

Név: Szvoboda Márk	Gyak: <i>Pilászy György (SZ.)</i>	240	Kód: BFAWX5
Feladat beadás: 2013.05.08 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 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:

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

- 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 DE regiszterpárban egy kezdőcímet, a BC regiszterpárban egy hossz értéket kap, az így meghatározott memóriablokkot kitölti úgy, hogy minden byte a saját címének alsó 8 bitjét egy bittel jobbra forgatva tartalmazza.
 - b. Írjon **ELLENOR** szubrutint, amely a DE regiszterpárban egy kezdőcímet, a BC 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 DE regiszterpár az utolsó (legmagasabb memóriacímű) megtalált hiba címét, a HL regiszterpár pedig a hibásnak talált byte-ok darabszámát tartalmazza. Ha nincs hiba **CY=1, HL=0** és DE a memóriablokk első elemére mutat.
 - c. Írjon programrészletet, amely a processzor SID bemenetén fellépő 1→0 átmenet hatására a **KITOLT** és **ELLENOR** szubrutinok segítségével ellenőrzi a RAM területet. A teszt indulását és befejeződését a SOD kimeneten egy-egy 2 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.

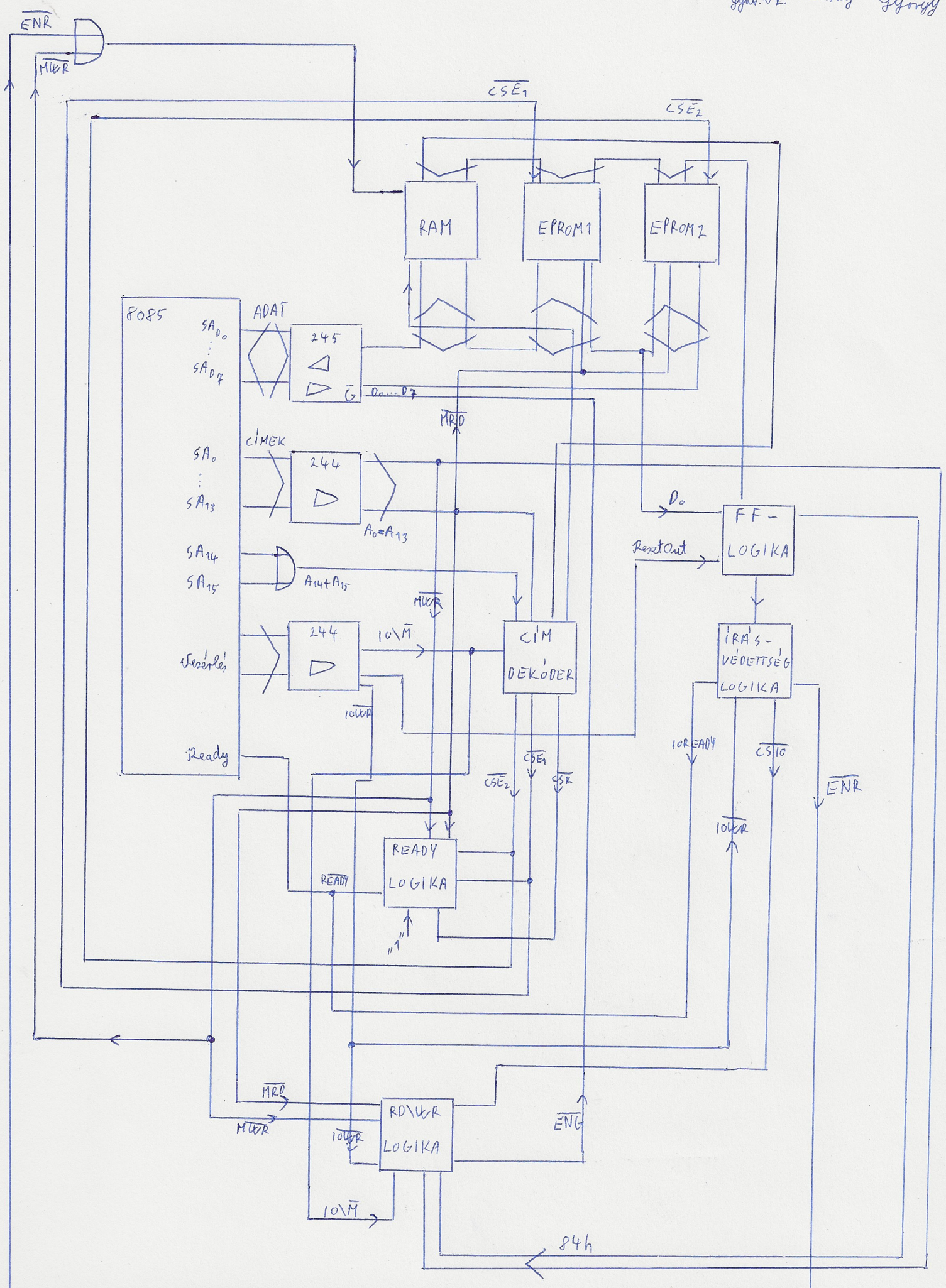
A szubrutinokat úgy írja meg, hogy a működéshez előírt regisztereken kívül más regiszterek értékét ne rontsák el! A szubrutinokat lássa el megjegyzésekkel és készítsen fejlődést is!

Nyilatkozat:

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

Dátum: 2013.

Aláírás:.....



CPU	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	MEM.
3FFF 3800	0	0	1	1	\bar{A}_{11} 1	RAM 17FF 1000
37FF 3000	0	0	1	\bar{A}_{12} 1	\bar{A}_{11} 0	RAM 0FFF 0800
2FFF 2800	0	0	1	0	\bar{A}_{11} 1	RAM 07FF 0000
27FF 2000	0	0	1	0	0	EPROM2 07FF 0000
1FFF 1800	0	0	0	1	1	φ 1FFF 1800
17FF 1000	0	0	0	1	0	φ 17FF 1000
0FFF 0800	0	0	0	0	1	φ 0FFF 0800
07FF 0000	0	0	0	0	0	EPROM1 07FF 0000

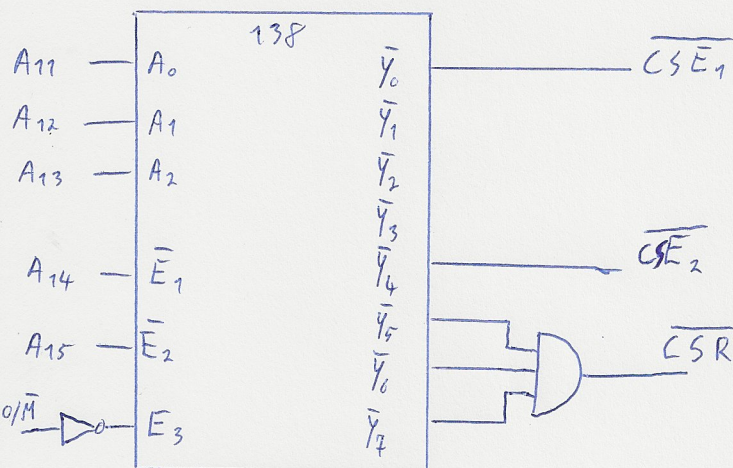
RAM:
 A₀...A₁₂

0 000 - 07FF: \bar{A}_{11}
 0800 - 0FFF: $\bar{A}_{11} \cdot \bar{A}_{12}$
 1000 - 17FF: \bar{A}_{11}

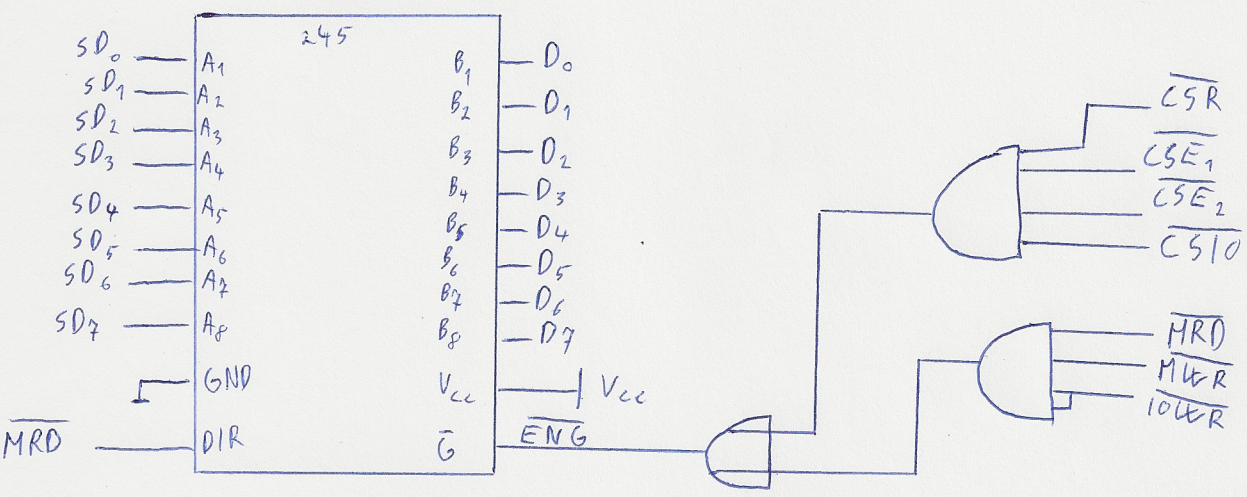
EPROM 2
 SA₀...SA₁₀ → A₀...A₁₀
 S_{IN}

Kihasználatlan

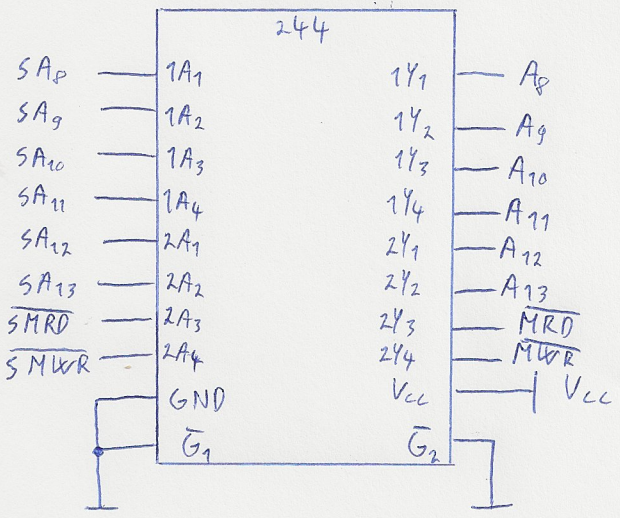
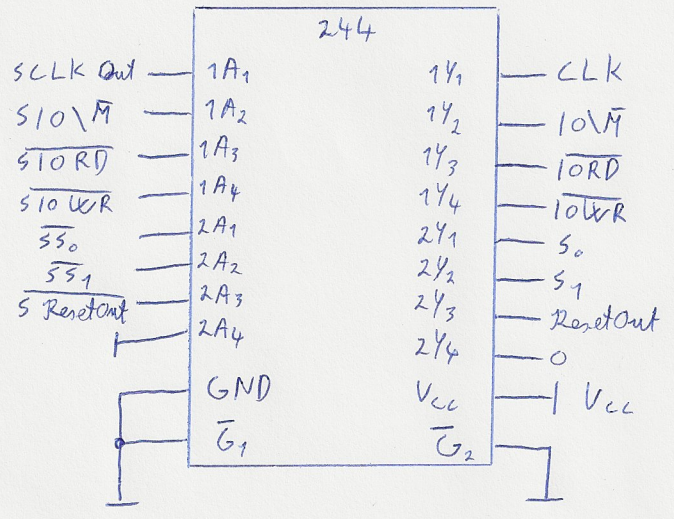
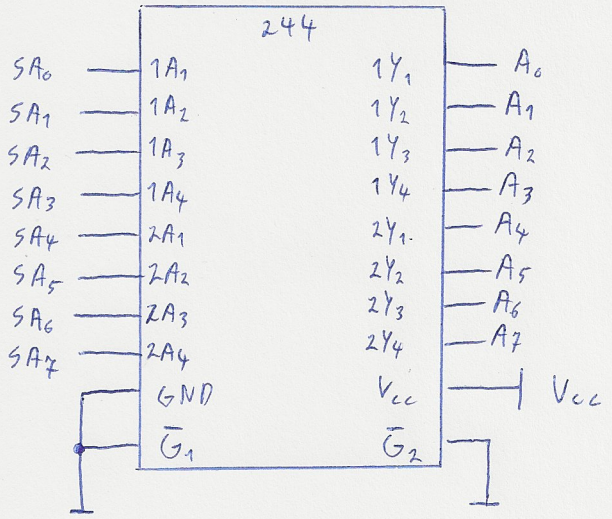
EPROM 1
 SA₀...SA₁₀ → A₀...A₁₀
 S_{IN}

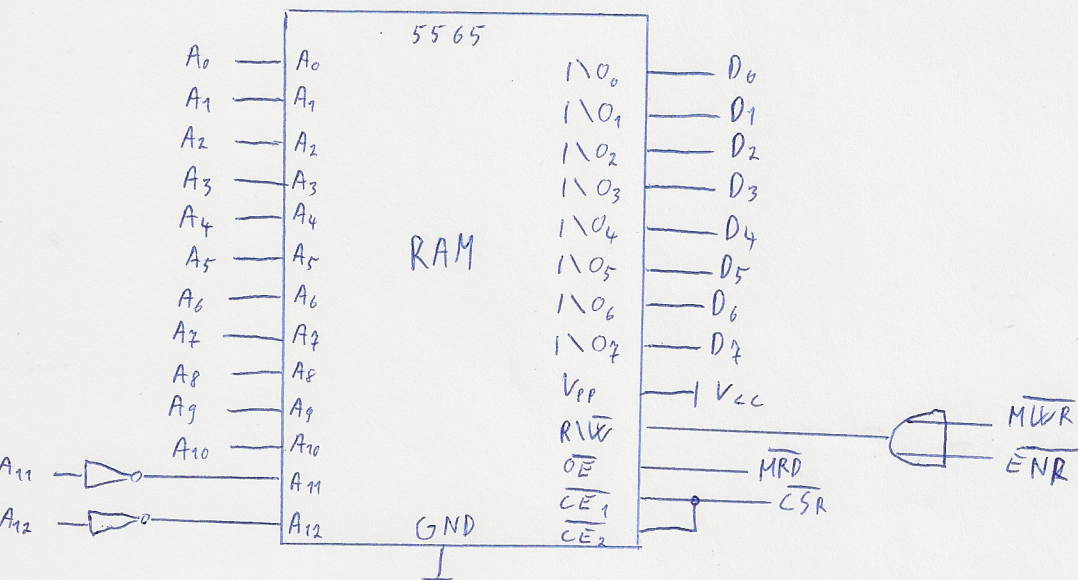
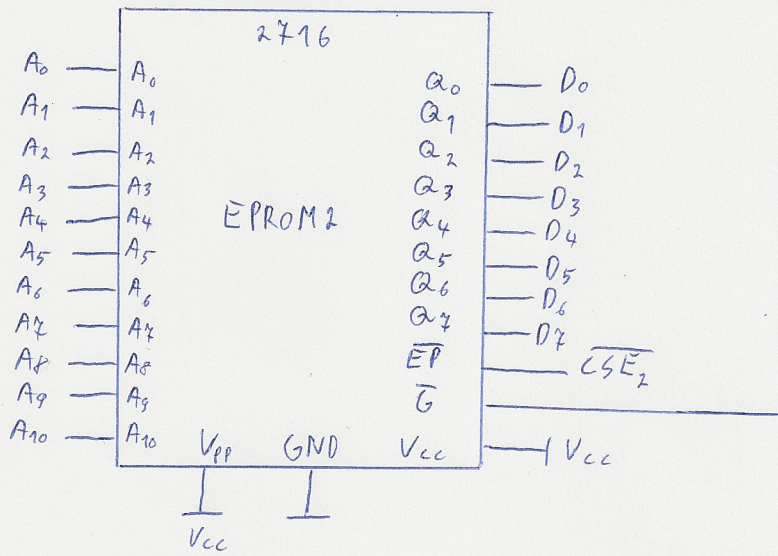
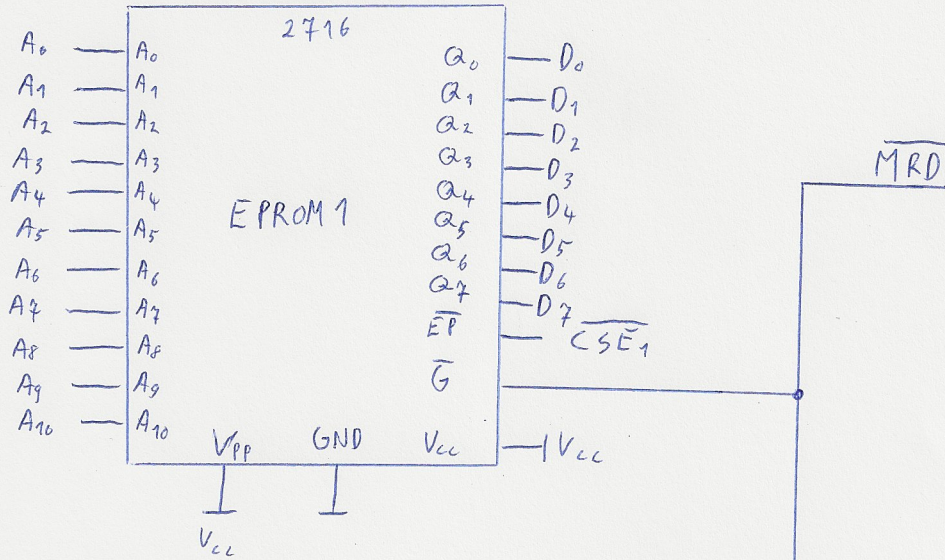


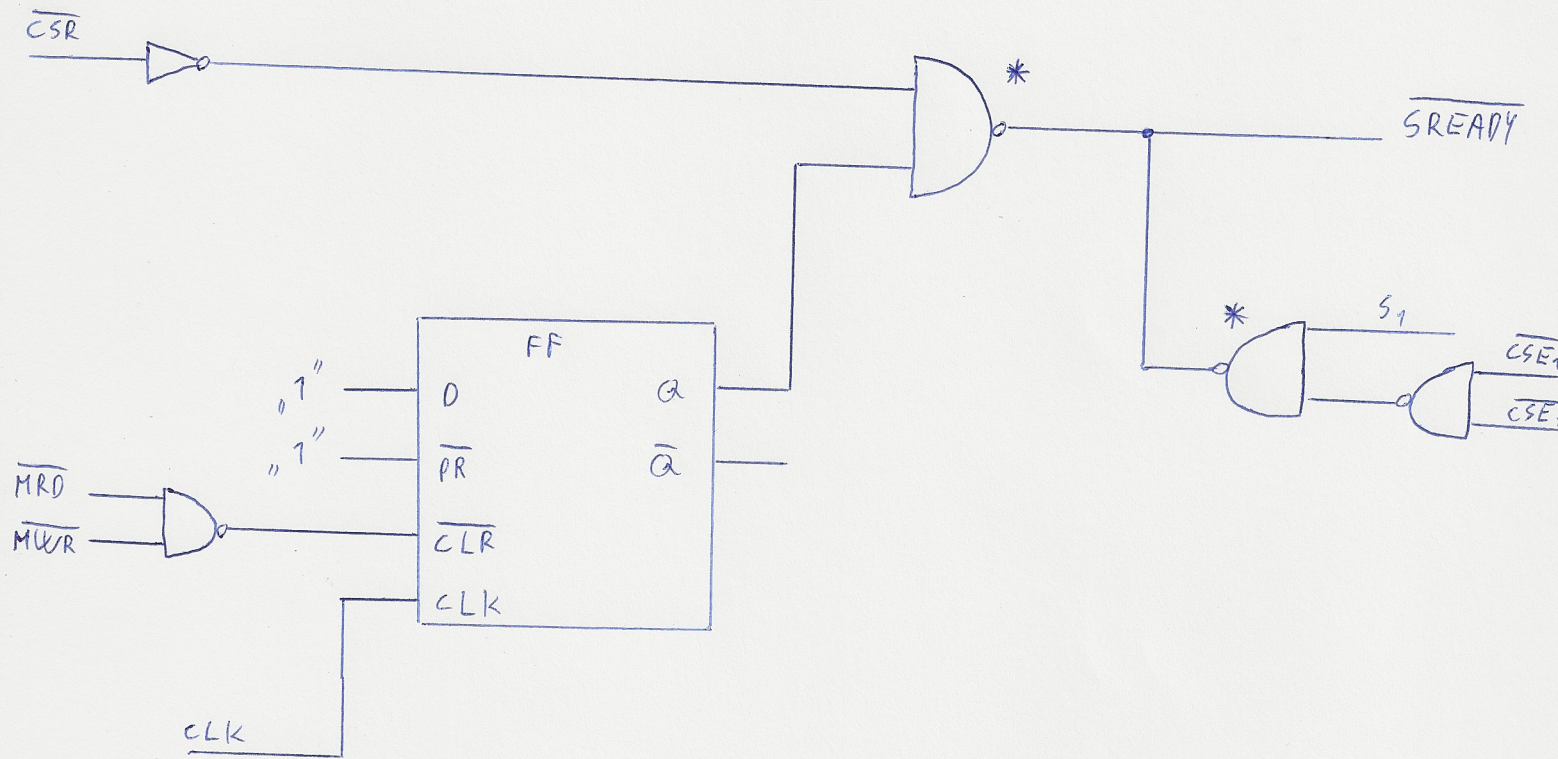
Adatleválasztás és adatirányítás:

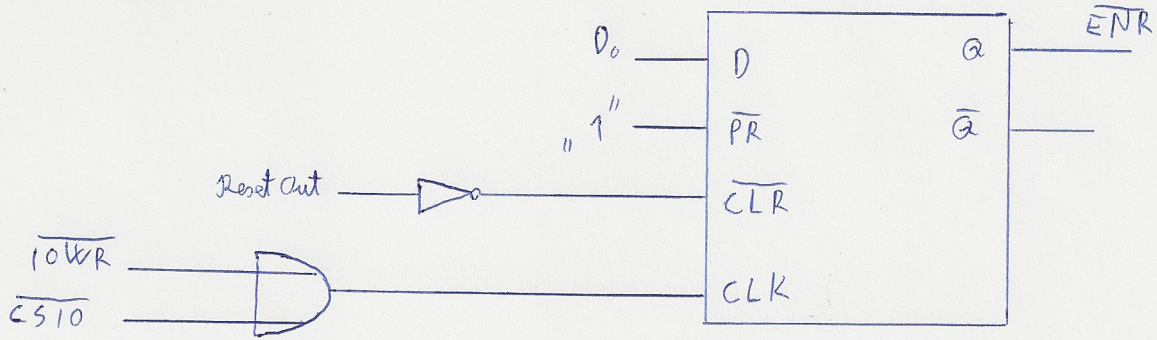


Címek és egyéb jelzések leválasztása:

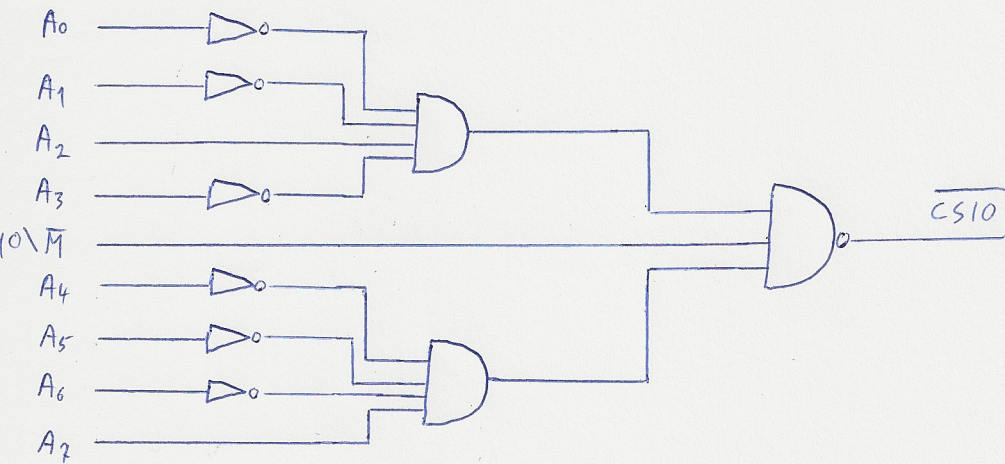








$\overline{CS10}$:



2/a

név: Szabolcs Márk
neptén: BFAXW5
gyul. nr.: Pálffy György

KITÖLT szubrutin

ront: -

kezdő cím: DE

hossz: BC

kezdőcímtől adott hosszal kitölti a memóriát az adott cím alsó bytejával
eggyel jobbrafordítottjával

KITÖLT: PUSH D

PUSH B

PUSH PSW ; PSW lementése

PUSH H ; HL regiszterpár mentése

MOV A, B

ORA C

SZ HO

CIKLUS1: MOV A, E ; alsó byte akkumulátorba

RRC ; forgatás jobbra

STAX D

INX D

DCX B

MOV A, B

ORA C

JNZ CIKLUS1

HO: POP H

POP PSW ; regiszterek visszaállítása

POP B

POP D

RET ; visszatér

2/20

ELLENOR subrutin

; kezdő cím: DE

; korr: BC

; Ellenőrzés DE című BC korrban a memória tartalmát

; Hiba esetén: CY=0 DE: utolsó hiba címe HL: hibák száma

név: Szabolcs Márk
név: BFAXW5
gyak. nr.: Pálffy György

ELLENOR: MOV A,B

ORA C

JZ L1

PUSH PSW

PUSH B

LXI H,0 ; HL nullázás

PUSH D ; kezdő címet mentjük

CIKLUSZ: LDAX D ; DE című adatot felhozzuk

RLC ; visszaforgatjuk

XRA E ; összehasonlítjuk a cím alsó byte-jával

JZ OK

POP PSW

PUSH D ; legfrissebb hiba címet tároljuk

INX H ; hiba számát növeljük

OK: INX D ; címet léptetjük

DCX B ; korrát csökkentjük

MOV A,B

ORA C

JNZ CIKLUSZ ; visszagyörgy a ciklusz-re ha kell

POP D ; utolsó hiba címet betöltjük DE-be

MOV A,H

ORA L

JZ L2

JNZ L3

L1: RET

folytatás a lap hátoldalán!!!

L2: POP B
 POP PSW
 STC
 JMP L1

L3: POP B
 POP PSW
 STC
 CMC
 JMP L1

ELLENOR
 MOV A, B
 OR A, C
 L1
 PUSH PSW
 B
 H, D
 D
 D
 RLC
 ARA E
 LE 0X
 POP PSW
 D
 H
 D
 B
 MOV A, B
 OR A, C
 JNF CIRCLES
 D
 MOV A, H
 OR A, L
 JZ L1
 JNS L2
 RET

ELLENOR
 MOV A, B
 OR A, C
 L1
 PUSH PSW
 B
 H, D
 D
 D
 RLC
 ARA E
 LE 0X
 POP PSW
 D
 H
 D
 B
 MOV A, B
 OR A, C
 JNF CIRCLES
 D
 MOV A, H
 OR A, L
 JZ L1
 JNS L2
 RET

; RAM terület tesztelése: 2800h-3FFFh
; indítás: SID bemeneten 1→0 átmenet
; SOD: 2ms impulzus

LXI D, 2800h

; ram kezdete

LXI B, 1FFFh

; ellenőrzés korra

MVI A, 0

OUT 84h

; iránvédettség megosztatása

WAIT 1: RIM

; beolvassuk az IT márkot

ANI 1000 0000B

JZ WAIT1

; márkolással megesszük az értéket, megvárjuk míg 1 lesz

WAIT 0: RIM

ANI 1000 0000B

JNZ WAIT0

; az előbbihez hasonlóan várunk, csak most nullára, itt lesz az 1→0 átmenet

CALL SOD

; impulzus kiadása

CALL KITOLT

; bitöltő meghívása

CALL ELLENOR

; ellenőrző meghívása

CALL SOD

; másik impulzus kiadása

HLT

; program vége

SOD: MVI A, 11000 000B

SIM

; SOD=1

MVI C, 236

VAR: DCR C

NOP

NOP

NOP

JNZ VAR

; 2ms várakozás

MVI A, 01000 000B

SIM

; SOD=0

RET

CPU órajele : 3,072 MHz

DCR 4 fázis
NOP 4 fázis
JNZ 10 fázis

} $4 + 3 \cdot 4 + 10 = 26$ fázis

$$x = \frac{2 \text{ ms}}{\frac{1}{3,072 \text{ MHz}} \cdot 26} \Rightarrow \underline{236} \text{ az időzítéshez való konstans}$$

Er a kód ezen sorában szerepel: MVI C, 236