Beugró kérdések:

1. USART jelalak (TdX) felrajzolása adott paritás és adott számú STOP bit mellett egy kétjegyű hexa szám átvitelére.

2. RST7.5, TRAP és INT megszakítási bemenetek összehasonlítása tilthatóság (DI), maszkolhatóság (SIM), illetve szubrutin kezdőcím alapján - 4 állítás és ahhoz kellett X-elni, amire igaz. (Állítások: tiltható DI-vel; maszkolható SIM-el; nem maszkolható; nem fix a megszakítási szubrutin kezdőcíme)

3. INTE FF-ot hogyan lehet tiltani/engedélyezni?

4. 74138-asba a /SRD /SWR IO/-M negáltak voltak bekötve, a szeparált /IORD /IOWR /MRD /MWR előállításához. A hiányzókat (/IORD és /IOWR) kellett bejelölni, hogy melyik kimeneten jelennek meg, illetve kérdés: hol kell ezt az áramkört használni és miért?

5. Rövid kód:

```
        ORG 100h
        DB .. (itt 2 byte volt, egy hexa és egy binárisan megadott)
        DS 2
        DW 7412h (írtam valamit, itt egy szó volt)
```

Táblázatba be kellett írni, hogy melyik memóriacímen milyen adat helyezkedik el? (Mindkettőt hexában kellett megadni.)

6. Rövid kód:

```
        SP EQU ...
        ORG ...
        JMP VALAMI
        ...
VALAMI: MVI B, 10
            MVI A, ....b (bináris szám, 8bit)
            OUT 70h (írtam valamit, egy perifériacím volt)
```

Kérdés: az OUT utasítás végrehajtásakor, a 3. gépi ciklusban mi van az adat és mi a címsíneken? (hexában)

7. 74138-ba volt bekötve az A15 A13 és A12 (CBA sorrend, $A=2^0$), a negált kimenetek közül az egyikre egy /CE1 volt írva, másik kettő pedig egy ÉS kapuval összekapuzva, az volt a /CE2. Kérdés: melyik hexadecimális címeket fedi le a /CE1 és a /CE2?

8. 8255-ös PPIO áramkör A portja 1-es üzemmódban működik. Egy bizonyos kimeneten lehet állítani az A porthoz tartozó INTE FF-ot. Melyik porton van ez a bit és mi a művelet, amivel lehet állítani?

9. A gépi ciklus mely fázisaiban, az órajel melyik élére mintavételezi a READY-vonalat a 8085-ös?

10. Volt három modul rákötve a D0 vezetékre. Mindegyik szeretne jelezni a D0-n, de nem egyszerre. Milyen kimenet kell? Miért?

Megoldások

1. USART TdX jelalak: START|8adatbit|paritás|STOP.
START=0, STOP=1. (Lehet 1-1.5-2 STOP bit)
A paritás bit a megadott paritás szerint pótolja ki az adatbiteket.
Páros paritás - páros számú 1-es legyen a 8adatbit+paritás bit (össz. 9bit) -ben.
Azt kell még tudni, hogy először a D0-t viszi át (legkisebb bit), majd a D7-et (legmagasabb bit).

2. RST 7.5: tiltható/engedélyezhető DI/EI-vel, SIM-el maszkolható, fix cím.
INT: tiltható/engedélyezhető DI/EI-vel, nem maszkolható, nem fix cím.
TRAP: nem tilható/engedélyezhető, nem maszkolható, fix cím (RST 4.5 címe)

3. Engedélyezés: EI, tiltás: RESET után, DI, megszakítás érvényre jut.

4. Ki kellett okoskodni...A DMA-vezérlő esetén kell, mert egy DMA-ciklusban egyszerre kell memóriát írni és perifériát olvasni, vagy memóriát olvasni és perifériát írni.

5. ORG - innen kezdi a foglalást
DB - inicializált bytefoglalás, a bináris számot át kellett írni hexába.
DW - egy szó (két byte) foglalása. Előbb az alsó, majd a felső byte.
DS - inicializálatlan helyfoglalás, DS 2 ⇨ 2 byte helyet foglal, a byte-ok értéke ismeretlen.

6. OUT utasítás
1. gépi ciklus: opkód fetch
2. gépi ciklus: port címének felhozása
3. gépi ciklus: az ACC-ben lévő adat kiírása a megadott portra

Tehát a kiíráskor az adatsín az előzőleg az ACC-be elhelyezett adat volt (binárisan volt megadva, át kellett írni hexába), a címsínen pedig a port címe.

7. Itt A15 A13 A12 értéke volt adott, a címek így néztek ki: A15|X|A13|A12. Meg kellett nézni, hogy a megadott kimenetek A15 A13 A12 milyen kombinációjához tartoznak. A címtartomány eleje X=0 helyettesítéssel volt megkapható: A15|0|A13|A12, a teteje pedig X=1 helyettesítéssel. Ahol kettő volt összekapuzva, ott a címtartomány alja az alacsonyabb sorszámú kimenet X=0 esetén, a teteje pedig a magasabb sorszámú X=1 esetén. Kerek ezres címek voltak, csak az első hexa értéket (4 bit) kellett kiszámolni.

8. A C porton lehet engedélyezni/letiltani (PC 4-es bit), a bit set/reset művelettel.

9. A T2 fázisban, illetve (ha van) a WAIT fázisban, az órajel felfutó élénél.

10. Nem volt külső felhúzó ellenállás, de kell a 3. állapot (nem kapcsolódik a vezetékhez), tehát 3state kimenet szükséges.

Nagypéldák

1. Egy 8k-s ROM és egy 2k-s RAM illesztése 8085-ös sínre. A címek:
ROM: 0000-0FFFh (4k), illetve 2000-2FFFh (4k) RAM: 1800-1FFFh (2k)

1.1 Címtérkép 2k-ként: A15...A11 | A10-A0 címvezetékek értéke, milyen memória van ott?
1.2 Hogy helyezkedik el az EPROM-ba írt adat a memóriacímekben, illetve az EPROM-ban? 2k-ként kellett megadni a címtartományokat.
1.3 Teljes címdekódoló hálózat készítése 74138-al és minimális kiegészítő áramkörrel.
1.4 Memóriaáramkörök bekötése, az EPROM-nál figyeljünk a az 1.2-ben felírtakra.
1.5 Adatbuszmeghajtó (74245-ös) DIR és /G jeleinek előállítása, illetve az A és a B oldal bejelölése (jobb oldalt volt a modul, bal oldalt az adatbusz).
1.6 READY-logika. RAM 0 Wait, az EPROM olvasásra 1 WAIT, írásra nem ad READY-t. (Egy D flip-flop volt lerajzolva, mellette egy kis hely a RAM ready-jének)

2. 8 bemenetű VAGY kapu áramkör tesztelő. Egy 8 bites regiszter volt kötve a D7..D0-ra, ennek a kimeneteire egy nagy VAGY kapu, majd ezután egy 74244-en (2*4 bit) egyirányú buszmeghajtó. A VAGY kapu kimenete a 74244-es egy bemenetére csatlakozott, a többi bemenet 0-ra volt kötve.
A tesztelés menete: egy pergésmentesített kapcsoló ad a SID-re egy jelet, ez indítja a tesztelést. A regiszterbe a 00h címen lehet írni, a kimeneti buszmeghajtót (74244) a 00h címen olvasni. Az így beolvasott adat legalacsonyabb bitje a VAGY kapu kimenete.

2.1 Be kellett kötni a D0..D7-et a regiszter bemenetére, illetve a regiszter /CK↑ jelét előállítani. (A regiszter felfutó élre ír be.) A buszmeghajtó áramkör kimenetére be kellett kötni a D7...D0-t a megfelelő sorrendben, illetve a /G1 és a /G2 jeleit bekötni.

2.2 Elő kellett állítani a modul címdekóderét, figyelve arra, hogy DMA-vezérlő is van a rendszerben és a /AEN=0 jel esetén nem a CPU, hanem a DMA vezérel, nincs /CS. Illetve elő kellett állítani a /READY-t. (Nincs WAIT)

2.3 Kapuáramkörökkel le kellett rajzolni egy kapcsoló pergésmentesítését.

3. Assembly-rutinok készítése a 2. feladat hardwere-éhez

3.1 **TEST**: az A regiszterben kapott értékkel végzi el a VAGY kapu tesztelését. Ha a VAGY kapu hibásan működik, akkor CY=1-el jelez. Regisztereket ne rontsunk, kommentezzünk.

3.2 **CHECK**: az összes lehetséges kombinációra elvégzi a tesztelést, az előzőleg megírt **TEST** felhasználásával. Az első hibánál megáll, és CY=1-el jelzi a hibát.

4. Kódanalízis. Volt egy hosszabb kód, feltételes ugrásokkal, egy RST 7 hívással (az RST 7-hez tartozó memóriacímre is volt elhelyezve kód!), a végén HLT-vel áll meg.

4.1 Melyik utasításkor mi van az adat/címsínen, RD/WR jelek értéke, illetve melyik regiszter és hogyan változott az adott utasításnál. (Fázisonként kellett)

4.2 A HLT utasítás kiadása előtt mi van az egyes regiszterekben? (Összes belső regisztert kérdezték, a feladat elején megadták, hogy mi a regiszterek kezdeti értéke.)

Megoldások

1.4 Az A12 címvezeték helyére a bekötésnél A13-at kell kötni, így a 2000-ből 1000 lesz (0010 ⇨ 0001), a 2FFF-ből pedig 1FFF, (a 0000-0FFF értéke így sem változik) így folytonosan helyezkedik el az EPROM-ban ez a tartomány.

1.5 Az adatbusz felőli oldal az A, a modul felőli oldal a B. Ekkor DIR=/MRD, G=(MRD+MWR)*(CSE+CSR). Ezt legkönnyebben három NAND kapuval lehet megoldani: az egyikbe mennek a /CS-k, a másikba a /MRD és a /MWR, majd ennek a kettőnek a kimenete megy a harmadik NAND kapuba, aminek a kimenete lesz a /G. (Vagy két AND és egy OR kapu, hasonló bekötésekkel, a negált logika miatt.)

1.6 Gyakanyagban volt sok READY: D bemenetre megy a /CSEPROM invertálva, /CL-re a /MRD invertálva, /Pr-re „1”-et kötünk, a CLK-ra pedig CLKOut-ot. Az invertált kimenetet (/Q) egy open collector (*) bufferen keresztül kötjük a /SREADY vezetékre. A RAM-nál a /CSRAM-ot egy open collector bufferen keresztül kötöm a /SREADY-re. (Vagy a ponált Q kimenetre egy o.c. invertert rakok, és úgy lesz /SREADY).

2.1 A regiszter bemeneteire simán rá lehet kötni az adatbusz SD7...SD0 vonalát. A regiszter CK↑ bemenetére a modul /CS-t, illetve a /IOWR-t kell kötni egy NOR-kapuba: a tesztelés elején a modul kiválasztódik, ekkor /CS-n egy 1⇨0 átmenet lesz. Ez bekövetkezik akkor is, ha olvasunk, azonos a beírási és a kiolvasási cím; nekünk csak írásra kell, ezért ÉS-elni kell a /IOWR-t (negált logikában VAGY kapu).
Ekkor, ha a modul címére írunk, akkor az OR kapu kimenetén egy 1⇨0 átmenet lesz. Nekünk egy 0⇨1 kell a beíráshoz, ezért kell az inverter ⇨ NOR-kapu.

A kimeneti meghajtó bekötésénél arra kellett figyelni, hogy a D0 (legalacsonyabb bit) címvezetékre kössük azt a vonalat, ahol megjelenik a VAGY kapu kimenete, a többi sorrendje tetszőleges, mivel mindegyik 0.

A kimeneti meghajtónál a /G1-re és /G2-re a /IORD és a /CS OR-kapuval képzett kombinációját kell kötni, így a kimeneti meghajtó csak olvasásnál hajtja meg a sínt, írásnál nagyimpedanciás állapotban vannak a kimenetek.

Ha van DMA, akkor a NOR, illetve OR kapuba még a /AEN-t is be kell kötni egy inverteren keresztül mind a kimenetnél, mind a bemenetnél, így biztosítva a modul leválasztását a sínről. (/AEN akkor 1, ha a CPU hajtja a sínt ⇨ //AEN akkor 0, ha a CPU hajtja a sínt)

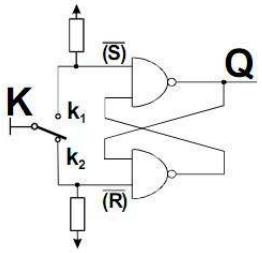
2.2 Biztos van szebb megoldás is, a favágó módszer: 00h úgy áll elő, ha minden címvezetéken (SA7...SA0) 0-n van. SA7-et kötöttem a /E1-re, /E2-re az IO/-M-et egy inverteren keresztül, míg az E3-ra a /AEN-t kötöttem: ha /AEN=1, akkor nem a DMA-hajt, ez nekünk jó.

A maradék (SA6...SA0) címvezetékeket tetszőleges sorrendben VAGY-oltam kapukkal, a lényeg, hogy mindhárom kiválasztó bemenetre (ABC) jusson legalább egy vezeték. Ekkor ha minden címvezetéken 0 van, akkor az Y0-s kimenet invertálásával kapjuk a CS-t a modulhoz.

Ha SA7=1 vagy IO/-M=0 (tehát nem perifériacím) vagy /AEN=0 (DMA hajt), akkor a címdekóder minden kimenete 1-es, nincs /CS. Ha SA6...SA0 közül bármelyik 1 lesz, akkor az ABC valamelyikén ez megjelenik (közvetve a VAGY kapuk által vagy közvetlenül, ha be van kötve), és nem az Y0 kimenet lesz kiválasztva.

2.3 (Digit1-es gyakanyagból)

Pergésmentesítés
(A kapcsoló pergése esetén a kimenet csak egyszer vált)



3.1 Az adatvezetékek VAGY kapcsolata miatt akkor és csak akkor lehet a visszaolvasott adat 00h, ha a kiírt adat is 00h. Ha a kiírt adat nem 00h, akkor a visszaolvasott adat legkisebb bitje (D0 vezeték) 1-es ⇨ 01h. Tehát az algoritmus:

1. Lementem a PSW-t (ez menti ACC-t is), illetve a regiszterpárokat.
2. OUT-al kiküldöm az ACC-ben kapott adatot a 00h-ra.
4. Megnézem, hogy ACC-ben 0 vagy nem 0 van (pl. ANA A, ORA A-vel), majd JZ vagy JNZ-val ugrok

JZ:
1. Közvetlen írok pl. a B-be 00h-t (ACC itt =00h, de tökmindegy, az IN úgyis felülírja)
2. Beolvasok (IN 00h)
3. CMP-vel összehasonlítom az A-t és a B-t. Ha a két érték egyezik, akkor JMP NINCSHIBA, egyébként JMP HIBA

JNZ:
1. Közvetlen írok pl. a B-be 01h-t (ACC itt !=00h, de tökmindegy, az IN úgyis felülírja)
2. Beolvasok (IN 00h)
3. CMP-vel összehasonlítom az ACC-t és a B-t.
4. Ha a két érték egyezik, akkor JMP NINCSHIBA, egyébként JMP HIBA

NINCSHIBA: fordított sorrendben POP-olok mindent, majd a POP PSW után STC és CMC paranccsal CY=0-t állítok (bár ez nem volt leírva, de szerintem ez is kell), végül RET

VANHIBA: fordított sorrendben POP-olok mindent, majd a POP PSW után STC paranccsal CY=1-t állítok, végül RET

3.2 Az összes teszteset száma 2^8=256, ez elfér egy regiszterben.

1. Lementem a regisztereket és PSW-t
2. Pl. B-be rakom a ciklusváltozót (B-be 256-ot töltök), ACC-ba pedig az első esetet: 00h
3. STC, majd CMC-vel CY=0-t állítok, hogy biztosan 0 legyen
CIKLUS:
3. CALL TEST (nem módosít semmilyen regisztert, csak a CY-flaget)
4. CY-t vizsgálok, ha CY=1, akkor hiba van (JC HIBA), egyébként pedig növelem ACC-t, csökkentem B-t.
5. B?=0
⇨ ha nem, akkor JNZ CIKLUS,
⇨ ha igen, akkor a teszt sikeresen lefutott, JZ NINCSHIBA

(Ez a „szemléletesebb” algoritmus. **De**: az ACC-ben ugyanazt számolom, mint a B-ben, csak fordítva, az ACC=FFh=256 érték növelése után ACC=00h=0 jön ⇨ az ACC mint a tesztadat és mint ciklusváltozó használható, a B regiszteres móka ekkor elmarad, az 5. lépésben elegendő az A?=0-t vizsgálni ⇨ ha 0, akkor átfordult a számláló, a teszt sikeresen végigment.)

HIBA: POP-olok mindent, majd RET előtt STC-vel CY=1-t állítok
NINCSHIBA: POP-olok, RET előtt STC és CMC-vel CY=0-t állítok

Ezek az általam leírt megoldások, vagy ahogy az IIT mondja: sajtóhibák előfordulhatnak.
A beugrók megoldását a segédlet alapján ellenőriztem; a 2.1, 2.2, 3.2 feladat helyes megoldását Ambrits D. közölte a tárgyi levlistán - az ő megoldásával egyetértek, ezért azt írtam le. Ha valaki hibát talál, kérem jelezze.

Dave (idavid KUKAC sch bme hu)