



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Commodore floppy meghajtó megvalósítása FPGA-val

MSC ÖNÁLLÓ LABORATÓRIUM 2.

Készítette
Weisz Pál

Konzulens
Horváth Kristóf

2022. december 15.

Feladatkiírás

A hallgató feladata, hogy egy Commodore számítógépekhez készült floppy meghajtó egységet valósítson meg FPGA segítségével. A munka magába foglalja az eredeti hardver megismerését és annak áttervezését úgy, hogy azt az interfészáramköröket leszámítva meg lehessen valósítani modern technológiák felhasználásával. Cél még, hogy kész rendszer képes legyen valódi (azaz nem virtuális) floppy lemezek kezelésére egy PC floppy meghajtó felhasználásával.

Tartalomjegyzék

1. Bevezető	3
1.1. A feladat összefoglalása	3
1.2. Rendszerterv	4
1.3. Protokollok	5
1.3.1. A Commodore IEEE-488 soros busz	5
1.3.2. Floppy meghajtó	6
2. A hardver	8
2.1. Bemutatom a fejlesztőkártyát	8
2.2. Az interfészpanel	8
3. Digitális tervezés	10
3.1. Modulok	10
3.1.1. CPU	11
3.1.2. FDC	11
3.1.3. CIA	11
3.2. Tesztelés	11
4. Tesztelés, hibajavítás	12
4.1. Címkék és hivatkozások	12
4.2. Ábrák és táblázatok	13
4.3. Irodalmi hivatkozások	13
4.4. A dolgozat szerkezete és a forrásfájlok	17
4.5. Alapadatok megadása	18
4.6. Új fejezet írása	18
Irodalomjegyzék	20
Függelék	21
F.1. Az eredeti 1581-es floppy meghajtó kapcsolási rajza	21
F.2. Az interfészkártya kapcsolási rajza	26
F.3. Az interfészkártya NYÁK-terve	30

1. fejezet

Bevezető

1.1. A feladat összefoglalása

Cél tehát, hogy egy szabványos IBM PC floppy meghajtót felhasználva FPGA segítségével egy olyan hajlékonylemez-es egységet valósítsak meg, mely Commodore-számítógépekkel kompatibilis és a hozzájuk való floppy-lemezeket írni-olvasni képes.

Az önálló laboratórium alatt végzett munkám során egy Commodore 1581 típusú, 3,5"-es lemezmeghajtó prototípusát készítettem el ily módon. A kitűzött cél megvalósításához felkutattam az eredeti meghajtó kapcsolási rajzát, szervizkönyvét, elolvastam a konstrukcióban alkalmazott alkatrészek adatlapjait. Megismerkedtem a Commodore számítógépes környezetben alkalmazott perifériák, interfészek szabványaival, működésével is. Ezek alapján, valamint az általam használt FPGA-fejlesztőpanel és PC-s meghajtó jelszintjeinek, tulajdonságainak figyelembe vételével megterveztem egy interfészpanelt, mely az alaplap vezérlő logikát implementáló FPGA-t illeszti a fizikai környezetéhez, vagyis a meghajtó hardveregységéhez és a számítógéppel való kapcsolatot biztosító soros porthoz.

Az eredeti 1581-es vezérlését egy komplett mikroprocesszoros rendszer látja el, melyet - funkcionalitását tekintve - egy az egyben az FPGA-n szintetizálva valósítottam meg. Az ehhez szükséges verilog-modulok egy részét korábbi projektekből emeltem át, a hiányzóakat pedig én magam implementáltam.

A rendszert először modulonként, majd különböző integráltsági szinteken is teszteltem, először a Xilinx ISE szimulációs környezetében, majd pedig logikai analízátorral a kész hardveren is. Az FPGA-n megvalósított rendszer alkalmas az eredeti 1581-es firmware-ének futtatására, mely bináris állomány formájában rendelkezésemre állt. A kezdeti sikeres integrációs teszteket követően processzor belső állapotát nagyon körülményessé vált vizsgálni, és mivel a rendszer egy ponton érzékelhetően hibára futott, másféle megközelítéssel próbálkoztam: írtam egy emulátorprogramot, mely segítségével sikerült meghatároznom és kijavítanom a készként átvett CPU implementáció hibáját.

További teszteket követően számos egyéb hibát derítettem fel és javítottam ki. A félév végeztével sikerül elérnem, hogy a meghajtó képes legyen hibamentesen kommunikálni soros porton keresztül a valódi Commodore számítógéppel.

1.2. Rendszerterv

Első lépés az irodalomkutatás volt, valamint az elinduláshoz szükséges információk összegyűjtése. Az eredeti meghajtó áttervezéséhez nélkülözhetetlen megismerni, hogyan működik, milyen részekből épül fel az eszköz - ebben nagy segítségemre voltak a nyilvánosan elérhető dokumentációk, kézikönyvek.[11] Azért esett a választás épp a Commodore 1581 típusú floppy meghajtóra, mert ez ma is többé-kevésbé elérhető kis méretű 3.5"-es hajlékonylemezeket használ, belső felépítését tekintve pedig jól elkülöníthető részekre dekomponálható, szemben például a 1541-essel.



1.1. ábra. A Commodore 1581 típusú hajlékonylemezes meghajtó

A Commodore 1581-es meghajtó tulajdonképpen maga is egy "számítógép", hiszen a kiszolgálni kívánt számítógéppel való kommunikációt, és magát a lemezkezelést egy, a meghajtó alaplaján található diszkrét integrált áramkörtől felépített processzoros rendszer végzi. A 1581-es és a számítógép közötti kommunikáció egy Commodore IEEE-488 szabványú soros buszon zajlik.[16] A floppy lemezhez fizikailag egy belső író-olvasó egység segítségével lehetséges hozzáférni, mely egyébként szabványos, Amiga számítógépeknél is használatos ún. Shugart-interfészsel rendelkezik.[9] Ezzel szemben az általam megtervezett rendszerben a floppy író-olvasó egységet egy mai modern asztali IBM PC-vel kompatibilis floppy meghajtóval helyettesítettem, mivel ilyen berendezéseket ma sem lehetetlen beszerezni, szemben az eredeti konstrukcióban alkalmazott típusúval.

A szervizkönyv mellékleteként megtalálható a meghajtó eredeti kapcsolási rajza is, melyet a további tervezéshez az alábbi részekre bontottam:

- *Soros port:* a soros busz meghajtását és leválasztását szolgáló áramkörtől elemek
- *Floppy interfész:* a PC floppy meghajtó illesztéséhez szükséges bufferek, inverterek
- *FPGA:* az egész processzoros környezetet és minden más integrált áramkört elemet a szükséges kiegészítő logikákkal együtt ezen fogok megvalósítani

Így jól elkülöníthetővé váltak azok a részek, melyekhez kiegészítő áramkör fizikai elkészítése szükséges az egyes protokollok tulajdonságainak ismeretében.

1.3. Protokollok

Következő feladat annak az előkészítése volt, hogy az itt ismeretetett protokollokon szabványos módon történjen kommunikáció az alaplap és a PC floppy-meghajtó, valamint a számítógép között.

1.3.1. A Commodore IEEE-488 soros busz

A 8-bites Commodore asztali számítógépek az ún. Commodore IEEE-488 soros busz (vagy röviden IEC-busz) segítségével kommunikálnak a hozzájuk kapcsolt főbb perifériákkal, jellemzően nyomtatókkal vagy mágneses adathordozó alapú háttértárakkal. Eredetileg az IEEE-488 szabványból alakították ki a költségek csökkentése végett (innen az elnevezése). Ez alapvetően egy fél-duplex szinkron soros adatátviteli interfész, viszont számos speciális megoldást alkalmaz a szabvány pl. handshake vagy a buszra kapcsolódó eszközök jelenléte-érzékelésének megvalósítására.[16]

1.1. táblázat. Az IEC-busz jelei

Pin	Jelölés	Funkció
1	SRQ	Service Request In
2	GND	Ground
3	$\overline{\text{ATN}}$	Serial Attention In/Out
4	$\overline{\text{CLK}}$	Serial Clock In/Out
5	$\overline{\text{DATA}}$	Serial Data In/out
6	RESET	Serial Reset

Alacsony-aktív, nyitott kollektoros meghajtású vonalak ezek, 5V-os logikai jelszinttel. A számítógépez csatlakoztatott eszközöknek saját maguknak kell gondoskodniuk a tápellátásukról. Az egyes adatvonalakat mindkét végükön felhúzó ellenállások zárják le, a buszt nyitott kollektoros inverterek hajtják meg. Csatlakozó gyanánt 6 lábú DIN dugót alkalmaznak.

A soros buszra több eszköz is csatlakozhat, ezeket fizikailag daisy-chainelve lehetséges összekapcsolni egymással és a számítógéppel, emiatt szokásosan két csatlakozót építenek ki az egyes perifériákra. Az egyes eszközöket azonosítójuk alapján képes megcímezni a számítógép. Az adott periféria azonosítója DIP kapcsolók segítségével hardveresen konfigurálható, mely lemezmeghajtók esetén tipikusan 8 és 11 közötti egész értéket vehet fel.

A kommunikáció egységei 8-bites adatsomagok, melyekben a legkisebb helyi értékű bit található az első helyen. A mintavételezés szinkron módon, a $\overline{\text{CLK}}$ jel felfutó élére történik. Ezek az adatsomagok lehetnek vezérlő parancsok, vagy adatok. Vezérlő parancsokkal lehet a busz használatára vagy elengedésére utasítani az egyes perifériákat, vagy épp valamilyen irányú adatátvitelt kezdeményezni a megcímezett eszközzel. A vezérlő parancsok a buszon az $\overline{\text{ATN}}$ jel alacsony szintje mellett érkeznek mégpedig kizárólag a számítógép felől, innen lehet tudni, hogy nem egy általános adat átvitele történik.

A $\overline{\text{CLK}}$ és $\overline{\text{DATA}}$ vonalak kétirányúak, ezeket a floppy meghajtó is lehúzhatja. Ilyen módon jelzi a számítógépnek, hogy készen áll az adat fogadására vagy küldésére, de például bizonyos időkorláttal kiegészített nyugtázásra is használja ezt a módszert a szabvány.[10]

1.3.2. Floppy meghajtó

A feladat egyik nehézségét az adta, hogy a 1581-es meghajtó eredetileg másfajta floppy író-olvasót tartalmazott, mint amilyeneket az IBM-kompatibilis személyi számítógépekben használtak, használnak. Ez nem is a működési módjukban, sokkal inkább az IBM PC és a Shugart interfészek közti különbségben nyilvánul meg:[13]

1.2. táblázat. IBM/PC és Shugart interfésszel rendelkező hajlékonylemez-meghajtók lábkiosztása

IBM/PC			Shugart		
Pin	Jelölés	Funkció	Pin	Jelölés	Funkció
2	REDWC	Density Select	2	DCD	Disk Change Detect
4	n/c	Reserved	4	key	no pin in this position
6	n/c	Reserved	6	DS3	Drive Select 3
8	INDEX	Index	8	INDEX	Index
10	MOTEA	Motor Enable A	10	DS0	Drive Select 0
12	DRVSB	Drive Select B	12	DS1	Drive Select 1
14	DRVSA	Drive Select A	14	DS2	Drive Select 2
16	MOTEB	Motor Enable B	16	MTRON	Motor On
18	DIR	Direction	18	DIR	Direction
20	STEP	Step	20	STEP	Step
22	WDATE	Write Data	22	WDATE	Write Data
24	WGATE	Write Enable	24	WGATE	Write Enable
26	TRK00	Track 0	26	TRK00	Track 0
28	WPT	Write Protect	28	WPT	Write Protect
30	RDATA	Read Data	30	RDATA	Read Data
32	SIDE1	Head Select	32	SIDE1	Side Select
34	DSKCHG	Disk Change	34	RDY	Ready

Nem tüntettem fel a táblázatban, de mindenképp érdemes megjegyezni, hogy minden páratlan számú lábat a földre kötnék. Mivel a gyakorlatban szalagkábelben szokás továbbítani ezeket a jeleket, zavarvédelem szempontjából előnyös, ha az áthallás csökkentése végett nem jelvezetékek haladnak közvetlenül egymás mellett, hanem felváltva követik egymást a földvezetékekkel.

Hasonlóan a soros porthoz, mindkét fajta interfész esetén is 5V-os, open kollektoros meghajtású alacsony aktív jelekkel lehet találkozni, viszont nem előírás, hogy az eszköz tartalmazza a felhúzó ellenállásokat. Ezek PC esetén a számítógépben, jelen megoldásban pedig a 1581-es meghajtó alaplapján találhatók.

Megfigyelhető, hogy a jelek többsége megfeleltethető egymásnak.[6] PC esetén egy szalagkábelre két meghajtó kapcsolható, ezeket az A és B kiválasztójelek segítségével lehet megkülönböztetni egymástól, míg a Shugart interfészen akár 4 különböző eszköz is kiválasztható a \overline{DS} jelek megfelelőjének földre húzásával.

Szembevetendő különbség ezen kívül még, hogy a Disk Change jel máshol van kivezetve a két esetben, a Density Select valamint a Ready jeleknek pedig nincs megfelelője a másik csatlakozón.

A Density Select jellel kapcsolatban egymásnak ellentmondó információkat találtam kutatásom során, valahol kimenetként, máshol bemenetként használják, de mindenképp a meghajtóban használt floppy lemez adatsűrűségével hozható kapcsolatba. Ennek oka, hogy fizikailag máshogy kell kezelni egy SD, egy DD vagy egy HD hajlékonylemezt. Korszerű meghajtók ezt az adathordozó tokján található kivágás segítségével érzékelik, régi fajta (és

főleg 5.25"-es) meghajtók esetén viszont ezzel a Density Select jellel írható elő, milyen adatsebességgel dolgozzon a meghajtó vezérlője. Az általam használt PC floppy meghajtóban például ez a jel nincs is kivezetve.[12]

A Ready jel előállításához mindenképp kiegészítő logika megtervezése kellett, mivel a 1581-es kapcsolási rajza szerint szükség van rá. Az eredeti író-olvasó egység adatlapja alapján ez a jel akkor aktív, ha van lemez a meghajtóban és emellett megfelelő sebességgel forog a motor.[7] Ha forog a motor, fordulatonként egy impulzus mérhető az Index jelen. Az általam használt PC-s meghajtóban ez az impulzussorozat kapuzva van, és csak akkor kerül ki a kimenetre, miután a motor elérte a megfelelő fordulatszámot. Ennek a felhasználásával előállítható a Ready jel, hiszen hardveresen biztosított, hogy a motor csak akkor forog, ha van a meghajtóban lemez.

A Commodore 1581-es meghajtó két oldalú, 800kB kapacitású kétszeres adatsűrűségű (double-density, DD) lemezek kezelésére alkalmas. Az így formázott lemez oldalanként 80 sávot és 10 szektort tartalmaz fizikailag, a biteket MFM kódolással tárolja a rendszer. Mivel az így formázott lemez nem IBM kompatibilis, így a mai asztali számítógépekkel nem értelmezhető a tartalma. A feladatban használt PC floppy meghajtó reményeim szerint mégis tudja kezelni, hiszen az valójában csak alacsony szintű hozzáférést biztosít a lemezhez, a rajta lévő adatok feldolgozásáról az alaplapi processzoros rendszer gondoskodik.

2. fejezet

A hardver

2.1. Bemutatom a fejlesztőkártyát

A feladat megoldásához egy X-SP6-X9 jelzésű Xilinx Spartan 6 alapú FPGA-fejlesztőkártya állt rendelkezésemre. Számos szabadon felhasználható és széleskörűen konfigurálható IO kivezetésén kívül előnye, hogy tartalmazza az általános célú fejlesztésekhez szükséges perifériákat, nyomógombokat, kapcsolókat, LED-eket. Külső JTAG programozóval tölthető fel rá a bitstream, de egy 16MB-os SPI flash is helyet kapott, melyről szintén elvégezhető a konfiguráció. 5V-os tápellátással rendelkezik, viszont az IO lábak 3.3V-nak megfelelő CMOS logikai jelszinttel dolgoznak.

Mivel a soros busz és az író-olvasó egységként szolgáló PC floppy meghajtó eltérő logikai jelszinteket használ, mint az FPGA, így mindenképp szükséges a valóságban realizálni egy szintillesztő áramkört. Ezt egy interfészpanel formájában terveztem és valósítottam meg.

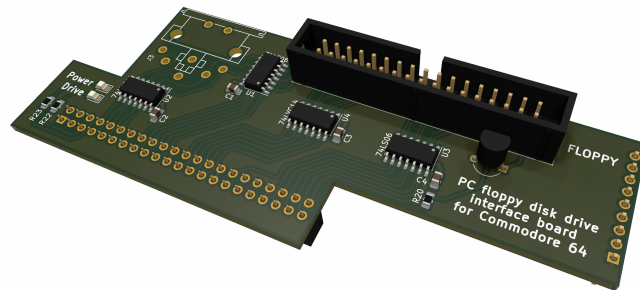
2.2. Az interfészpanel

Már a 1581-es kapcsolási rajzában is látható, hogy mind a soros port, mind a lemezmeghajtó esetén is meghajtófokozatokat, nyitott kollektoros buffereket vagy invertereket alkalmaztak, 1k Ω -os felhúzó ellenállásokkal kiegészítve. Ezeket az áramkörü elemeket a szintillesztésre való tekintettel a valóságban is realizálnom kellett.

Az interfészpanel tervezésekor az eredeti terveknek megfelelő tulajdonságú integrált áramkörü alkatrészeket választottam azzal a kritériummal kiegészítve, hogy a szükséges jelszint-illesztési feladatot is képesek legyenek ellátni. Gazdaságossági okokból, valamint az áramkör egyszerűsítése végett az összes kimeneti vonalat 74LS06 típusú nyitott kollektoros inverterrel, a bemeneteket 5V toleráns bemenetű 74LVC14 típusú CMOS inverterrel kapcsoltam a rendszerhez. Ebből a megoldásból ugyan az következik, hogy az eredeti kapcsolással ellentétben néhány jel így invertáltan áll rendelkezésre, de ezek visszaállítását az FPGA-n megvalósított rendszer implementálásakor figyelembe vettem.

Az illesztő áramkör kapcsolási rajza nem bonyolult, külön vettem rajta a soros porthoz és a floppy meghajtóhoz tartozó részeket, valamint helyet kapott rajta az eredeti 1581-es előlapján megtalálható két státuszjelző LED is. Az eredeti kapcsolási rajznak megfelelően a soros port kétirányú jeleit is különválasztottam.

Az áramkörnek két tápfeszültségre van szüksége, egy 3.3V-osra, és egy 5V-osra, mindkettőt az FPGA-fejlesztőpanelről biztosítottam. Ehhez rendelkezésemre állt a fejlesztőpanel kapcsolási rajza és lábkiosztása. A nyomtatott áramkört úgy terveztem meg, hogy közvetlenül illeszkedjen az FPGA-panel tüksoraira. A huzalozás és a lábkiosztás megtervezésekor törekedtem arra, hogy a lehető legkevesebb kereszteződéssel elférjenek egymás mellett a vezetősávok. Ergonómiai szempontokat is figyelembe vettem a tervezés során, hogy az FPGA-kártya kialakításának megfelelően maradjon hely a programozó csatlakozó, a meghajtó azonosítójának beállítására használt DIP kapcsolók és a RESET gomb számára, és ne takarja el az interfészpanel ezeket a felületeket, miközben rajta van. Hasonlóképp helyeztem el a floppy meghajtó csatlakoztatására szolgáló IDC csatlakozót és a soros port számára a 6 lábú Tuchel aljzatot. A mechanikai kialakításkor nehézséget okozott, hogy nem volt pontos műszaki rajz az FPGA-panel oldalsó tüksorának pontos helyéről, így azt becslés alapján tudtam csak pozicionálni, ez elég pontosan sikerült is.



2.1. ábra. *A saját tervezésű interfészpanel 3D terve*

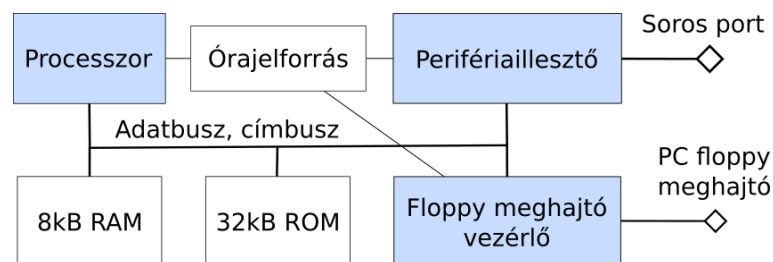
3. fejezet

Digitális tervezés

Az FPGA-ra a Xilinx ISE 14.3-as verziójú fejlesztői környezetének segítségével fejlesztettem a rendszert. Ez nem egy kimondottan korszerű ám szerencsére hibamentes szoftver. Azért ezt választottam mégis, mert a Spartan 6-os FPGA-kat a szoftver korszerű alternatívája (a Vivado) már nem támogatja. A leírófájlokat, modulokat és testbench-eket Verilog nyelven készítettem el, viszont helyenként szükséges volt VHDL-ben implementált részekkel is foglalkoznom később a hibakeresés során.

3.1. Modulok

A 1581-es meghajtót egy MOS 6502 típusú processzoron futó operációs rendszer irányítja. Az alaplapon megtalálható a firmware bináris állományát tartalmazó 32kB ROM és 8kB RAM. A perifériák kiszolgálását egy 8520-as komplex interfész adapter (CIA), a lemez író-olvasó egység vezérlését egy WD1772 típusú floppy meghajtó vezérlő (FDC) integrált áramkör végzi. Ezek képezik a főbb logikai egységeket, így kézenfekvő volt modulba szervezni őket, és így beilleszteni a projektbe.



3.1. ábra. Az FPGA-n megvalósított rendszer sematikus rajza

A rendszer 8-bites adatbuszt és 16 bites címbuszt használ, 2MHz-es rendszerórajelről jár minden a floppy vezérlő kivételével, mely 8MHz-es órajelet igényel. Ezen órajelek előállítása egy 4-bites számlálóval történt eredetileg, egy 16MHz-es oszcillátor jelének leosztásával. Érdekeség, hogy az ehhez alkalmazott 74LS93-as IC konfigurálható bitszámú: 3 vagy 4 bites számlálót is lehetséges kialakítani a segítségével. Egy ebben a témában végzett korábbi próbálkozás során a nem megfelelő konfiguráció jelenthetett hibalehetőséget.

Az eredeti kapcsolatban az egyes logikai egységekhez tartozó címdekódolást és engedélyező jelek előállítását multiplexerek végzik, a címtartomány az alábbiak szerint alakul:

3.1. táblázat. *Az 1581-es meghajtó processzoros rendszerének címtartomány-kiosztása*

Eszköz	Címtartomány
RAM	0x0000-0x1FFF
CIA	0x4000-0x4010
FDC	0x6000-0x6003
ROM	0x8000-0xFFFF

3.1.1. CPU

A 6502-es CPU 8-bites[14][?][?]

3.1.2. FDC

[15]

3.1.3. CIA

A 8520-as komplex interfész adapter IC tartalmaz két 8-bites portot, két 16-bites időzítőt, és egy valós idejű órát és egy dedikált soros port perifériát.[8]

különböző intellectual property-ket vadásztam össze innen-onnan (=korábbi munkák eredményét felhasználva) kiegészítő logikák milyen verilog modulok vannak, miket csináltam én

melyik alegység mit csinál, hogy működik, mi van bennne CIA: io bővítő + soros port + 2 timer FDC: floppy vezérlő xd CPU: 6502 (belső felépítése, stb -> assembly oldalról fontos inkább, lásd brk flag)

3.2. Tesztelés

4. fejezet

Tesztelés, hibajavítás

na ide jöhet minden

hogyan zajlott ez egyes modulok tesztje: órajel ram rom proci (speckó, asszimmetrikus órajelet kér)

tesztelés a valóságban: valódi C64 hardverrel (+hibajel kiolvasó szkript) + digitális analízátor + speckó kábelek floppy meghajtó ready logikája

proci bug -> emulátor szkript soros port hiba: órajel-forrás helyessége -> PLL drive nem ready -> hibás ucf fájl (dskchg és read láb megcserélve)

4.1. Címkék és hivatkozások

A L^AT_EX dokumentumban címkéket (`\label`) rendelhetünk ábrákhoz, táblázatokhoz, fejezetekhez, listákhoz, képletekhez stb. Ezekre a dokumentum bármely részében hivatkozhatunk, a hivatkozások automatikusan feloldásra kerülnek.

A sablonban makrókat definiáltunk a hivatkozások megkönnyítéséhez. Ennek megfelelően minden ábra (*figure*) címkéje **fig:** kulcsszóval kezdődik, míg minden táblázat (*table*), képlet (*equation*), fejezet (*section*) és lista (*listing*) rendre a **tab:**, **eq:**, **sect:** és **listing:** kulcsszóval kezdődik, és a kulcsszavak után tetszőlegesen választott címke használható. Ha ezt a konvenciót betartjuk, akkor az előbbi objektumok számára rendre a `\figref`, `\tabref`, `\eqref`, `\sectref` és `\listref` makrókkal hivatkozhatunk. A makrók paramétere a címke, amelyre hivatkozunk (a kulcsszó nélkül). Az összes említett hivatkozástípus, beleértve az `\url` kulcsszóval bevezetett web-hivatkozásokat is a `hyperref`¹ csomagnak köszönhetően aktívak a legtöbb PDF-nézegetőben, rájuk kattintva a dokumentum megfelelő oldalára ugrik a PDF-néző vagy a megfelelő linket megnyitja az alapértelmezett böngészővel. A `hyperref` csomag a kimeneti PDF-dokumentumba könyvjelzőket is készít a tartalomjegyzékből. Ez egy szintén aktív tartalomjegyzék, amelynek elemeire kattintva a nézegető behozza a kiválasztott fejezetet.

¹Segítségével a dokumentumban megjelenő hivatkozások nem csak dinamikussá válnak, de színezhetők is, bővebbet erről a csomag dokumentációjában találunk. Ez egyúttal egy példa lábjegyzet írására.

4.2. Ábrák és táblázatok

A képeket PDFLaTeX esetén a veszteségmentes PNG, valamint a veszteséges JPEG formátumban érdemes elmenteni. Az EPS (PostScript) vektorgrafikus képformátum beillesztését a PDFLaTeX közvetlenül nem támogatja. Ehelyett egy lehetőség 200 dpi, vagy annál nagyobb felbontásban raszterizálni a képet, és PNG formátumban elmenteni. Az egyes képek mérete általában nem, de sok kép esetén a dokumentum összmérete így már szignifikáns is lehet. A dokumentumban felhasznált képfájlokat a dokumentum forrása mellett érdemes tartani, archiválni, mivel ezek hiányában a dokumentum nem fordul újra. Ha lehet, a vektorgrafikus képeket vektorgrafikus formátumban is érdemes elmenteni az újrafelhasználhatóság (az átszerkeszthetőség) érdekében.

Kapcsolási rajzok legtöbbször kimásolhatók egy vektorgrafikus programba (pl. Corel-Draw) és onnan nagyobb felbontással raszterizálva kimenthetők PNG formátumban. Ugyanakkor kiváló ábrák készíthetők Microsoft Visio vagy hasonló program használatával is: Visio-ból az ábrák közvetlenül PNG-be is menthetők.

Lehetőségeink Matlab ábrák esetén:

- Képernyőlopás (*screenshot*) is elfogadható minőségű lehet a dokumentumban, de általában jobb felbontást is el lehet érni más módszerrel.
- A Matlab ábrát a **File/Save As** opcióval lementhetjük PNG formátumban (ugyanaz itt is érvényes, mint korábban, ezért nem javasoljuk).
- A Matlab ábrát az **Edit/Copy figure** opcióval kimásolhatjuk egy vektorgrafikus programba is és onnan nagyobb felbontással raszterizálva kimenthetjük PNG formátumban (nem javasolt).
- Javasolt megoldás: az ábrát a **File/Save As** opcióval EPS *vektorgrafikus* formátumban elmentjük, PDF-be konvertálva beillesztjük a dolgozatba.

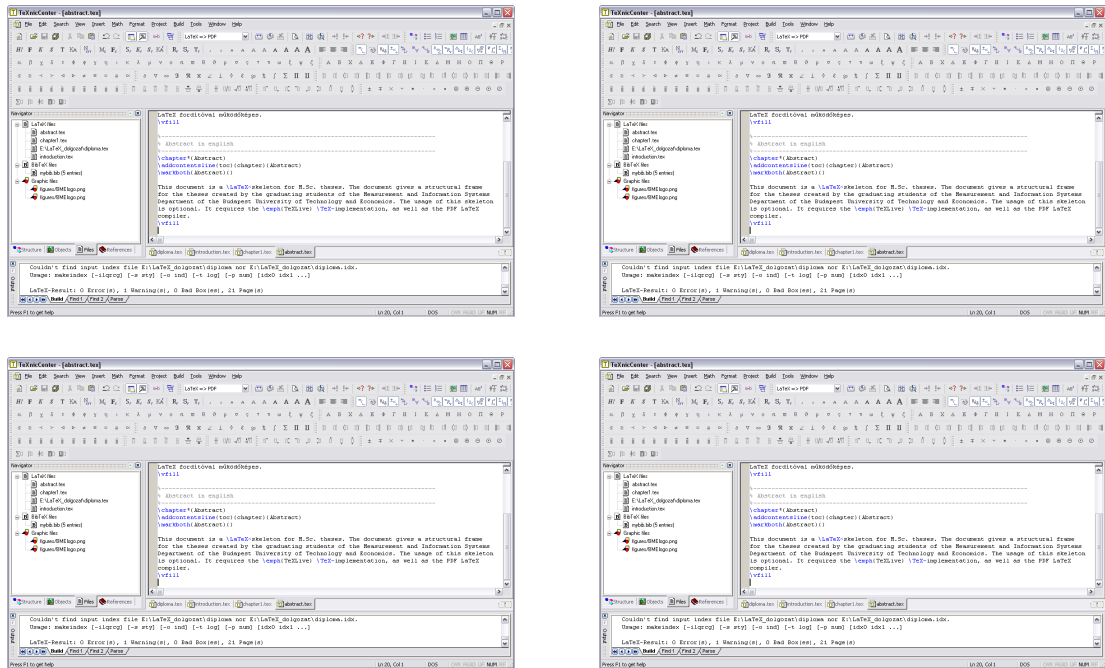
Az EPS kép az `epstopdf` programmal² konvertálható PDF formátumba. Célszerű egy batch-fájlt készíteni az összes EPS ábra lefordítására az alábbi módon (ez Windows alatt működik).

```
@echo off
for %%j in (*.eps) do (
echo converting file "%%j"
epstopdf "%%j"
)
echo done .
```

4.3. Irodalmi hivatkozások

Egy L^AT_EX dokumentumban az irodalmi hivatkozások definíciójának két módja van. Az egyik a `\thebibliography` környezet használata a dokumentum végén, az `\end{document}` lezárás előtt.

²a korábban említett L^AT_EX-disztribúciókban megtalálható



4.1. ábra. Több képfájl beillesztése esetén térközőket is érdemes használni.

```
\begin{thebibliography}{99}

\bibitem{Lamport94} Leslie Lamport, \emph{\LaTeX: A Document Preparation System}.
Addison Wesley, Massachusetts, 2nd Edition, 1994.

\end{thebibliography}
```

Ezek után a dokumentumban a `\cite{Lamport94}` utasítással hivatkozhatunk a forrásra. A fenti megadás viszonylag kötetlen, a szerző maga formázza az irodalomjegyzéket.

Egy sokkal professzionálisabb módszer a BiBTeX használata, ezért ez a sablon is ezt támogatja. Ebben az esetben egy külön szöveges adatbázisban definiáljuk a forrásmunkákat, és egy külön stílusfájl határozza meg az irodalomjegyzék kinézetét. Ez, összhangban azzal, hogy külön formátumkonvenció határozza meg a folyóirat-, a könyv-, a konferenciacyklistb. hivatkozások kinézetét az irodalomjegyzékben (a sablon használata esetén ezzel nem is kell foglalkoznia a hallgatónak, de az eredményt célszerű ellenőrizni). A felhasznált hivatkozások adatbázisa egy `.bib` kiterjesztésű szöveges fájl, amelynek szerkezetét a 4.1. kódrészlet demonstrálja. A forrásmunkák bevitelkor a sor végi vesszők külön figyelmet igényelnek, mert hiányuk a BiBTeX-fordító hibaüzenetét eredményezi. A forrásmunkákat típus szerinti kulcsszó vezeti be (`@book` könyv, `@inproceedings` konferenciakiadványban megjelent cikk, `@article` folyóiratban megjelent cikk, `@techreport` valamelyik egyetem gondozásában megjelent műszaki tanulmány, `@manual` műszaki dokumentáció esetén stb.). Nemcsak a megjelenés stílusa, de a kötelezően megadandó mezők is típusról-típusra változnak. Egy jól használható referencia a <http://en.wikipedia.org/wiki/BibTeX> oldalon található.

A stílusfájl egy `.sty` kiterjesztésű fájl, de ezzel lényegében nem kell foglalkozni, mert vannak beépített stílusok, amelyek jól használhatók. Ez a sablon a BiBTeX-et használja, a hozzá tartozó adatbázisfájl a `mybib.bib` fájl. Megfigyelhető, hogy az irodalomjegyzéket a

4.1. lista. Példa szöveges irodalomjegyzék-adatbázisra BiBTeX használata esetén.

```
@BOOK{Wettl04,
  author="Ferenc Wettl and Gyula Mayer and Péter Szabó",
  title="\LaTeX~kézikönyv",
  publisher="Panem Könyvkiadó",
  year=2004
}
@ARTICLE{Candy86,
  author = "James C. Candy",
  title = "Decimation for Sigma Delta Modulation",
  journal="{IEEE} Trans.\ on Communications",
  volume = 34,
  number = 1,
  pages = "72--76",
  month = jan,
  year = 1986,
}
@INPROCEEDINGS{Lee87,
  author = "Wai L. Lee and Charles G. Sodini",
  title = "A Topology for Higher Order Interpolative Coders",
  booktitle = "Proc.\ of the IEEE International Symposium on
Circuits and Systems",
  year = 1987,
  vol = 2,
  month = may # "~4--7",
  address = "Philadelphia, PA, USA",
  pages = "459--462"
}
@PHDTHESIS{KissPhD,
  author = "Peter Kiss",
  title = "Adaptive Digital Compensation of Analog Circuit Imperfections
for Cascaded Delta-Sigma Analog-to-Digital Converters",
  school = "Technical University of Timi\c{s}oara, Romania",
  month = apr,
  year = 2000
}
@MANUAL{Schreier00,
  author = "Richard Schreier",
  title = "The Delta-Sigma Toolbox v5.2",
  organization = "Oregon State University",
  year = 2000,
  month = jan,
  note = "\newline URL: http://www.mathworks.com/matlabcentral/fileexchange/"
}
@MISC{DipPortal,
  author="Budapesti {M}űszaki és {G}azdaságtudományi {E}gyetem
{V}illamosmérnöki és {I}nformatikai {K}ar",
  title="{D}iplomaterv portál (2011 február 26.)",
  howpublished="\url{http://diplomaterv.vik.bme.hu/}",
}}

```

dokumentum végére (a `\end{document}` utasítás elé) beillesztett `\bibliography{mybib}` utasítással hozhatjuk létre, a stílusát pedig ugyanitt a `\bibliographystyle{plain}` utasítással adhatjuk meg. Ebben az esetben a `plain` előre definiált stílust használjuk (a sablonban is ezt állítottuk be). A `plain` stíluson kívül természetesen számtalan más előre definiált stílus is létezik. Mivel a `.bib` adatbázisban ezeket megadtuk, a BiBTeX-fordító is meg tudja különböztetni a szerzőt a címtől és a kiadótól, és ez alapján automatikusan generálódik az irodalomjegyzék a stílusfájl által meghatározott stílusban.

Az egyes forrásmunkákra a szövegből továbbra is a `\cite` paranccsal tudunk hivatkozni, így a 4.1. kódrészlet esetén a hivatkozások rendre `\cite{Wettl04}`, `\cite{Candy86}`, `\cite{Lee87}`, `\cite{KissPhD}`, `\cite{Schreier00}` és `\cite{DipPortal}`. Az irodalomjegyzékben alapértelmezésben csak azok a forrásmunkák jelennek meg, amelyekre ta-

lálható hivatkozás a szövegben, és ez így alapvetően helyes is, hiszen olyan forrásmunkákat nem illik az irodalomjegyzékbe írni, amelyekre nincs hivatkozás.

Mivel a fordítási folyamat során több lépésben oldódnak fel a szimbólumok, ezért gyakran többször (TeXLive és TeXnicCenter esetén 2-3-szor) is le kell fordítani a dokumentumot. Ilyenkor ez első 1-2 fordítás esetleg szimbólum-feloldásra vonatkozó figyelmeztető üzenettel zárul. Ha hibaüzenettel zárul bármelyik fordítás, akkor nincs értelme megismételni, hanem a hibát kell megkeresni. A `.bib` fájl megváltoztatáskor sokszor nincs hatása a változtatásnak azonnal, mivel nem mindig fut újra a BibTeX fordító. Ezért célszerű a változtatás után azt manuálisan is lefuttatni (TeXnicCenter esetén **Build/BibTeX**).

Hogy a szövegbe ágyazott hivatkozások kinézetét demonstráljuk, itt most sorban meghivatkozzuk a [1], [2], [3], [4] és az [?] forrásmunkát, valamint az [5] weboldalt.

Megjegyzendő, hogy az ékezetes magyar betűket is tartalmazó `.bib` fájl az `inputenc` csomaggal betöltött `latin2` betűkészlet miatt fordítható. Ugyanez a `.bib` fájl hibaüzenettel fordul egy olyan dokumentumban, ami nem tartalmazza a `\usepackage[latin2]{inputenc}` sort. Speciális igény esetén az irodalmi adatbázis általánosabb érvényűvé tehető, ha az ékezetes betűket speciális latex karakterekkel helyettesítjük a `.bib` fájlban, pl. `á` helyett `\'{a}`-t vagy `ő` helyett `\H{o}`-t írunk.

Oldaltörés következik (ld. forrás).

4.4. A dolgozat szerkezete és a forrásfájlok

A diplomatervsablon (a kari irányelvek szerint) az alábbi fő fejezetekből áll:

1. 1 oldalas *tájékoztató* a szakdolgozat/diplomaterv szerkezetéről (`guideline.tex`), ami a végső dolgozathoz törlendő,
2. *feladatkiírás* (`project.tex`), a dolgozat nyomtatott verzójában ennek a helyére kerül a tanszék által kiadott, a tanszékvezető által aláírt feladatkiírás, a dolgozat elektronikus verziójába pedig a feladatkiírás egyáltalán ne kerüljön bele, azt külön tölti fel a tanszék a diplomaterv-honlapra,
3. *címoldal* (`titlepage.tex`),
4. *tartalomjegyzék* (`diploma.tex`),
5. a diplomatervező *nyilatkozata* az önálló munkáról (`declaration.tex`),
6. 1-2 oldalas tartalmi *összefoglaló* magyarul és angolul, illetve elkészíthető még további nyelveken is (`abstract.tex`),
7. *bevezetés*: a feladat értelmezése, a tervezés célja, a feladat indokoltsága, a diplomaterv felépítésének rövid összefoglalása (`introduction.tex`),
8. sorszámmal ellátott *fejezetek*: a feladatkiírás pontosítása és részletes elemzése, előzmények (irodalomkutatás, hasonló alkotások), az ezekből levonható következtetések, a tervezés részletes leírása, a döntési lehetőségek értékelése és a választott megoldások indoklása, a megtervezett műszaki alkotás értékelése, kritikai elemzése, továbbfejlesztési lehetőségek (`chapter{1,2..n}.tex`),
9. esetleges *köszönetnyilvánítások* (`acknowledgement.tex`),
10. részletes és pontos *irodalomjegyzék* (ez a sablon esetében automatikusan generálódik a `diploma.tex` fájlban elhelyezett `\bibliography` utasítás hatására, a 4.3. fejezetben leírtak szerint),
11. *függelékek* (`appendices.tex`).

A sablonban a fejezetek a `diploma.tex` fájlba vannak beillesztve `\include` utasítások segítségével. Lehetőség van arra, hogy csak az éppen szerkesztés alatt álló `.tex` fájlt fordítsuk le, ezzel lerövidítve a fordítási folyamatot. Ezt a lehetőséget az alábbi kódrészlet biztosítja a `diploma.tex` fájlban.

Ha az alábbi kódrészletben az egyes sorokat a `%` szimbólummal kikommentezzük, akkor a megfelelő `.tex` fájl nem fordul le. Az oldalszámok és a tartalomjegyzék természetesen csak akkor billennek helyre, ha a teljes dokumentumot lefordítjuk.

```

\includeonly{
    guideline,%
    project,%
    titlepage,%
    declaration,%
    abstract,%
    introduction,%
    chapter1,%
    chapter2,%
    chapter3,%
    acknowledgement,%
    appendices,%
}

```

4.5. Alapadatok megadása

A diplomaterv alapadatait (cím, szerző, konzulens, konzulens titulusa) a `diploma.tex` fájlban lehet megadni az alábbi kódrészlet módosításával.

```

\newcommand{\vikszerzo}{Bódis-Szomorú András}
\newcommand{\vikkonzulens}{dr.~Konzulens Elemér}
\newcommand{\vikcim}{Elektronikus terelők}
\newcommand{\viktanszek}{Méréstechnika és Információs Rendszerek Tanszék}
\newcommand{\vikdoktipus}{Diplomaterv}
\newcommand{\vikdepartmentr}{Bódis-Szomorú András}

```

4.6. Új fejezet írása

A főfejezetek külön `chapter{1..n}.tex` fájlban foglalnak helyet. A sablonhoz 3 fejezet készült. További főfejezeteket úgy hozhatunk létre, ha új `chapter{i}.tex` fájlt készítünk a fejezet számára, és a `diploma.tex` fájlban, a `\include` és `\includeonly` utasítások argumentumába felvesszük az új `.tex` fájl nevét.

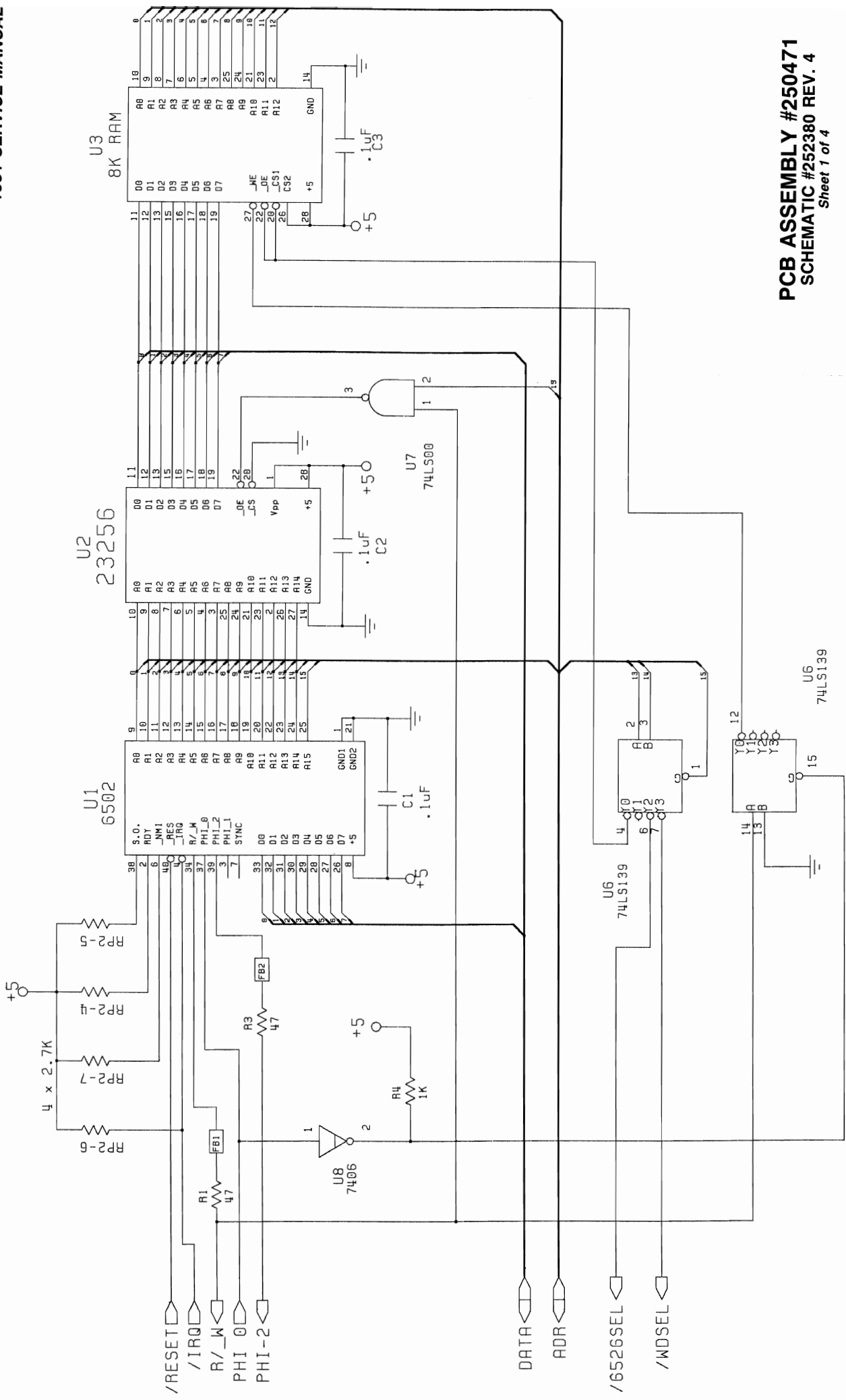
Irodalomjegyzék

- [1]
- [2]
- [3]
- [4] PhD thesis.
- [5]
- [6] Thomas Brase. Floppy disk drive / bus interface. <https://retrocmp.de/fdd/general/floppy-bus.htm>, 2022. [Online; accessed 15-dec-2021].
- [7] CHINON INDUSTRIES, INC. *Chinon F-354C 135TPI Double-Sided 3.5 In Specifications*. URL: http://www.bitsavers.org/pdf/chinon/Chinon_F-354C_135TPI_Double-Sided_3.5_In_Specifications.pdf.
- [8] COMMODORE SEMICONDUCTOR GROUP, Postscript-conversion by Markku Alén. *6526 COMPLEX INTERFACE ADAPTER (CIA)*, February 2001. URL: http://archive.6502.org/datasheets/mos_6526_cia_recreated.pdf.
- [9] NEC Corporation. *FDI036A 3.5" FLOPPY DISK DRIVE PRODUCT DESCRIPTION*, 1985. URL: <https://raw.githubusercontent.com/MiSTer-devel/AtariST-MiSTer/master/rtl/fdc1772/NEC%20FD1036%20Floppy.pdf>.
- [10] J. Derogee. *IEC dissected - IEC-bus documentation as used for the 1541-III*, February 2008. URL: <http://www.zimmers.net/anonftp/pub/cbm/programming/serial-bus.pdf>.
- [11] Commodore Electronics Limited. *SERVICE MANUAL 1581 3.5 DISK DRIVE*, June 1987. URL: [http://www.zimmers.net/anonftp/pub/cbm/schematics/drives/new/1581/1581_Service_Manual_314982-01_\(1987_Jun\).pdf](http://www.zimmers.net/anonftp/pub/cbm/schematics/drives/new/1581/1581_Service_Manual_314982-01_(1987_Jun).pdf).
- [12] Sony Corporation. *MPF920-Z 3 1/2" Floppy Disk Drive*, 2004. URL: <https://pro.sony/s3/cms-static-content/operation-manual/4668278111.pdf>.
- [13] Sven Olaf Kamphuis, Malvineous, Kikinou, Peter Bye, archyx et. al. Floppy disk-drive pinout and wiring. https://old.pinouts.ru/HD/InternalDisk_pinout.shtml, 2019. [Online; accessed 15-dec-2022].

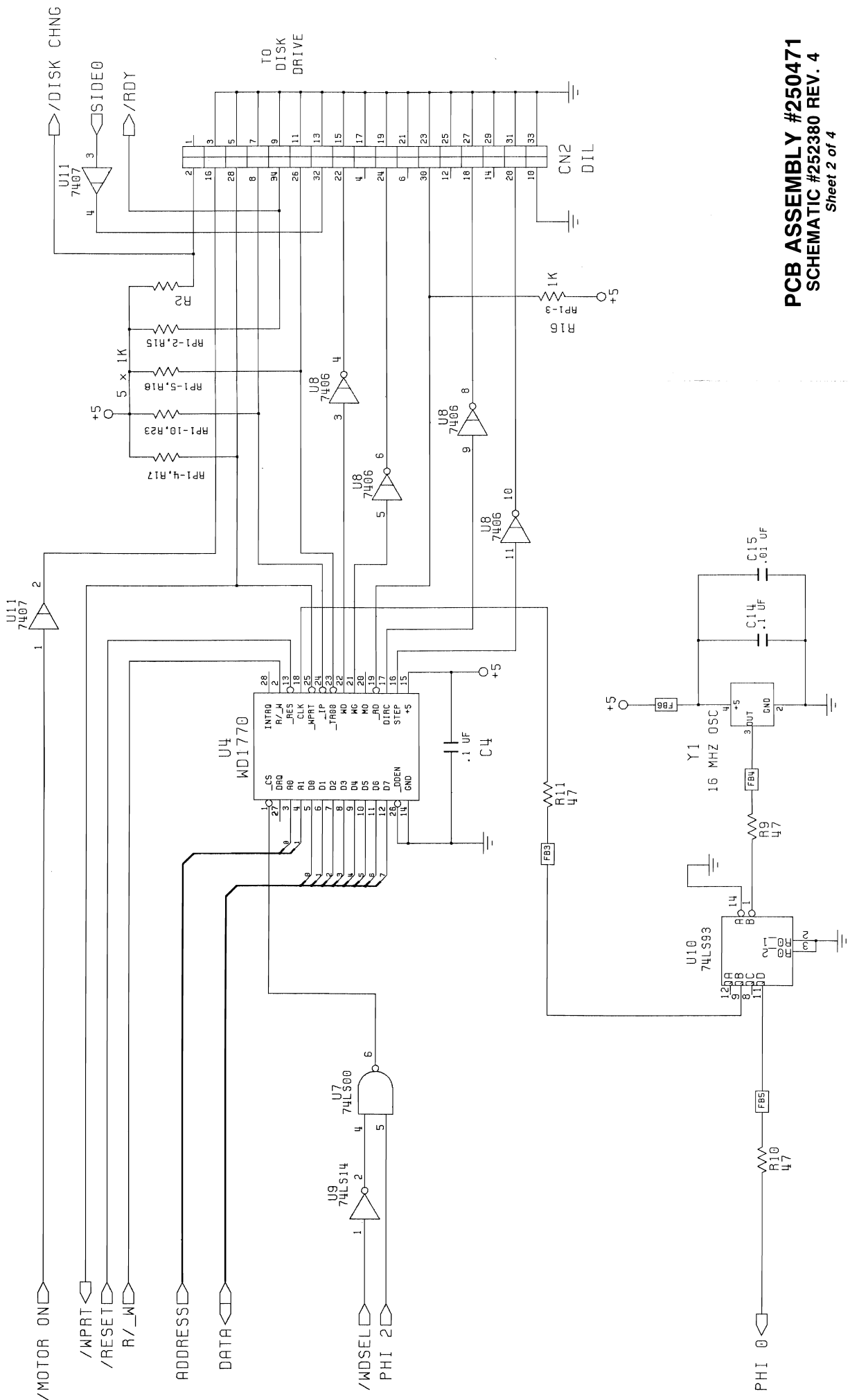
- [14] Synertek. *SY6500 8-Bit Microprocessor Family*. URL: <https://www.princeton.edu/~mae412/HANDOUTS/Datasheets/6502.pdf>.
- [15] WESTERN DIGITAL CORPORATION, Edited By Jean Louis-Guérin. *WD1772 Floppy Disk Formatter/Controller*, January 2015. URL: http://info-coach.fr/atari/documents/_mydoc/WD1772-JLG.pdf.
- [16] Wikipedia contributors. Commodore bus — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Commodore_bus, 2022. [Online; accessed 04-dec-2022].

Függelék

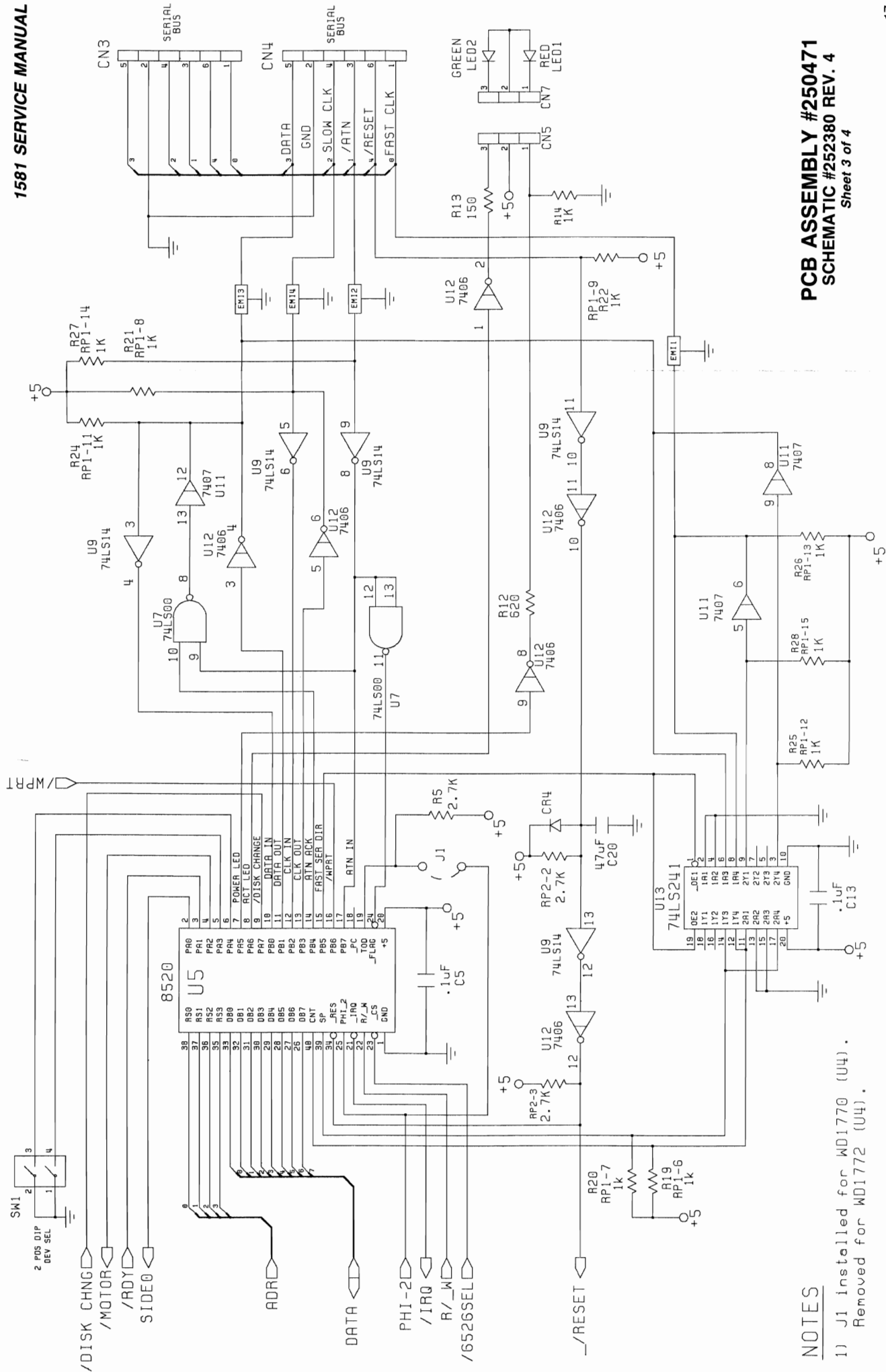
F.1. Az eredeti 1581-es floppy meghajtó kapcsolási rajza



PCB ASSEMBLY #250471
SCHEMATIC #252380 REV. 4
Sheet 1 of 4

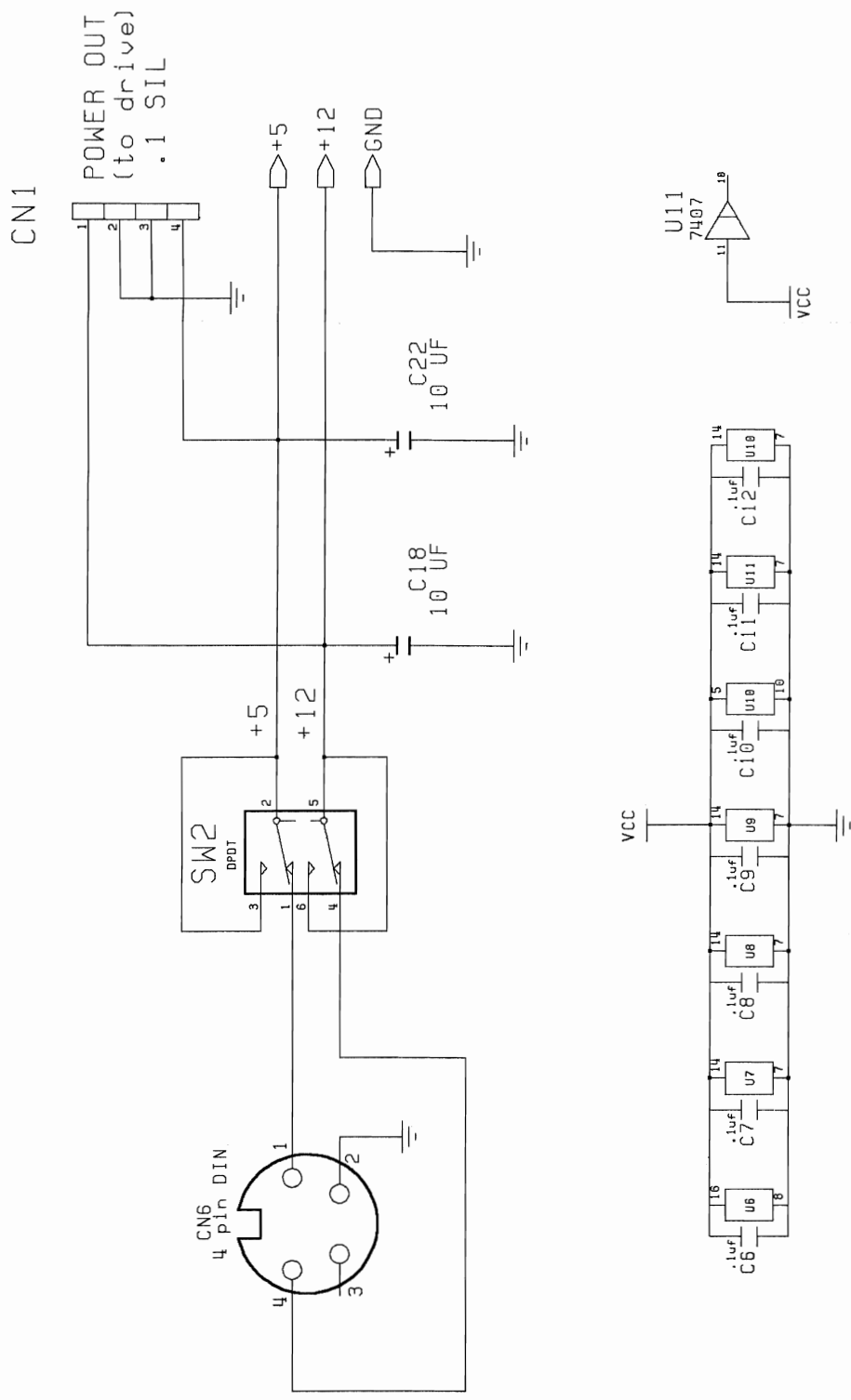


PCB ASSEMBLY #250471
SCHEMATIC #252380 REV. 4
Sheet 2 of 4

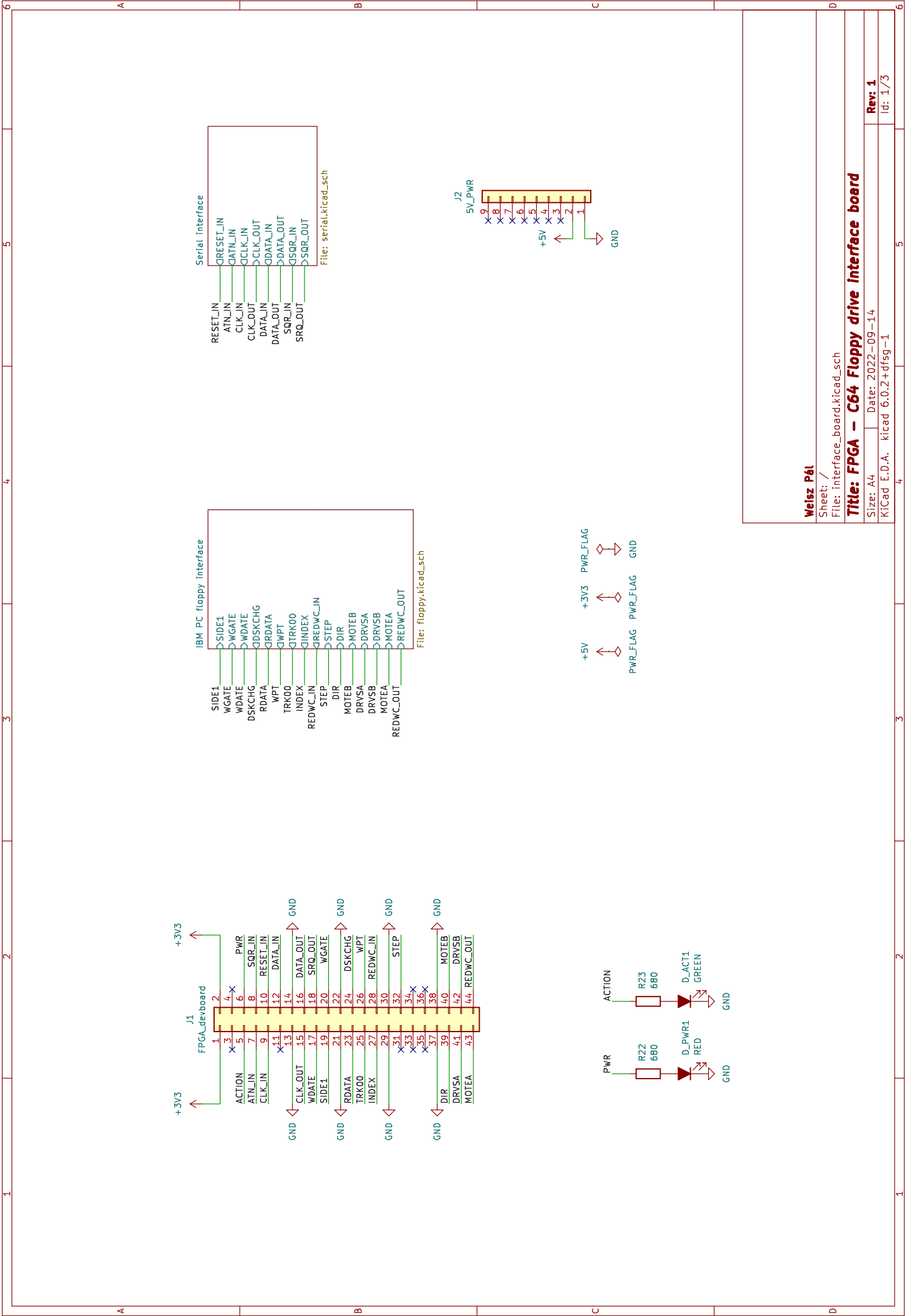


NOTES

- 1) J1 installed for WD1770 (U4).
Removed for WD1772 (U4).



F.2. Az interfészkártya kapcsolási rajza



Welsz Pál

Sheet: /

File: interface_board.kicad_sch

Title: FPGA – C64 Floppy drive interface board

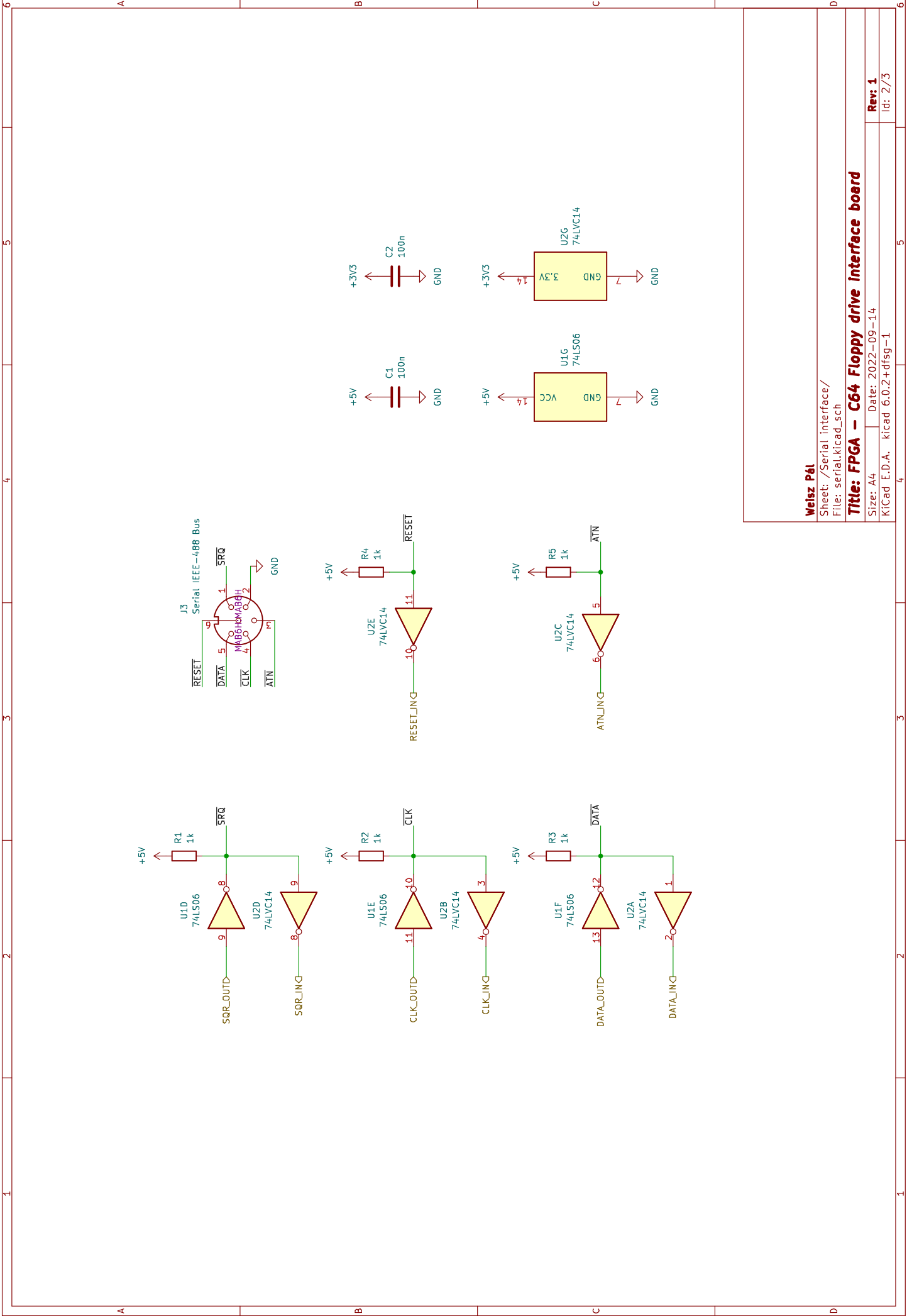
Size: A4

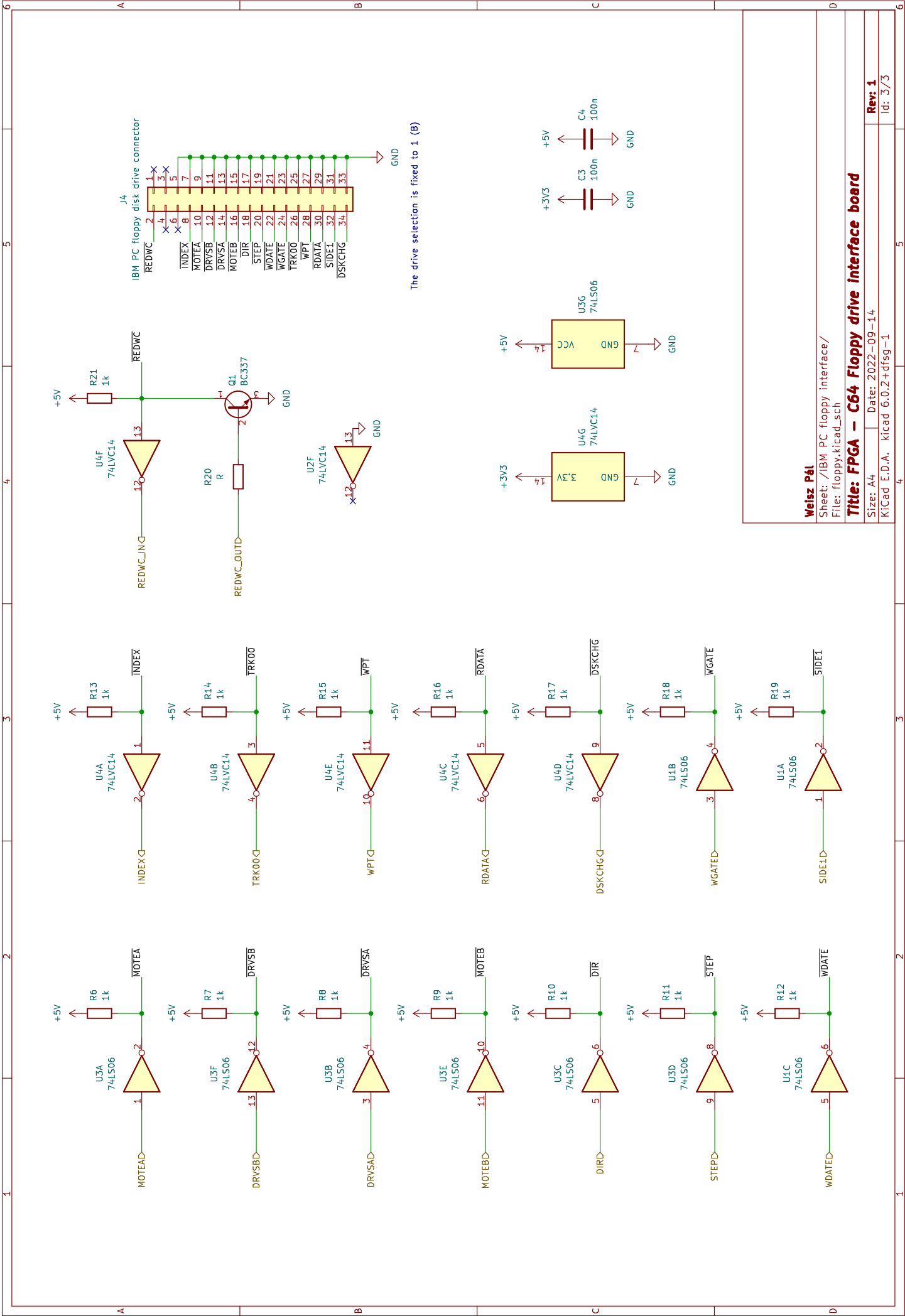
Date: 2022-09-14

KiCad E.D.A. kicad 6.0.2 + dfsg-1

Rev: 1

Id: 1/3





Welsz Pál

Sheet: /IBM PC floppy interface/
File: floppy.kicad_sch

Title: FPGA - C64 Floppy drive Interface board

Size: A4

Date: 2022-09-14

KiCad E.D.A. kicad 6.0.2 + dfsg - 1

Rev: 1

Id: 3/3

F.3. Az interfészkártya NYÁK-terve

