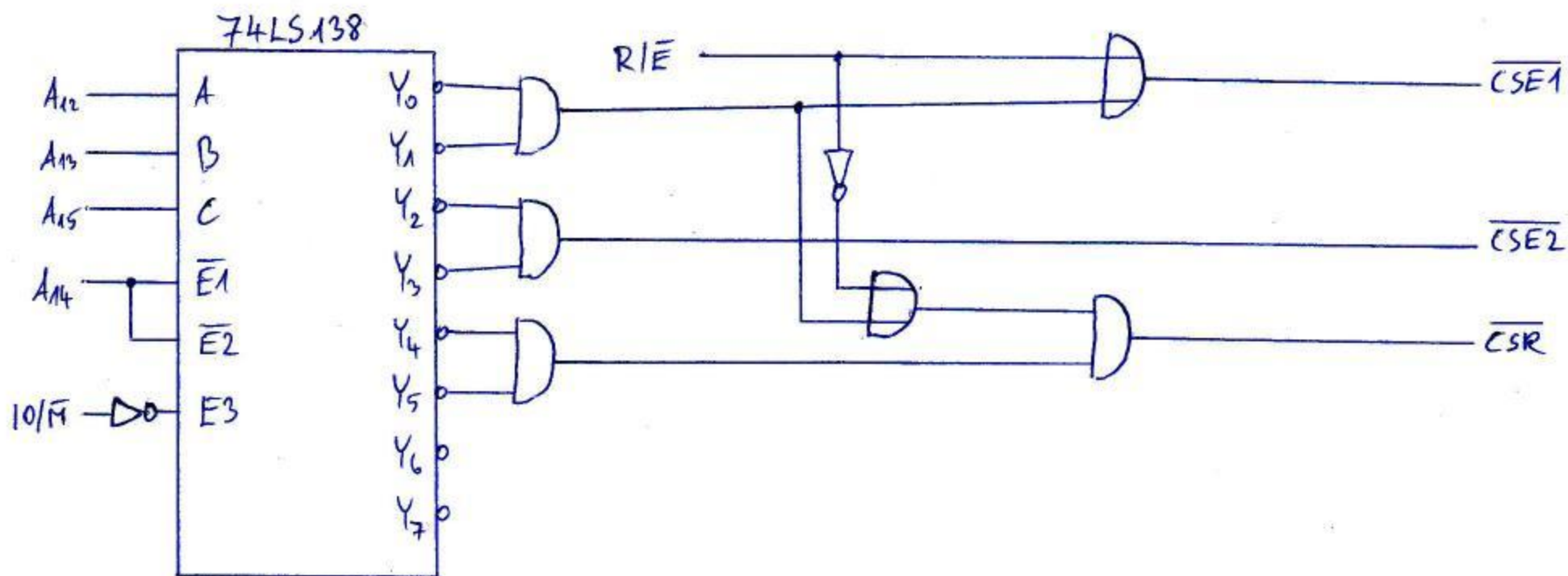


1. b)

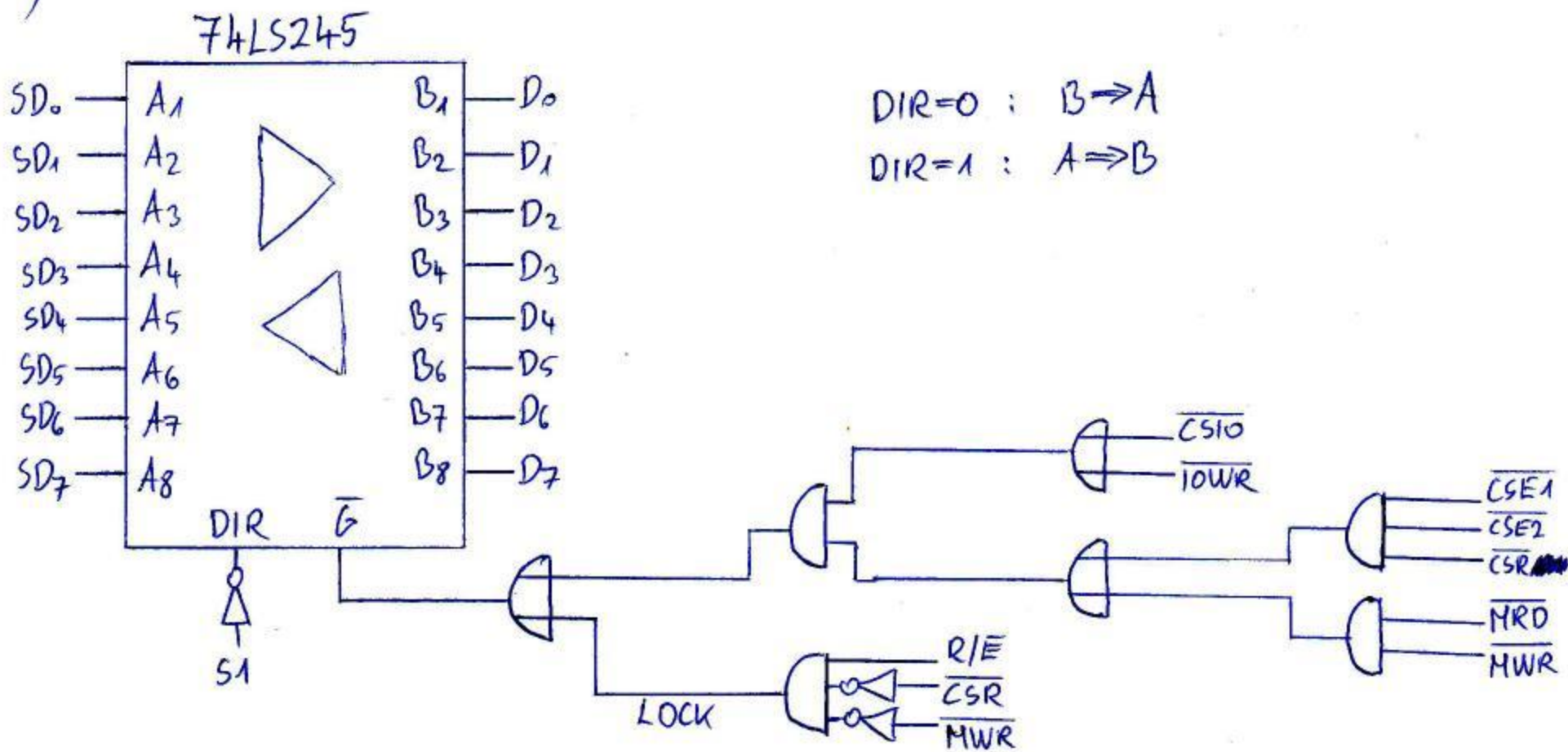
SZABÓ NORBERT
AJD5YL

| CPU | A ₁₅ | A ₁₄ | A ₁₃ | A ₁₂ | MEMÓRIA |
|----------------------------------|-----------------|-----------------|-----------------|-----------------|---|
| 0000H 0FFFH 1000H 1FFFH | 0 | 0 | 0 | 0 | } 8K RAM: 0000H 1FFFH vagy EPROM1: 0000H 1FFFH |
| 2000H 2FFFH 3000H 3FFFH | 0 | 0 | 1 | 0 | |
| | | | | | |
| 8000H 8FFFH 9000H 9FFFH | 1 | 0 | 0 | 0 | } 8K RAM: 0000H 1FFFH |
| | 1 | 0 | 0 | 1 | |

10 80H adat: RIE

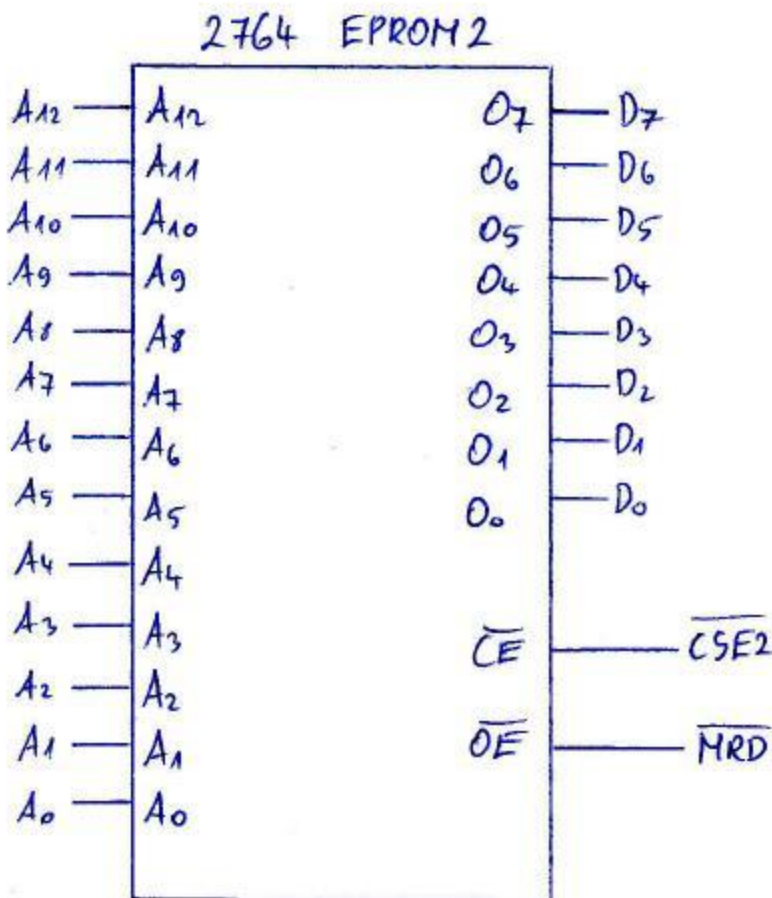
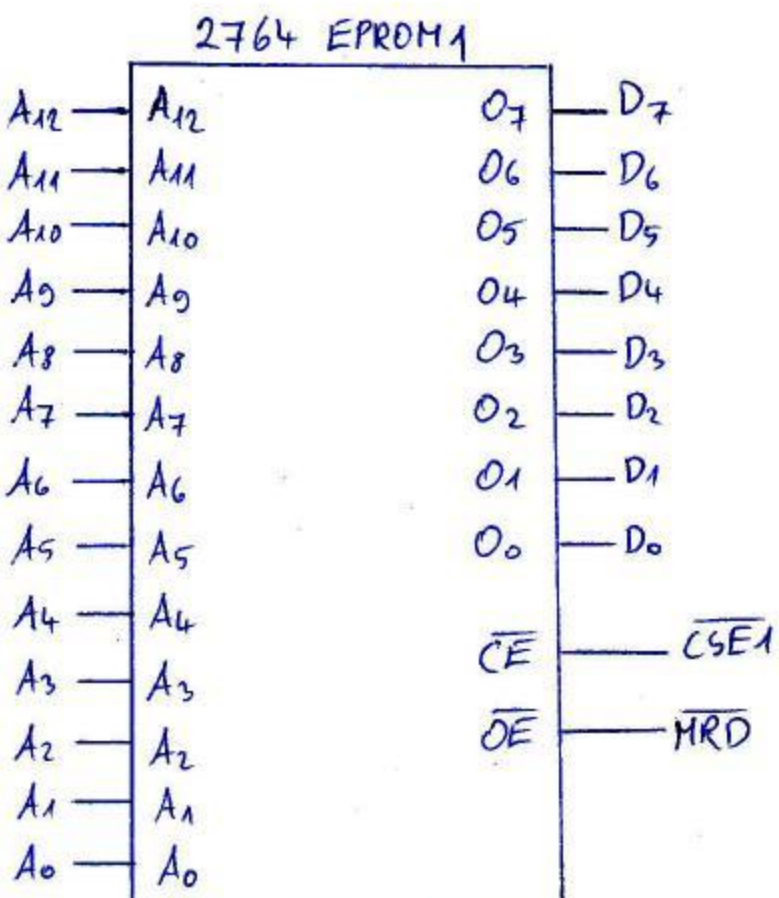
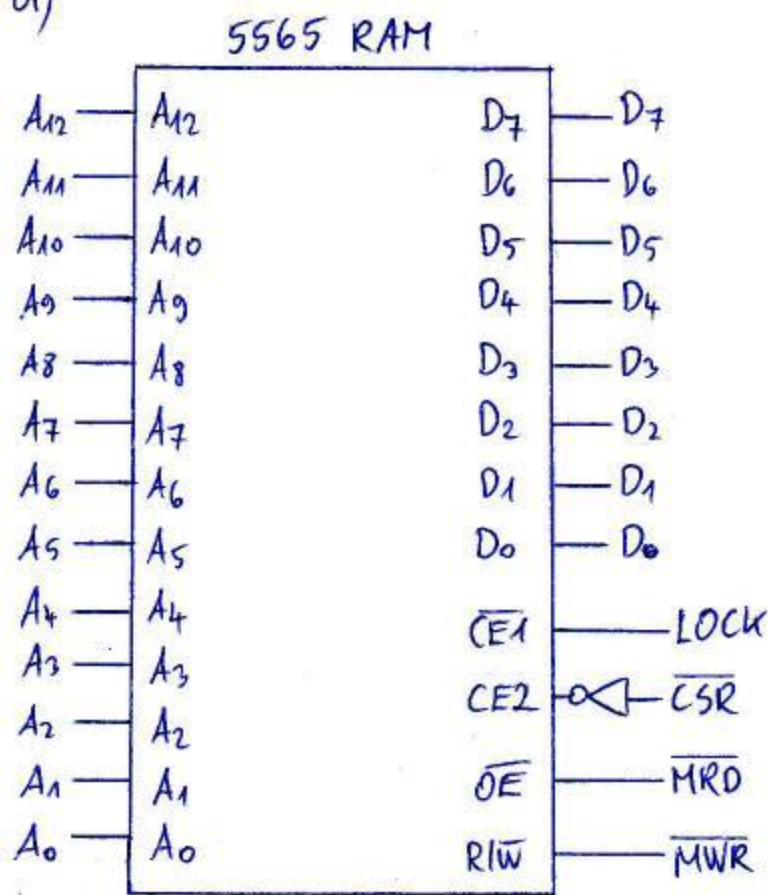


1.c)

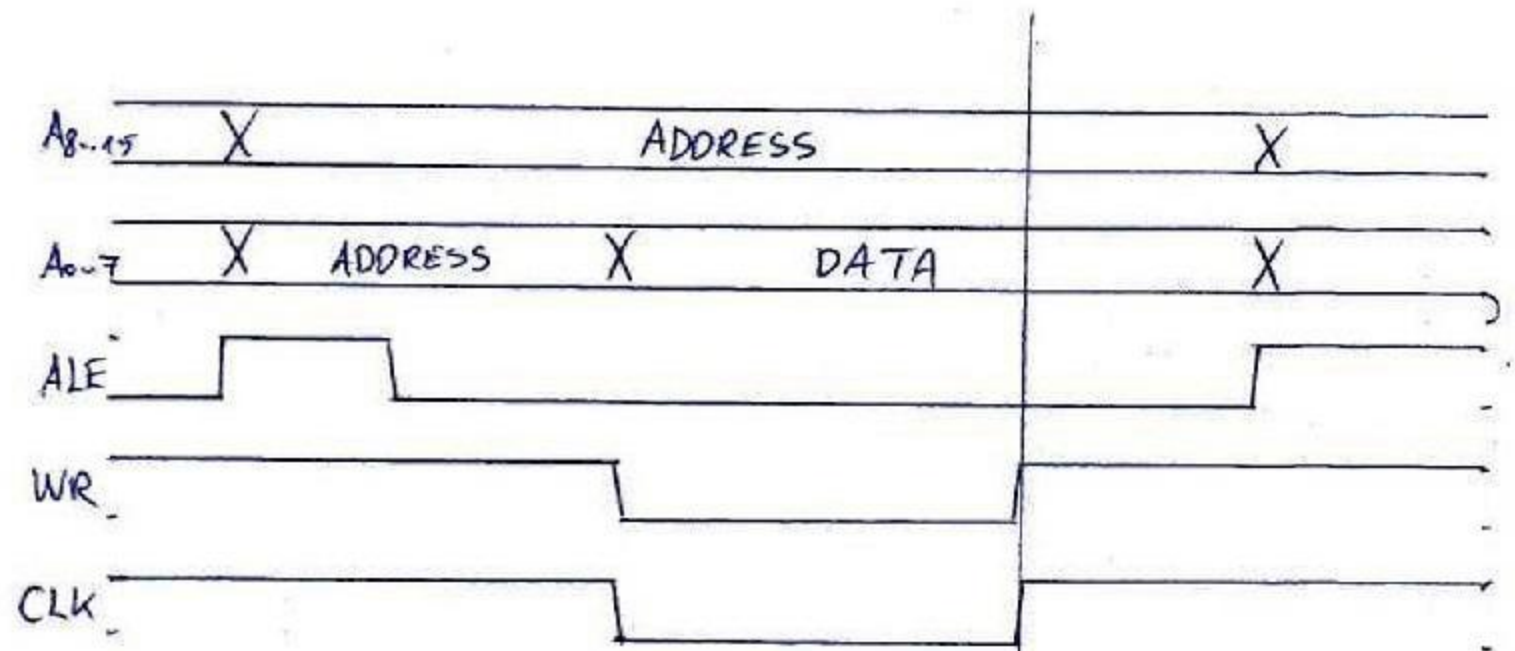
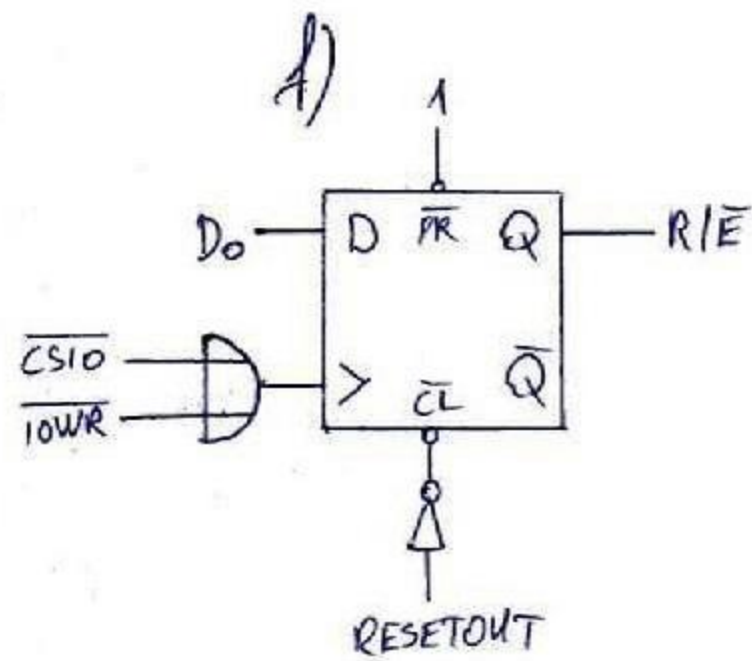
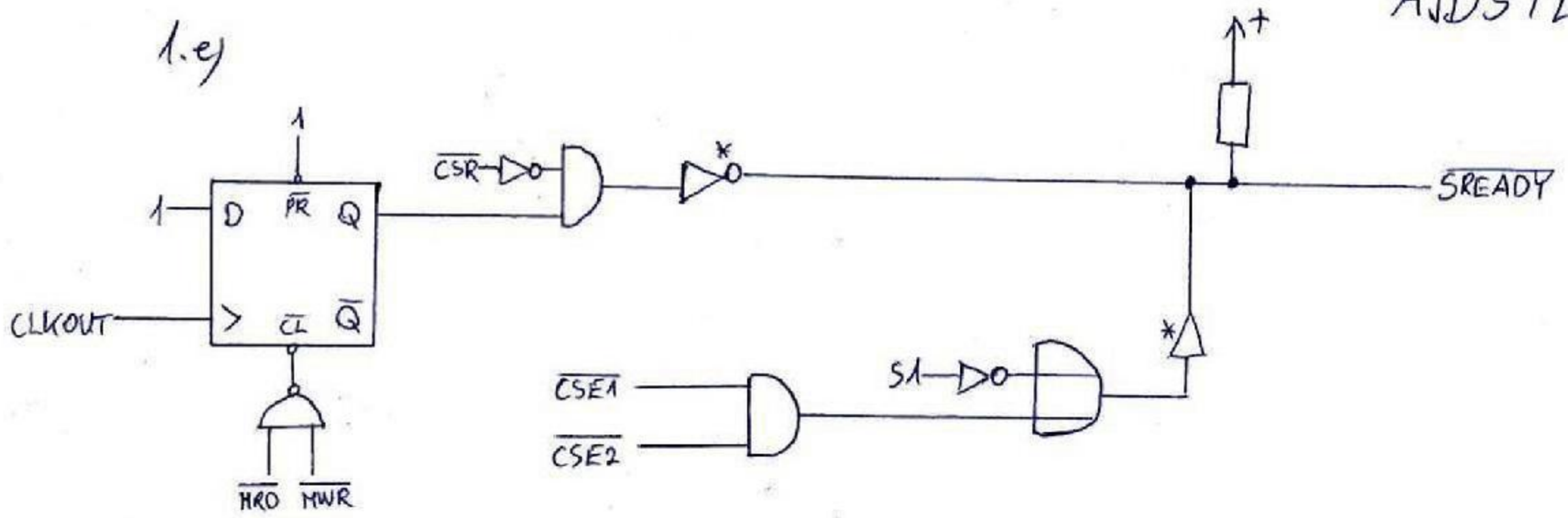


Ha $R/\overline{E} = 1$, akkor a 0000H címen a RAM látni, tehát inaktivált.

d)



1.ej



$$CLK = \overline{CS10} + \overline{10WR}$$

Ha nem a megadott cím van kiválasztva, $\overline{CS10} = 1$, $CLK = 1$ végig!
Ha a megadott cím van kiválasztva, $\overline{CS10} = 0$, $CLK = \overline{10WR}$

2.9)

SZABÓ NORBERT
AJDSYL

PAROSDATA EQU 55H
PARATLANDATA EQU 0AAH

KITOLT: PUSH PSW
PUSH H
PUSH D ; regiszterek mentése a stack-be
MOV A, E
ANA A ; zero flag állítás
JNZ MEHET ; vizsgáljuk, hogy a hosszúság nulla-e
MOV A, D
ANA A
JZ KVEGE ; ha D és E tartalma is nulla, vége

MEHET: MOV A, L
ANI 01H ; kezdőcím vizsgálata (páros/páratlan)
JNZ KPARATLAN

KPAROS: MVI A, PAROSDATA ; páros címre
XCHG ; DE és HL cseréje
STAX D ; DE által tárolt címre kiírjuk az akkumulátor tartalmát
XCHG
DCX D ; határoló méret csökken
INX H ; következő cím
MOV A, E
ANA A
JNZ KPARATLAN ; ha E nem nulla, megyünk tovább
MOV A, D
ANA A
JZ KVEGE ; ha DE tartalma nulla, vége

KPARATLAN: MVI A, PARATLANDATA ; páratlan címre
XCHG
STAX D
XCHG
DCX D
INX H
MOV A, E
ANA A
JNZ KPAROS ; ha E nem nulla, megyünk tovább
MOV A, D
ANA A
JZ KVEGE ; ha DE nulla, vége
JMP KPAROS ; ha D és E se nulla, folytatjuk a műveletet

KVEGE: POP D
POP H
POP PSW ; visszaállítás
RET

2. b)

; ELLENOR subrutin
 ; HL: kezdőcím
 ; DE: hossz

SZABÓ NORBERT
 AJD5YL

PAROSDATA EQU 55H

PARATLANDATA EQU 0AAH

ELLENOR: PUSH D

PUSH PSW

PUSH H ; mentés

LXI B, 0H ; kezdetben nulla hibra

MOV A, L

ANI 0AH ; kezdőcím páros/páratlan

JNZ EPARATLAN

EPAROS: MOV A, M

XRI PAROSDATA ; flag állítás

JZ OK1 ; ha megkelelő este volt, ugrunk

INX B ; egyelőre hibák nincsenek

INX SP

INX SP ; stack pointer visszaállítás

PUSH H ; felhívás a címet

OK1: DCX D ; határoló hossz csökken

INX H ; cím nö

MOV A, D

ORA E

JZ EVEGE ; ha E és D is 0, vége

EPARATLAN: MOV A, M ; ugyanaz páratlan címre

XRI PARATLANDATA

JZ OK2

INX B

INX SP

INX SP

PUSH H

OK2: DCX D

INX H

MOV A, D

ORA E

JZ EVEGE ; ha DE nem 0, folytatjuk

JMP EPAROS

EVEGE: MOV A, B

ORA C

JZ NINCSHIBA ; ha BC nullát tartalmaz, nincs hibra

VANHIBA: POP H ; utolsó hibra címre

POP PSW

STC

CMC ; CY=0

JMP FINISH

NINCSHIBA: POP H ; kezdőcím

POP PSW

STC ; CY=1

FINISH: POP D

RET

2.c) RAM EQU 8000H
SIZE EQU 2000H

PUSH PSW
PUSH B
PUSH D
PUSH H ; mentés

MVI A, 0H
OUT 80H ; RAM a 8000H címen látni, nem indokolt

WAIT: RIM
ANI 80H ; első bit vizsgálata
JNZ WAIT ; ha 1 a program indulásakor, akkor várunk

WAIT2: RIM
ANI 80H
JZ WAIT2 ; amíg 0, várunk, ha megjött a 0 → 1 átmenet, tovább

CALL IMPULZUS
LXI H, RAM ; kezdőcím
LXI D, SIZE ; méret
CALL KITOLT
CALL ELLENOR
CALL IMPULZUS

POP H
POP D
POP B
POP PSW
RET

IMPULZUS: PUSH PSW
PUSH H
MVI A, 0COH ; SOD=1 és engedélyezés
SIM ; SOD 1-be áll
LXI H, 459 ; alul vizsgálolva
MVI A, 0H
VAR: DCX H ; visszazárol
CML ; akkor isL összehasonlításra
JNZ VAR ; ha L nem nulla, visszazárol
CMP H
JNZ VAR ; ha H nem nulla, vissza
MVI A, 40H ; SOD=0 és engedélyezés
SIM ; SOD 0-ba áll
POP H
POP PSW
RET

Az EPROM 0 wait-tel működik!

CPU órajel 3,072 MHz
↓
1 fázis: $\frac{1}{3,072 \text{ MHz}} = 325,5 \text{ ns}$

Két SIM parancs között 3ms.

2 db MVI : 2 · 7 = 14 fázis
1 db LXI : 10 fázis
459 db DCX : 459 · 6 fázis
461 db CMP : 461 · 4 fázis 459 = 1 CBH
- 459 db CML
- 2 db CMP H 2 db CMP H
458 db igaz JNZ : 458 · 10 fázis
3 db hamis JNZ : 3 · 7 fázis
Összesen: 9223 fázis

$325,5 \text{ ns} \cdot 9223 = 3002087 \text{ ns} \approx \underline{\underline{3 \text{ ms}}}$