



M Ű E G Y E T E M 1 7 8 2

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

Szekeres Balázs

LED MÁTRIX KIJELZŐ TERVEZÉSE

KONZULENSEK

Dr. Benesóczky Zoltán

Sándor József Benedek

BUDAPEST, 2015



DIPLOMATERV-FELADAT

Szekeres Balázs (IYMZ38)

szigorló villamosmérnök hallgató részére

LED-es mátrix kijelző tervezése

A feladat egy egyszínű LED mátrix kijelző fejlesztése. A kijelző RS 485 interfészen érkező adatok megjelenítését tegye lehetővé. A megjelenített információk kb. 40 m távolságból is legyenek láthatóak. Gondolja át a tápellátás problémáját, de annak megtervezése nem része a feladatnak.

1. Végezzen irodalomkutatást a hasonló LED mátrix kijelzők és az erre a célra használható LED vezérlők között!
2. Készítse el a megtervezendő készülék részletes specifikációját!
3. Készítse el a készülék hardver és szoftver rendszertervét, továbbá kapcsolási rajzát! Válassza fel a különféle alternatívákat, a fontosabb tervezői döntéseket indokolja meg részletesen!
4. Írja meg a szoftver legfontosabb részeit!
5. Elemezze a megtervezett rendszert és adjon ötleteket, hogy miképpen lehetne továbbfejleszteni!

Tanszéki konzulens: Dr. Benesóczky Zoltán, mestertanár

Külső konzulens: Sándor József Benedek (Körös Consult Kft.)

Budapest, 2014. október 20.

.....
Dr. Jobbágy Ákos
tanszékvezető

Tartalomjegyzék

Összefoglaló	6
Abstract.....	7
1 Bevezetés	8
2 Irodalomkutatás.....	9
2.1 LED technológia rövid bemutatása.....	9
2.2 LED meghajtó áramkörök.....	11
2.3 Hasonló célú LED mátrix kijelzők	16
3 Specifikáció.....	18
3.1 A kijelzővel szemben támasztott követelmények	18
3.2 A kijelző paraméterei.....	19
4 Rendszerterv.....	20
4.1 Hardver rendszerterv.....	20
4.1.1 LED Panel.....	22
4.1.2 Vezérlő Board	27
4.1.3 Az FPGA hardverterve	30
4.1.4 Az FPGA belső rendszerterve.....	31
5 Hardvertervezés	32
5.1 Felhasznált hardverkomponensek	32
5.1.1 Spartan 6 FPGA	32
5.1.2 Raspberry Pi B+	33
5.2 LED-panelek tervezése	34
5.3 Vezérlő board tervezése.....	39
5.3.1 Tápáramkör.....	40
5.3.2 FPGA kapcsolási rajzának terve	43
5.3.3 Kimeneti csatlakozó a LED panelek felé.....	44
5.3.4 JTAG (Joint Tag Action Group) csatlakozó	45
5.3.5 SPI FLASH memória.....	45
5.3.6 Raspberry Pi B+ csatlakozó.....	46
5.4 FGPA hardverterve	47
5.4.1 Az SPI interfész bemutatása	47

5.4.2 A fejlesztés folyamata.....	48
5.4.3 FPGA modulok tervezése	53
5.4.4 Panelvezérlés tesztelése	59
5.5 Kiegészítő elemek tervezése	60
5.5.1 Tápellátás tervezése	60
5.5.2 Kábelezés kialakítása	61
6 Szoftvertervezés	62
7 Összefoglalás.....	64
8 Irodalomjegyzék.....	66
9 Függelék.....	67

HALLGATÓI NYILATKOZAT

Alulírott **Szekeres Balázs**, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2015. 12. 15.

.....
Szekeres Balázs

Összefoglaló

A diplomadolgozat készítése során specifikáltam a megtervezett LED-mátrix kijelző paramétereit. A rendelkezésre álló hely, pixelszám és költségkeret figyelembevételével döntés született a LED kijelző felbontásáról. A felbontás 60 x 80 pixel lett és ez optimálisan 1.49 x 1.98 m nagyságú kijelzőt eredményezett. A specifikáció kijelző monokróm, grafikus és karakteres adatokat képes megjeleníteni.

A külsős konzulenssel meghatároztuk a rendszer funkcionális felépítését. Több lehetőség figyelembevételével megterveztük a rendszer főbb alkotórészeit, a vezérlő egységeket és a kommunikációs interfészeket. A végleges beágyazott számítógép egy Raspberry Pi B+ lett, amely Etherneten fogadja a kívülről érkező adatokat és SPI-on keresztül küldi tovább őket a Spartan 6-os FPGA-nak. Az FPGA egy FIFO-ként funkcionál és a kapott sorrendben tovább küldi az adatokat a LED-paneleken található TLC 5955-es LED meghajtó áramköröknek. Egy 297 x 396 mm-es panelen 4 darab LED vezérlő IC található és ezekhez 192 darab zöldszínű LED tartozik. A teljes kijelző 25 darab panelből áll.

Az FPGA belső felépítése moduláris. Ennek a módszernek az előnye, hogy amint a LED-panel vezérlőmodul elkészült, tetszőlegesen lehetett skálázni a tervet. Öt modult készítettem el: az SPI Master modult, az SPI Slave modult, a központi modult, az inicializáló modult és a blokk RAM-ot. Ezek a modulok valósítják meg az adatátvitelt a beágyazott számítógép és a LED-panelek között. A projekt során elkészítettem egy szoftvert a Raspberry Pi-ra, amely egy pufferből átküldi az adatokat az FPGA-ba.

Abstract

For my thesis I specified the parameters of the designed LED matrix display. Considering the available place, the pixel number and the budget a decision was made about the resolution of the LED display. The final resolution was 60 x 80 pixel, which resulted in a 1.49 x 1.98 m display. According to the specifications, the display is monochrome and able to visualize characters and graphics.

After cooperation with the firm's consultant we defined the functional structure of the system. After considering some possibilities, we designed the main elements of the system, for example the controller unit and the interfaces of communication. We chose a Raspberry Pi B+ model for embedded computer. This unit receives the information via Ethernet and sends it to the Spartan 6 FPGA via SPI. The FPGA is modelled as a FIFO, which transmits the received data to the TLC 5955 LED driver ICs. These are located on the LED panels. Each panel contains 4 LED drivers and 192 green LEDs. One panel is 297 x 396 mm and the whole display consists of 25 of them.

The inner structure of the FPGA is built up by modules. This modular structure makes the system easily scalable. I designed 5 modules, an SPI Master module, an SPI Slave module, an initialization module, a central module and a block RAM. These modules are responsible for the communication between the embedded computer and the LED-driver ICs. I also made a software for the Raspberry Pi, which transmits data to the FPGA from a predefined buffer.

1 Bevezetés

A Budapesti Műszaki- és Gazdaságtudományi Egyetemen folytatott villamosmérnöki MSc-s tanulmányaim során a diplomatervem elkészítését egy külsős cégnél, a Körös Consult Kft-nél végeztem. A cégnél több mint 50 mérnök dolgozik, fő profiljuk az egyedi igényeket kielégítő, magasfokú mérnöki szolgáltatás biztosítása.

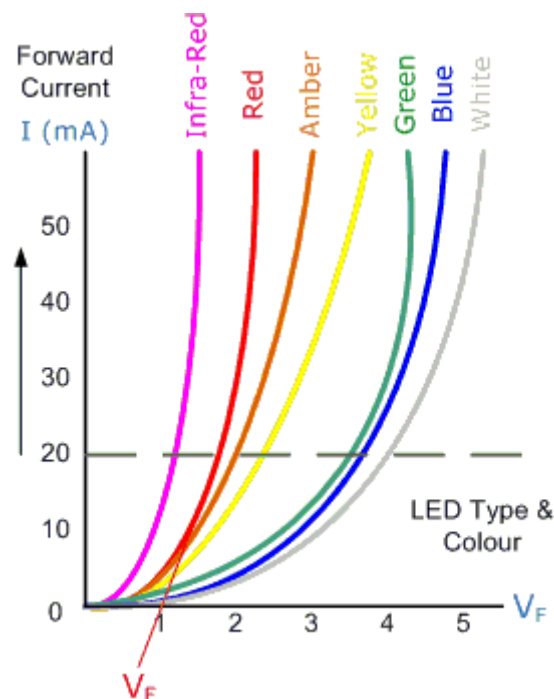
Az általam végzett projekt célja egy egyszínű LED-es mátrixkijelző kijelző tervezése és dokumentálása. Ez a kijelző egy erőmű külső oldalán kerül elhelyezésre. Feladata a termeléssel kapcsolatos karakteres és grafikus adatok megjelenítése. A grafikus adatok alatt termelési chartok, valamint különböző animációk megjelenítésére kell felkészülni.

Az erőmű egy folyó partján helyezkedik el. A vezérlést a túlpartról valósítják meg, ezért olyan eszközt kell tervezni, ahol a megjelenített adatok messziről is olvashatóak legyenek.

2 Irodalomkutatás

2.1 LED technológia rövid bemutatása

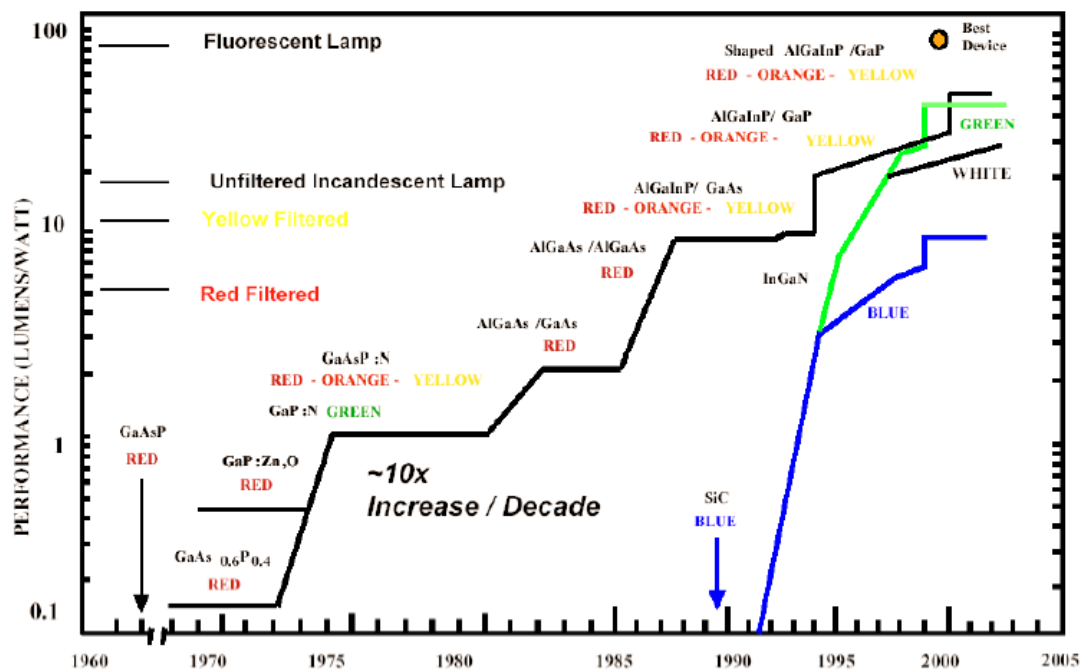
A XX. századi világítás technikában komoly áttörést jelentett a világítódiodák, közismert nevükön a LED-ek kifejlesztése. A LED az angol Light Emitting Diode kifejezés rövidítése. Bár az első LED-et már a 70-as években sikerült a tudósoknak kifejlesztetniük, a technológia csak a 90-es években vált versenyképesé a hagyományos fényforrásokkal. A LED belsejében egy elektrolumineszcencia nevű jelenség játszódik le. Az elektronok elektromos térrel gerjesztődnek, majd a többletenergiát sugárzás formájában adják le. A felvezető „p-n” átmenetében történő töltéshordozó által keltett elektrolumineszcencia az alapja a fénykibocsátó diódáknak és a félvezetőlézereknek egyaránt. Mivel az előbb említett két eszköz a tiltott sávjának megfelelő hosszúságú fényt bocsát ki, a félvezető fényforrások színe erősen függ a félvezető összetételétől. Az alábbi ábrán látható, hogy a LED-ek villamos működés szempontjából valóban dióda karakterisztikájúak.



1. ábra: Az eltérő színű LED-ek karakterisztikái ^[1]

A technológia használatával nem csak a látható tartományban sugárzó LED-eket lehet létrehozni, hanem olyanokat is, amelyek az infra- vagy az UV-tartományban

emittálnak fényt. Mivel a LED-ek keskeny sávban sugároznak, közel monokromatikus színt emittálnak, emiatt önmagukban nem alkalmasak fehér fény kibocsátására. Erre a problémára több megoldás is létezik, az egyik, hogy kék LED-et sárga fénnyel kombinálnak be. A LED-es világítótest sok szempontból jobb, mint a hagyományos volfrám izzószálalás égő. Legnagyobb előnye az energiatakarékosság, hiszen a fénykibocsátás nem egy fémzalag izzításával, hanem elektronok gerjesztésének segítségével történik. Ebből adódik a kisebb hődisszipáció, valamint az is hogy vibráció nélkül működik, ezért kíméli az emberi szemet. Előnye még a hosszabb élettartam is.

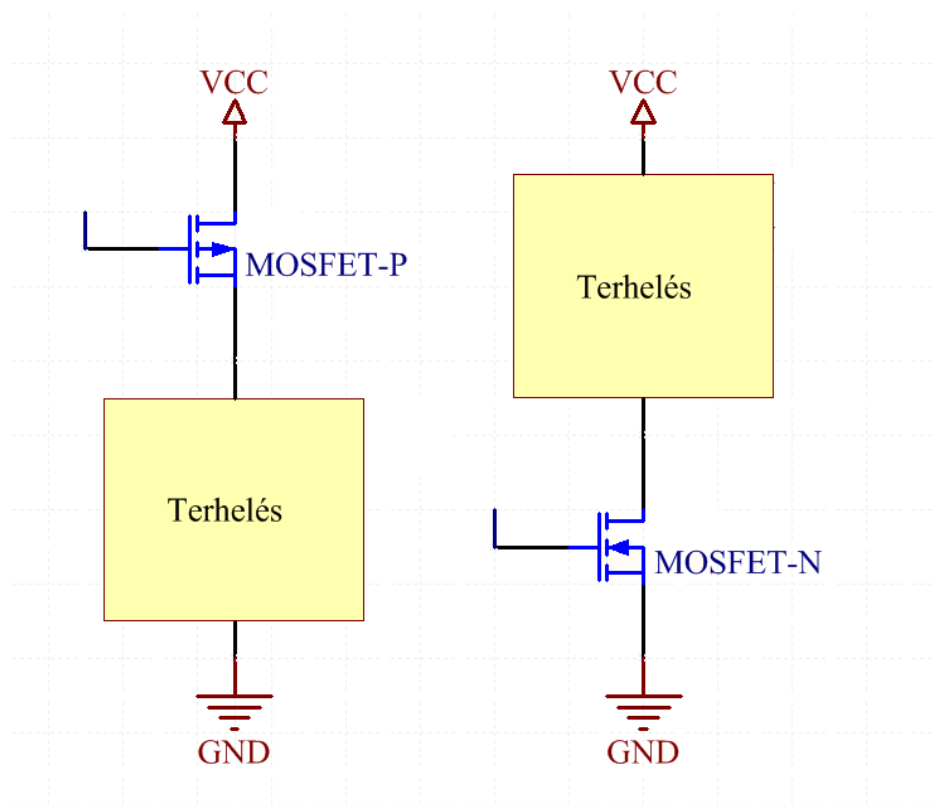


2. ábra: A LED technológia fejlődése

Mindennapi életünk során egyre sűrűbben találkozunk LED-es kijelzőkkel. Leggyakoribbak a teljesen monokróm vagy néhány különböző színű LED-ből felépített kijelzők, de nem ritkák az RGB LED-es programozható megjelenítők sem. Ez a megoldás praktikusán használható reklámozásra, közlekedési információk közlésére, hirdetőtáblákhoz és videók lejátszásához is.

2.2 LED meghajtó áramkörök

Az egyenfeszültségről működtetett eszközök vezérelt kapcsolása kétféle módon oldható meg. Az egyik a high-side switch alkalmazása, amikor a kapcsolni kívánt eszköz fixen kapcsolódik a földponthoz és a tápfeszültség csatlakozóján található az eszköz kapcsolója, ami tipikusan egy „p” csatornás MOSFET (Metal-Oxide Semiconductor Field-Effect Transistor, szigetelőréteges térvezérlésű tranzisztor) vagy „pnp” bipoláris tranzisztor. A másik megoldás a low-side switch, ahol az eszköz folyamatosan rá van kötve a tápfeszültségre és a földcsatlakozás szakítható meg kívülről. A kapcsolgatást ebben az esetben egy „n” csatornás MOSFET vagy „npn” tranzisztor végzi.



3. ábra: Low-side- és high-side switch egyszerűsített vázlata

Az „n” csatornás MOSFET-ek (és az npn tranzisztorok) kapcsolási ideje kisebb, ezáltal a kapcsolási veszteség is csökken. Az ebbe a típusba tartozó eszközök ára alacsonyabb, mint a hasonló paraméterekkel rendelkező „p” csatornásoké, emiatt szélesebb körben alkalmazzák őket. Lehetőség van lehetőség high-side switch építésére

„n” csatornás MOSFET és egy chargepump felhasználásával is. Ezen az okok miatt a LED meghajtó IC-k többsége is low-side switch-ként viselkedik, az általam vizsgáltak is mind ebbe a típusba tartoznak.

Fontos megjegyezni, hogy a kapcsolási módon kívül az eszközök döntő többsége áramgenerátoros üzemmódban működik.

A vizsgált LED driverek legfontosabb paraméterei a projekt során:

- 1.) Tápfeszültség: fontos hogy az IC működjön 3 vagy 5 V-ról.
- 2.) Vezérelhető csatornák száma: a csatornaszám növekedésével nő az IC ára, de csökken a csatorna fajlagos költsége és a kiegészítő áramkörü elemek száma.
- 3.) Csatornák áramterhelhetősége: a megfelelő fényerő biztosításához szükséges, hogy legalább 30 mA áramot tudjon kapcsolni az IC, az összes csatornán egyszerre is, ha szükséges.
- 4.) Tokozás: csak SMD (Surface Mounted Device, felület szerelt alkatrész) alkatrészeket érdemes használni az egyszerű beültethetőség valamint a kis helyigény miatt.
- 5.) Kommunikációs interfész típusa: lényeges paraméter a maximális sebesség, ugyanis ez elengedhetetlen a megfelelő képfrissítés eléréséhez.

Az általam részletesebben vizsgált IC típusok:

- 1.) ST Microelectronics STP08CP05, típusú shift regiszter
- 2.) Nxp Semiconductors PCA9552, LED meghajtó
- 3.) Nxp Semiconductors PCA9956A, LED meghajtó
- 4.) Nxp Semiconductors PCU9656, LED meghajtó
- 5.) Texas Instruments TLC5940
- 6.) Texas Instruments TLC5955

ST Microelectronics STP08CP05:

- Tápfeszültség: 3 – 5.5 V
- Vezérelhető csatornák száma: 8
- Csatornák áramterhelhetősége: 5-100 mA
- Tokozás: TSSOP16
- Kommunikációs interfész: Soros

Az IC shift regiszter, amely bemeneti 4 bemeneti- és 8+1 kimeneti lábbal rendelkezik, a belsejében pedig 2 x 8 bit-es regiszter van. A CLK a külső órajel, az SDI a soros adatbemenet, az LE a latch bemenete, az OE pedig az alacsony aktív kimenet engedélyezés. Az egyik 8 bit-es regiszter latch-elt információt tartalmaz, a másik 8 bit pedig a kommunikációs vonal része. A CLK bemenetre adott felfutó él hatására egy bitet shift-elődik a 8 belső regiszter tartalma, a legutolsó regiszter tartalma megjelenik az adatkimeneten (SDO) a legelső regiszterbe pedig betöltődik az SDI lábon lévő feszültség logikai értéke. Az LE (latch enable) lábon logikai 1 van az órajel felfutó értékére az adatvonalon található 8 regiszter értéke betöltődik a belső adatregiszterbe. A kimenetek engedélyezése az OE bemeneti lábra adott logikai 0 értékkel történik. Lehetőség van fényerő szabályozásra, amennyiben impulzusszélesség modulációval van ellátva a kimenetet engedélyező jel, de ekkora az összes engedélyezett kimenet ezzel a fix kitöltési tényezővel fog működni, nincs lehetőség a szeparált vezérlésre. A CLK-ra adott maximális órajel 30 MHz lehet, és mivel a kommunikációban nincs fejléc képfrekkvenciától függően akár több 100 LED vezérlése is lehetséges egy adatvonalon.

Nxp Semiconductors PCA9552:

- Tápfeszültség: 2.3 – 5.5 V
- Vezérelhető csatornák száma: 16
- Csatornák áramterhelhetősége: 0 - 25 mA
- Tokozás: SOP24, TSSOP24, HVQFN24
- Kommunikációs interfész: I²C (Inter-Integrated Circuit)

Az IC-n 6 bemeneti, 16 kimeneti valamint egy táp és egy földponti láb van kivezetve. Az I²C-s kommunikáció használata miatt az IC-nek rendelkeznie kell egy slave address-el, aminek segítségével be tudja azonosítani a számára releváns adatsomagokat. Ehhez 3 db bemeneti láb áll rendelkezésre (A0, A1, A2). Ezeken a lábakon nincs belső fel- vagy lehúzó ellenállás így a logikai érték beállítását kívülről tudjuk megvalósítani. A 3 láb segítségével összesen 8 cím osztható ki, így maximálisan ennyi eszköz helyezhető egy I²C buszon. Az IC különböző funkcióit egy 8 bit-es control regiszter-en keresztül tudjuk elérni. Az IC belsejében 10 darab belső 8 bit-es regiszter van. 2 db Input regiszter, amely a 16 csatorna programozható engedélyezéséért felel, 4

regiszter van az impulzusszélesség moduláció beállításához, és 4 regiszter, ami hozzárendeli a megfelelő kitöltési tényezőt az egyes csatornákhöz.

A busztranzakció során az első bájt kiválasztja a megfelelő című slave egységet, a második bájt a command regiszterben beállítja a bejövő adat célregiszterét, majd a harmadik bájt, ami a tényleges információt tartalmazza, beíródik a megfelelő belső regiszterbe. Ezután a következő ciklustól kezdve a frissített paraméterek szerint működnek a kimeneti csatornák.

Mivel a kimenetek nem rendelkeznek áramszabályozással, így a rájuk kötött LED áramkorlátját egy külső ellenállás beépítésével lehet beállítani.

Nxp Semiconductors PCA9956A:

- Tápfeszültség: 3 – 5.5 V
- Vezérelhető csatornák száma: 24
- Csatornák áramterhelhetősége: 0 - 57 mA
- Tokozás: HTSSOP38
- Kommunikációs interfész: I²C

Ez az IC is NXP-s gyártmányú, de sok paraméterben eltér az előző típustól. Legfontosabb különbség, hogy egy külső ellenállást az R-EXT bemenetre kötve beállíthatjuk az összes csatorna áramkorlátját így nem kell minden LED elé egy hozzá dedikált ellenállás. Másik lényeges különbség, hogy bár ennek a típusnak is 3 bemeneti láb van (A0, A1, A2) a kommunikációs cím kiválasztásához, képes megkülönböztetni, hogy lebeg-e, földre vagy tápra van-e kötve vagy pedig fel-/lehúzó ellenállás található-e rajta, így egy buszon akár 125 különböző című eszköz lehet egyszerre. A buszvezérlő tud kiadni egy „allcall” nevű parancsot, ezzel egyszerre üzenetet küldve az összes buszon lévő IC-nek. Akár minden egyes csatorna eltérő kitöltési tényezőre programozható és egyedi áramkorlát is beállítható rájuk (maximálisan a külső ellenállás által korlátozott értékre). A programozás során lehetőség van egy-egy paranccsal az összes kimenet kitöltési tényezőjét vagy áramkorlátját egyszerre állítani. A kimenetek engedélyezése egy bemeneti lábra (OE) adott „0” logikai értékkel lehetséges. Van egy alacsony aktív RESET bemeneti láb is, amely kitöröl minden addigi beállítást. A RESET és az OE lábat érdemes egy felhúzó ellenálláson keresztül tápfeszültségre kötni a buszvezérlőhöz való bekötéssel párhuzamosan.

Nxp Semiconductors PCU9656A:

- Tápfeszültség: 2.3 – 5.5 V
- Vezérelhető csatornák száma: 24
- Csatornák áramterhelhetősége: 0 - 100 mA
- Tokozás: LQFP48
- Kommunikációs interfész: I²C

A chip 5 cím bemeneti lábbal rendelkezik, viszont mindegyiket vagy tápfeszültségre, vagy földre kell huzalozni. Így egy buszon akár 32 vezérlő is lehetne, de van 6 db más célokra lefoglalt cím. A működése nagyon hasonlít az PCA9552-es típusú IC-hez. Aktív low kimenet engedélyező lába van és programozható a csatornák kitöltési tényezője (8 biten, 256 lépésben). Ami eltér, hogy ezen az IC-n nincs szoftveresen vagy külső ellenállással egy bemeneti lábon beállított áramkorlát, hanem minden egyes csatornára külön ellenállást kell a kapcsolt LED-ekkel sorba kötni. Vannak viszont előre beállított csatornaengedélyezési sémák, ami megkönnyíti az egyszerűbb ábrák és animációk kirajzolását.

Texas Instruments TLC5940:

- Tápfeszültség: 3 – 5.5 V
- Vezérelhető csatornák száma: 16
- Csatornák áramterhelhetősége: 0 – 60 mA
- Tokozás: HTSSOP28, PQFN-32
- Kommunikációs interfész: Serial Data Interface

Az IC 8 bemeneti és 18 kimeneti lábbal rendelkezik, amelyből 16-ot adnak a vezérelt csatornák. A bemeneti lába között van a soros adatbemenet (SDI), a hozzá tartozó órajel (SCL), a kimenetek engedélyezéséért felelős alacsony aktív láb (BLANK). Az eszköz kitöltési tényező 12 bites (4096 lépés) fokozatban állítható ehhez szükséges a GSCLK bementre adott megfelelő órajel, valamint lehetőség van pixeles szoftveres áramkorlátozásra is (Dot Correction) az IREF lábra kötött külső ellenállás mellett. Ez azért szükséges, mert nem biztos, hogy sikerül minden LED-et azonos karakterisztikájúra legyártani és emiatt különböző lehet a fényerejük azonos külső paraméterek mellett. Az üzemmód beállítása a DCPRG és VPRG láb megfelelő logikai szintre való húzásával lehetséges. Az esetleges hibákat az XERR kimeneti láb logikai „0” szintje jelzi. A megjelenített tartalom változtatása DCPRG és VPRG = 0 esetén

lehetséges. Először le kell kapcsolni a LED-ek engedélyezését (BLANK = 1), majd az XLAT magasba húzása esetén a shiftregiszterek tartalma bemásolódik a belső regiszterbe, ami a kitöltést vezérli. A vezérlés lényege, hogy a 12 bit-en megadott információ értékét elmentjük az XLAT felhúzásakor, majd a GSCLK lábra adott órajelet számolja az IC és minden LED kimenetre ellenőrzi, hogy elérte-e a PWM órajel számlálója a beállított értéket. Amint eléri a beállított értéket az adott kimenet lekapcsolásra kerül. A GSCLK-hoz tartozó számláló újraindítása és a LED-ek felkapcsolása az XLAT bemenet felfutó élére történik.

Texas Instruments TLC5955:

- Tápfeszültség: 3 – 5.5 V
- Vezérelhető csatornák száma: 48
- Csatornák áramterhelhetősége: 0 – 32 mA
- Tokozás: HTSSOP56, QFN-56
- Kommunikációs interfész: Serial Data Interface

Működése nagyban hasonlít az előzőleg említett típuséhoz, viszont van néhány eltérő paraméter. Legszembetűnőbb különbség a nagy csatornaszám, ezért RGB LED-hez is ajánlják ezt a típust, így a 48 csatornával egyszerre 16 db ilyen világítóeszköz vezérelhető. A színmélységet 16 biten lehet állítani 65536 lépésben, és be van építve egy 128 lépéses globális fényerővezérlés funkció is. A vezérlésével kapcsolatban egyetlen lényeges különbség, hogy nincs BLANK, VPRG és DCPRG lába, ami előny, hiszen kevesebb huzalozás szükséges az amúgy is sűrű NYÁK-terven. Az IC alján található egy hűtő felület, ami össze van kötve belül a földkivezetéssel. A működés részletesebb bemutatása a rendszertervben olvasható.

2.3 Hasonló célú LED mátrix kijelzők

Internetes kutatás során az ACE Communications Kereskedelmi és Szolgáltató Kft-nél találtam hasonló funkciójú és kijelzőket. Ők szállították a Corvin mozi előtti téren található kijelzőt. Több fajta kijelző forgalmazásával foglalkoznak. Kínálatukban szerepel fix telepítésű-, mobil- és átlátszó LED-fal. Standard méretű kijelzők között a legnagyobb kapható méret az 1280x1280 mm. Ezen kívül foglalkoznak egyedi igények kielégítésével is. A fix telepítésű LED-falakat jellemzően boltok vagy üzletek kirakatában, irodaházakban, repülőtereken, kiállításokon, bevásárlóközpontokban

használják. A LED-fal kivitele lehet kültéri vagy beltéri. A kijelző fényereje egy külső fényszenzor segítségével a külső fényhez állítható. Használhatóságuk érdekében nagy fényerejű LED-eket kell használni és tükröződésmentes borítást kell alkalmazni.

A LEDFUL Ltd 2008-ban alakult és azóta fő profilja a LED képernyők gyártása. A legnagyobb méret, amivel szériagyártásban foglalkoznak az 1280 x 960 mm nagyságú kijelző. Négyzetméterenként 2500 pixel-lel dolgoznak 20 mm-es pixel távolsággal, a kijelző maximális fogyasztása 400 W/m². Kültéri kijelzőiket IP65-os tokozással látják el.

A PixelFLEX Ltd nagyfelbontású RGB modulok értékesítésével foglalkozik. A legnagyobb moduljuk 768 x 768 mm, de ezekből többet egymáshoz illesztve nagyméretű kijelzőket lehet összeállítani. Ezek a modulok 4.8 mm-es pixeltávolsággal rendelkeznek, így 1 m²-en közel 40000 darab Led található. A kültéri egység tokozása IP43-as védettséggel rendelkezik.

3 Specifikáció

3.1 A kijelzővel szemben támasztott követelmények

A paraméterek többségét én határoztam meg, mert a cég nem sok szigorú kitélt írt elő. Az általuk szabott feltételek:

Maximális külső méret: 1.5 x 2.5 m (magasság x szélesség)

Ekkora hely lett kihagyva az építészeti terven a kijelzőnek, ezért lényeges ez a feltétel.

Egyoldalas NYÁK-terv:

Költséghatékonysági okokból házon belül szeretnék legyártani a NYÁK-okat, mivel ekkora felületnél a gyártási költség is nagyon magas lenne. Kezdetben A4-es méretű (297 mm x 210 mm) NYÁK-panelek használatát tervezték, de később ez A3-as (420 mm x 297 mm) méretűre módosult.

Felületszerelt (SMD) alkatrészek használata:

A cégen belül gyártott NYÁK-okon a furatok kialakítása bonyodalmas lenne nagy számosság esetén, ezért célszerű a felületszerelt alkatrészek alkalmazása.

Elérhető legjobb felbontás:

A LED-ek számát 4-7000 körülre tervezték. Ez már élvezhető képminőséget biztosít. A felbontás növelésével rohamosan emelkedik az alkatrész költség, így ennél több LED használata nem fért volna bele a költségkeretbe.

Pixelenkénti fényerő szabályozhatóság:

Az animációk és grafikus adatok megjelenítéséhez szükség van rá, hogy pixeleket eltérően lehessen ábrázolni. Mivel a kijelző monokróm ezt csak a fényerő szabályzásával lehet elérni.

Lehetőleg RS485-ös adatfogadás:

Ez lesz a rendszer kapcsolata a külvilággal, a csatlakozó másik oldalának kiépítése már nem ennek a projektnek a része. Később ezt a csatlakozót Ethernet-re kellett cserélni, mivel az RS485 nem volt képes az elvárt sávszélességre.

3.2 A kijelző paraméterei

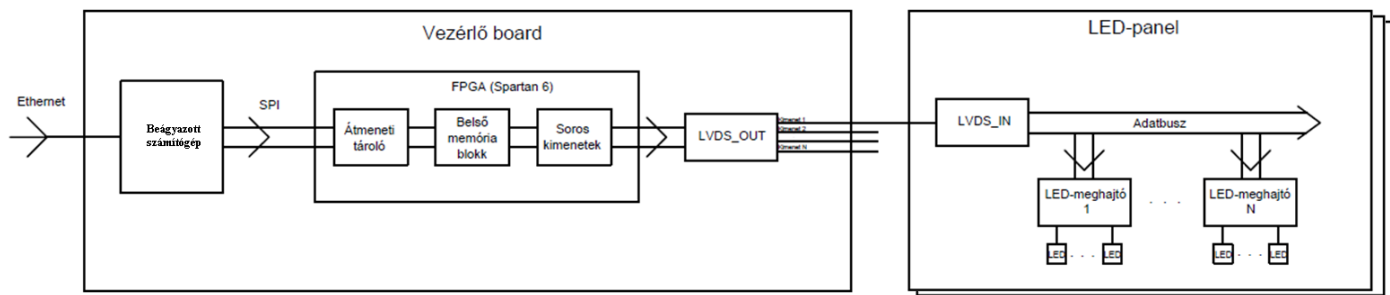
- A kijelző tervezett mérete: 1.49 x 1.98 m (magasság x szélesség)
- Felbontása: 60 x 80 pixel, 4800 LED
- Felhasznált LED-meghajtó típusa: TLC5955
- LED-meghajtók száma: 100
- Teljes színmélység: 16 bit
- Egy LED maximális áramfelvétele: 32 mA
- Maximális fogyasztás: 1 kW
- LED-panelek száma: 5 x 5 db
- LED panelek mérete: 297 x 396 mm
- LED-ek száma egy panelen: 192
- LED-meghajtó IC-k száma egy panelen: 4 db
- LED-meghajtók maximális disszipációja: ~2.5 W
- LED-ek színe: zöld
- Felhasznált mikrovezérlők: Raspberry Pi, Spartan-6 (FPGA)
- Bemenet a külvilág felől: Ethernet 10/100 Mbit/s

4 Rendszerterv

4.1 Hardver rendszerterv

Maga a kijelző nem közvetlenül a felhasználó előtt lesz, hanem jelentős távolságban egy folyó túloldalán. Ahhoz, hogy ilyen távolságból is látható képet kapjunk nagyméretű kijelző szükséges. Jelen projekt esetében a maximális rendelkezésre álló méret a 1.5 x 2.5 m, mert az építészeti terven ekkora hely lett kihagyva az erőmű oldalán. Ekkora méretű NYÁK legyártása nagyon nehezen lenne kivitelezhető és a költsége is óriási lenne ezért kézenfekvő volt a paneles felépítés alkalmazása. Ezen a felületen kell elhelyezni 4-7000 LED-et, ez a mennyiség nagyban függ a kialakítástól.

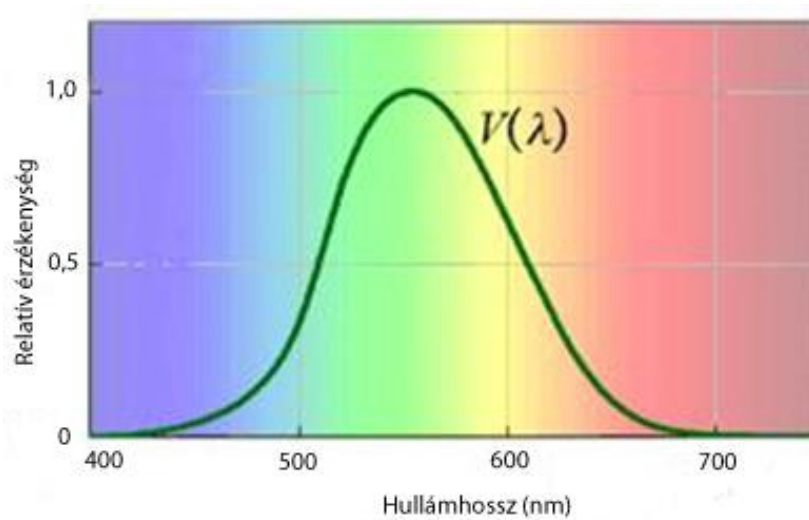
A hardver két fő részre bontható. Az első csoportot a LED panelek alkotják. Ez tartalmazza a megjelenítéshez használt LED-eket, valamint a hozzájuk tartozó LED-meghajtó IC-eket. A LED-fal azért áll több egyforma panelből, mivel így sokkal egyszerűbb legyártani és karbantartani őket. A második részegység a vezérlésért felelős board. A nevéből is következik, hogy ő fogja tartalmazni a beágyazott mikrovezérlőt. Ez a NYÁK fizikailag nem lesz beépítve a LED-falba, hanem a közelében lesz elhelyezve. Ez a távolság differenciális jelátvitel használatát indokolja a LED-panelek és a vezérlő egység között. Felépítését tekintve ez összetettebb és bonyolultabb egység, viszont ebből csak egy darab lesz az egész rendszerben.



4. ábra: A hardveres rész egyszerűsített blokkvázlata

Felhasznált LED-ek:

A kijelző több ezer azonos színű LED-ből fog állni, így ennél a résznél rendkívül fontos volt az alacsony egység ár. Erre a célra az olcsó kínai LED-ek is megfeleltek, mivel még egy esetleges meghibásodás esetén gyorsan és egyszerűen lehet cserélni őket. A felhasznált LED-ek színének kiválasztásánál figyelembe vettem az emberi szem sajátosságát, miszerint a látható spektrum közepén (550 nm környéke) sokkal érzékenyebb, mint a két szélén.



5. ábra: Láthatósági függvény, az emberi szem nappali érzékenysége ^[3]

Az 5. ábra alapján zöld színű LED-eket érdemes használni, mert így ugyanakkora energiafelhasználással fényesebb képet fogunk látni, mint zöldessárga, piros vagy akár kék LED-ek esetén.



6. ábra a projekt során használt LED-ek

Felhasznált LED-meghajtó típusának kiválasztása:

Az ST Microelectronics gyártmányú shiftregiszter hamar elvetésre került, mivel nem tudja teljesíteni a fényerő szabályozhatósági követelményt minden pixelre egymástól függetlenül. Ezek után az NXP Semiconductor által kínált lehetőségeket próbáltam ki. A PCA9552 típus legnagyobb hátránya, hogy egy buszra maximum 8 db fűzhető fel belőle, ezért egy LED-panelen több I²C buszra lett volna szükség. Ez jóval bonyolultabb tervet és méretesebb kiegészítő áramkört eredményez. A következő lehetőségként a PCA9956-os típussal kísérleteztem. Egy próbapanel megtervezése után a mintadarab beszerzésénél problémák merültek fel. A teljes LED-falhoz nagyságrendileg 400 IC szükséges. Ez túl nagy beruházás ahhoz, hogy az IC kipróbálása nélkül bele lehessen vágni. Amikor megpróbáltam a gyártó által megjelölt beszállítóktól mintadarabot beszerezni kiderült, hogy a PCA9956-os típus már kifutó széria, csak a készlet erejéig lesz kapható. Ezután a panelt módosítottam a hozzá leginkább hasonló PCU9656-ös IC felhasználásához. Ennek a chipnek az a hátránya, hogy csak teljes tekercset árultak belőle megbontás nélkül, amely 2500 db-ból áll. Ez a mennyiség a szükséges négyszerese és ekkora többlet költség meghaladta a projekt költségvetését.

Ezután sikerült beszerezni a Texas Instruments-től 2 db TLC5940-es típusú IC-t, így ehhez építettem egy próbakapcsolást. A végső választás azonban az újonnan megjelent TLC5955-ös típus lett, mivel a fajlagos csatorna költsége alacsonyabb, valamint biztosan kapható lesz még a projekt lezárása után is.

Az általam tervezett LED-panel 4 ilyen IC-t tartalmaz, tehát egy képkocka adattartalma $4 \times 769 = 3076$ bit. Ha mozgóképet szeretnénk látni, akkor kell a másodpercenkénti 50 Hz-es képfrissítés, ami $50 \times 3076 = 153800$ bit. 25 panel esetén minimum 3.845 Mbit/s sávszélesség szükséges. Ha a teljes 16 bites színmélységet szeretnénk használni, akkor a PWM órajelnek $2^{16} \times 50 = 3,277$ MHz-nek kell lennie. Az általam választott differenciális jellevő 100 megatranszfer/s-ra képes a 4 csatornáján összesen, így ezeket gond nélkül képes átvinni.

4.1.1 LED Panel

Ezen a panelen helyezkednek el a LED-ek és LED-meghajtó IC-k. A bejövő differenciális jelet először vissza kell alakítani logikai jellé, amelyet a meghajtók már képesek feldolgozni. Ez a jel egy soros adatbuszon keresztül jut el, kommunikációs

protokolltól függően, a LED-meghajtókhoz, melyek ennek megfelelő kitöltési tényezővel működtetik a hozzájuk tartozó LED-eket. Az adatbusz a következő vonalakból áll:

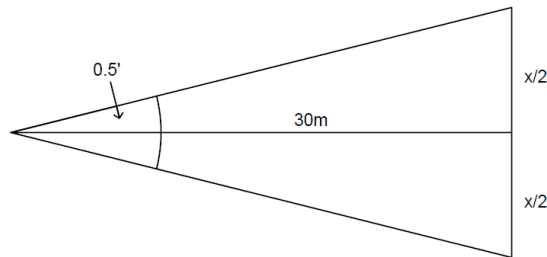
- Soros adatvonal
- Adatvonal órajele
- PWM órajel
- Latch engedélyező jel

Ezek a vonalak minden IC-be be vannak kötve.

A LED-ek saját független tápforrással fognak rendelkezni, a LED-meghajtók áramellátásáról a vezérlőpanel fog gondoskodni, mivel ezek fogyasztása jóval kisebb, mint a LED-eké.

LED-ek távolsága:

Egy átlagos emberi szem felbontóképessége 1 ívperc (1' a fok 1/60-ad része). Ezt felhasználva 30 m-es távolságban a legközelebbi megkülönböztethető pontok távolsága az alábbi jelölést felhasználva:



7. ábra: Az emberi szem felbontóképességének határa

$$x = 2 \cdot 30 \cdot \operatorname{tg} \frac{1}{60 \cdot 2} = 8.72 \cdot 10^{-3}$$

Tehát 9 mm-nél közelebb felesleges a LED-eket elhelyezni, mivel nem tudjuk egyedi pixelként felismerni őket. Ennél egy kicsit szigorúbb korlátot jelent a LED-meghajtó IC fizikai mérete. Mivel egyoldalas NYÁK-on felületszerelt alkatrészeket kell használni két szomszédos LED között el kell férnie a meghajtónak. Az általam kiválasztott TLC5955-es IC mérete 14 mm x 8 mm, a LED-ek mérete pedig

5 mm x 5 mm. Ahhoz, hogy az A3-as panel rövidebb oldalát teljes mértékben kihasználja a LED rács a 24.75 mm-es távolságot választottam. Ekkor ugyan teljes hosszában nem fér el az IC két LED között, viszont a csatornáinak kimenetei az IC középső lábain helyezkednek el, így a vezérlőjelek elvezetése megoldható. Az adatvonalak pedig rájuk merőleges irányban elférnek.

Egy LED panel topológiája:

Ahhoz, hogy a LED panel egyszerűen és átláthatóan legyen huzalozva érdemes a LED-meghajtó áramkörök környezetét egyformára kialakítani. Ez azért is előnyös, mert az általam használt CAD program támogatja az azonos felépítésű szegmensek elrendezésének lemásolását. Ez azt takarja az esetemben, hogy elég volt megtervezni és megrajzolni egy IC és a hozzá tartozó LED-ek huzalozását és ezt a topológiát néhány perc alatt át tudtam másolni a panel többi részére.

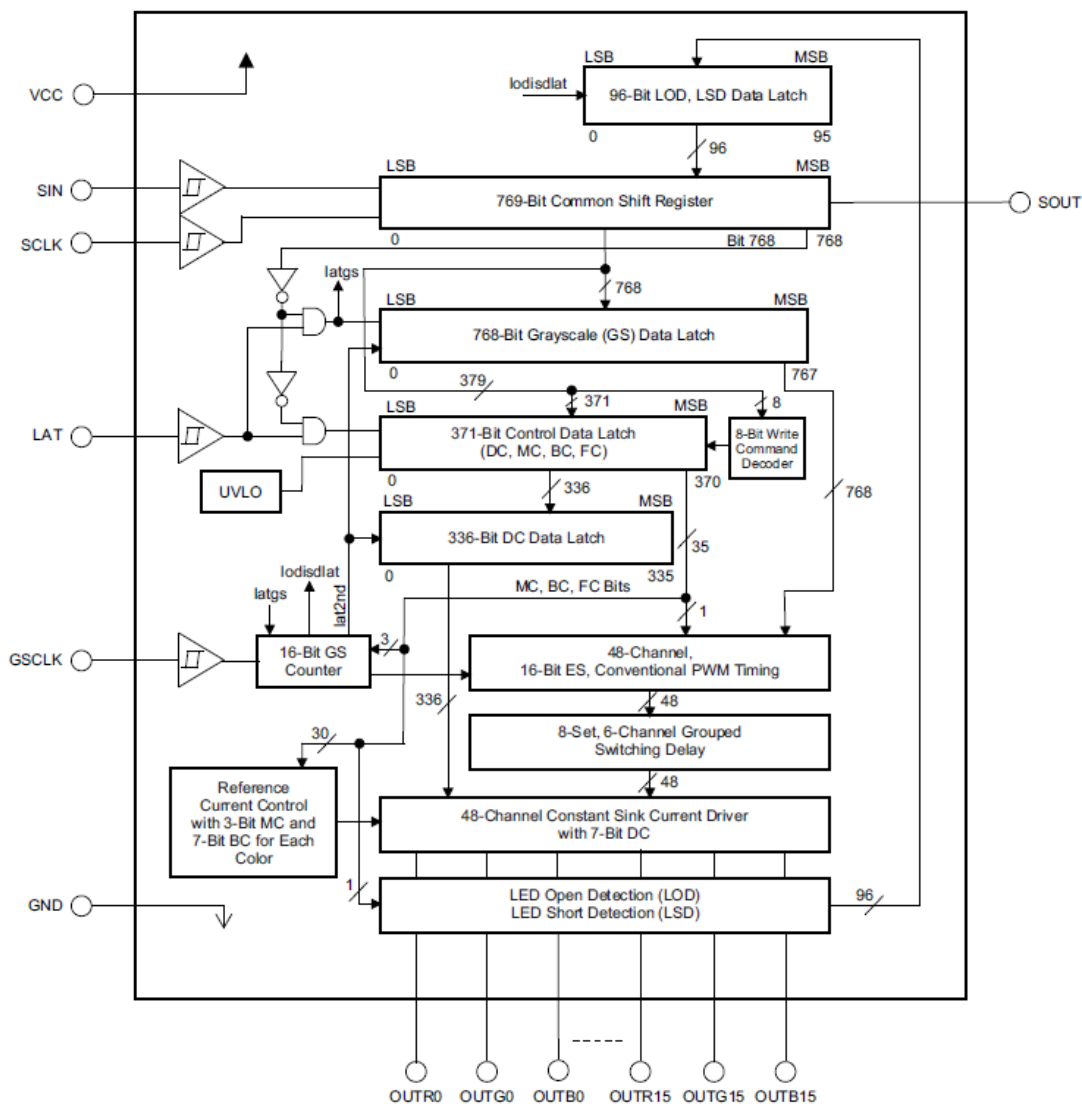
Az általam is használt TLC5955-ös LED-vezérlő 48 csatornával rendelkezik, így ha azonos elrendezésben szeretném használni az egy panelen elhelyezett IC-ket érdemes a LED-eket 6 x 8-as mátrixban elhelyezni. Lehetne használni 4 x 12-es vagy 2 x 24-es elrendezést is, viszont ekkor az adatvonalak huzalozása nehezen kivitelezhető egyoldalas NYÁK-on. Nem lenne megoldhatatlan, de ha a NYÁK alsó oldalára is rajzolat kerül, akkor ahhoz átvezető furatok kellene, viszont ónozott falú furatokat nem tudok létrehozni, tehát marad az a módszer, hogy alkatrészlábat vagy vezetékkel kell átfűzni a lyukon.

A 6 x 8-as elrendezésnél panel közepén mennek a kommunikációs vonalak, a két szélén pedig a LED-ek tápellátása. Ekkora 4 db LED-meghajtót elhelyezve 2 sorban egy panelen $12 \times 16 = 192$ LED-et lehet elhelyezni. Ezzel a paraméterrel a LED-ek egy 24.75 mm-es négyzetrácsban lesznek elhelyezve. ($420 / 16 = 26.25$ és $297 / 12 = 24.75$, a kettő érték közül pedig a kisebb valósítható meg). Ez eredményezte a LED-panel $24.75 \times 16 = 396$ mm-es szélességét.

A panelek számának és elrendezésének megtervezésénél több variációt is számításba vettem. A paneleket el lehet helyezni állítva és fektetve is, így teljesen más méretű LED mátrix kijelzőt lehet létrehozni. Néhány variációt bemutatva a céges konzulensemnek úgy döntöttünk, hogy legyen 5 x 5 db fektetett panel, ami 60 x 80 pixeles kijelző felbontást eredményezett. Ennek az elrendezésnek a mérete egy 1.49 m magas és 1.98 m széles kijelző, ami beilleszthető az erőmű oldalán kijelölt falrészbe.

A TLC5955-ös LED-meghajtó IC vezérlése:

Az TLC5955-ös funkcionális blokkdiagramja az alábbi ábrán látható:



8. ábra: A TLC5955-ös IC funkcionális blokkdiagramja ^[2]

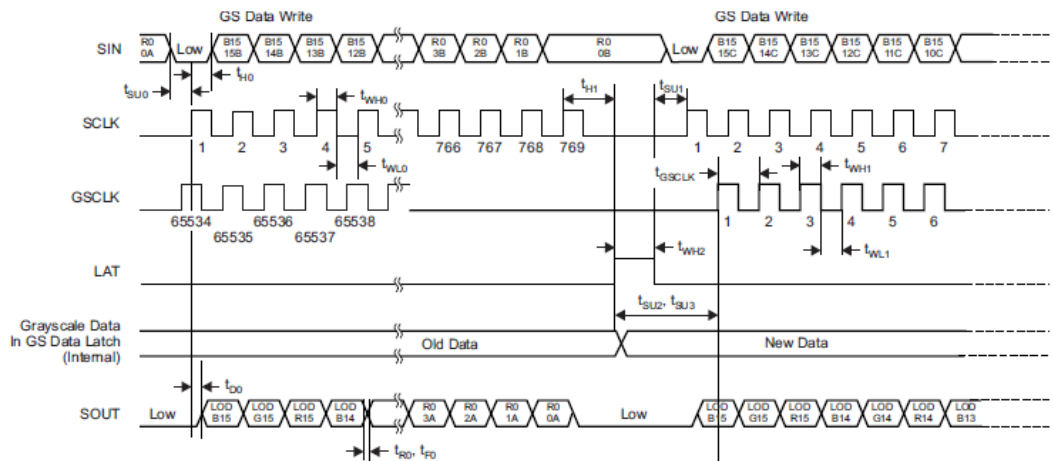
Az eszközt főleg RGB LED-ek vezérléséhez tervezték, ezért minden kimeneti csatornája hozzá van rendelve egy színcsatornához (kék, zöld, piros), bár én ezt a funkciót nem használom majd, mert minden csatornára egyformán zöld LED lesz rákötve. Az áramkörben található egy 769 általános shiftregiszter, ami közvetlen része az adatvonalnak. Az IC-k kaszkádosítása úgy lehetséges, ha láncba (Daisy chain) vannak felfűzve, vagyis az első kimenete a második bemenete és így tovább. A különböző funkciók elérése viszont nem külső programozó jelek bekötésével van megoldva, hanem szoftveresen, a 769. bit használatával. Ez az úgynevezett latch select

bit, ha ez 0, akkor az általános célú shiftregiszter tartalma betöltődik a kitöltési tényezőt tároló regiszterbe (Grayscale Data Latch), ha pedig 1, akkor tartalomtól függően valamelyik másik belső regiszter programozása válik elérhetővé. Ezek a regiszterek kisebb méretűek és az alábbi funkciókért felelősek:

- Control Data Latch: vezérlési információkat tárol. Bekapcsoláskor tartalma véletlenszerű ezért mindenképpen fel kell programozni, mielőtt a kimenetek bekapcsolásra kerülnek.
- Dot Correction (DC) Data Latch: A kimenetek áramát lehet csatornánként korrigálni 7 bit mélységben. Ennek az a célja, hogy a LED-ek nem egyformák és így lehetőség van a fényerejüket egymáshoz hangolni. Bekapcsolásnál ennek a tartalma nem definiált.
- Maximum Current (MC) Data Latch: 3 biten tárolja a 3 színcsatorna maximális áramkorlátját. 24 mA-nél csak akkor lehet nagyobb a korlát, ha a LED-ek tápfeszültsége nagyobb, mint 3.6 V. A regiszter módosításához a tartalmát kétszer kell felülírni a módosított értékkel.
- Global Brightness Control (BC) Data Latch: A három színcsatorna fényerejét lehet állítani egymáshoz képest. Ennek a regiszternek is véletlenszerű a tartalma bekapcsoláskor.
- Function Control (FC) Data Latch: olyan funkciók állíthatók be vele, mint az automatikus kijelzés ismétlés, időzítés újraindításának lehetősége, automatikus kitöltési tényező frissítés vagy rövidzár érzékelés.
- Status Information Data (SID): az áramkör képes érzékelni, ha egy csatornáján szakadás van és ez az információ lekérdezhető tőle.

Az általam legsűrűbben használt funkció egyértelműen a kitöltési tényező állítása lesz, hiszen a megfelelő működési paramétereket elég egyszer induláskor megadni, viszont a LED-ek fényerőssége képkockánként változhat. Minden LED-panel közös órajeleket fog használni, amit az FPGA (Field Programmable Gate Array, programozható kapumátrix) generál, így szinkronban fog működni az egész rendszer. A fényerő vezérlésének idődiagramja az alábbi ábrán láthat:

Timing Diagrams (continued)



9. ábra: A TLC5955-ös időzítési diagramja

Az áramkör 4 bemenettel (SIN, SCLK, GSCLK, LAT) és egy kimenettel rendelkezik (SOUT). A soros adatbemenet (SIN) és a hozzátartozó órajel (SCLK) szinkronizált működése elengedhetetlen. SCLK felfutó élére egyet shift-elődik az általános regiszter tartalma. Az első bit helyére bekerül az SIN-en található érték, az utolsó bit értéke pedig megjelenik SOUT kimeneten. Ezekről függetlenül működik a GSCLK, aminek felfutó éleit egy 16 bites számláló számolja. A LAT bemenet felfutó élére az általános regiszter tartalma bemásolódik a belső tárolóba, a számláló értéke nullázódik és kezdődik a következő ciklus.

4.1.2 Vezérlő Board

Ezen a NYÁK-on található a központi vezérlő egység, egy FPGA továbbá a differenciális jelátvitelhez szükséges LVDS-meghajtók (Low-Voltage Differential Signaling, alacsony feszültségű differenciális jelátvitel) adó oldali része.

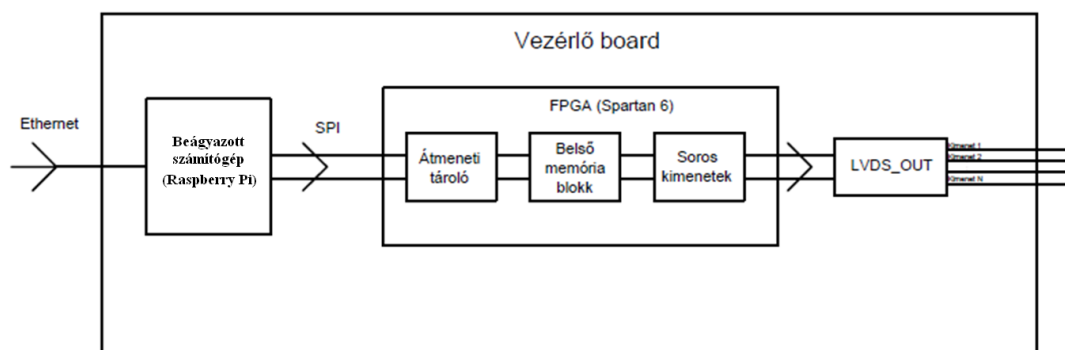
A vezérlő board-on helyezkedik majd el egy beágyazott számítógép. Erre azért van szükség, hogy egy könnyen és rugalmasan alakítható eszköz tartsa a kapcsolatot a külvilág és a kijelző között. A cégnél erre a célra a Cosmic Board nevű Phytex gyártmányú beágyazott számítógépet ajánlották és kipróbálásra biztosítottak is egy példányt számomra. Rövid mérlegelés után viszont arra jutottam, hogy egy Raspberry Pi-t fogok használni erre a célra, mivel ez egy népszerű és széles körben támogatott eszköz az olcsó beágyazott számítógépek piacán. A Raspberry Pi-on Linux operációs rendszer futtatható. Erre az eszközre saját disztribúcióval rendelkezik a gyártó, valamint számos előre megírt alkalmazás és periféria kezelés is található hozzá. Legnagyobb

előnye az volt a Cosmic Board-al szemben, hogy rendelkezett egy egyszerű telepítő programmal valamint grafikus felülettel, ami nagy segítséget nyújtott a szoftverfejlesztés során.

Kezdeti koncepció:

Az első iteráció során még NXP Semiconductor gyártmányú IC-ket terveztem használni, amivel I²C-n keresztül lehet kommunikálni. Az I²C-nél viszont minden slave egységnek külön címe van, így az egy buszra felfűzhető LED-meghajtók számát korlátozza a maximálisan kiosztható címek száma. A buszokat a PCA9646-os típusú áramkör segítségével lehet négy különböző busz felé szétosztani. Ezeket a pufferező és kapcsoló áramköröket kaszkádosítani kellene és ezzel már 16 I²C busz állna rendelkezésre. Mivel a LED-panelek mérete még A4-es volt, a központi vezérlő pedig csak három darab I²C port-tal rendelkezik még egy SPI (Serial Peripheral Interface, soros periféria interfész) buszt is fel kellett volna használni és I²C busszá konvertálni egy SC18IS600 típusú IC közbeiktatásával. Ez az elrendezés rendkívül bonyolult szoftveres megoldást igényelt volna, valamint a közbenső áramkörök folyamatos átkapcsolása pedig többlet adatforgalmat.

Áttervezett struktúra:



10. ábra: A vezérlő board felépítésének váza

A fentebb említett okok miatt volt szükség egy FPGA közbeiktatására, mivel az ilyen típusú beágyazott vezérlők általában 2-4 db soros adatbusz (pl.: I²C, SPI) kezelését támogatják és a feladat során minden panel külön buszra lesz felfűzve. Ez panelmérettől függően 20-40 db csatornát jelent. Az FPGA használatával viszont könnyen elő lehet állítani a kívánt mennyiségű kimenetet, amelyek egymástól függetlenül képesek működni. Ezzel a megoldással szükség lesz az FPGA belső hardvertervének elkészítésére és egy felfelprogramozó rész kialakítására.

Az FPGA és a beágyazott számítógép között párhuzamos vagy soros adatkapcsolat lehet. Soros adatátvitel használata esetén másodpercenként 50 kiküldött képpel számolva $(769 \times 4) \times 25 \times 50 = 3.845$ Mbit/s lenne az adatforgalom. Ezt a sebességet az SPI elbírja, így akár ki is lehetne hagyni az FPGA-t és sorban felfűzni a paneleket egymás után. Ehhez a Raspberry Pi-nak kellene előállítani a többi vezérlőjelet is (SCLK, GSCLK, LAT). Ezeket GPIO lábak és timer-ek alkalmazásával meg lehet oldani. Ennek a megoldásnak az lenne a hátránya, hogy egy LED-vezérlő meghibásodása esetén az összes többi kiesne vele együtt. Másik ellenérv, hogy a paneleket fizikai csatlakoztatásánál is figyelni kellene a helyes felfűzési sorrendet az adatbusznál, valamint bonyolultabb lenne a NYÁK-terv is, mert figyelni kellene a megfelelő továbbvezetés kialakítására. Soros felfűzés esetén ekkora távolságnál megnövekedne az elküldött jelek útja csak a panelek között közel 10 m-re (a 25 panelen keresztül haladó adatbusz esetén), amihez hozzá jön még a vezérlő board-tól való távolság is. Ekkora távolságon már érdemes néhány helyen erősítést beépíteni a hibalehetőség csökkentése végett.

Az 3.85 Mbit/s-es adatforgalomhoz célszerű soros adatátvitel használni. Az FPGA egy SPI blokk használatával fogadja a bejövő adatokat a mikrokontroller felől. Mivel viszont az nem biztosított, hogy bejövő adat fogadásakor az FPGA már nem dolgozik az előző képinformációval, azaz nincs éppen küldési folyamat, kettős pufferelés szükséges. A fényinformáció először egy átmeneti tárolóba kerül. Innen csak akkor töltődik át abba a tárolóba ahonnan a kimenő adatok küldése történik, ha véget ért az előző adás ciklusa. Így elkerülhető, hogy a képernyő egyik részén a régebbi, míg másik részén az újabb kép egy részlete jelenjen meg.

Az FPGA kiválóan alkalmas, hogy több kimenetet párhuzamosan kezeljen. Képes időeltolódás nélkül kezelni a kimeneti interfészeket, valamint mivel tetszőleges konfiguráció programozható bele, egy chip-el megoldható az összes adatbusz létrehozása és vezérlése. Az adatbusz SPI lesz, a hozzá szükséges órajelet pedig egy kimeneti láb használatával el tudja látni az FPGA az összes szükséges adatbusz számára. Ezzel biztosítva a LED-meghajtók szinkronizált működését. Az FPGA hardverének leírása Verilog nyelven fog történni.

Az FPGA és a külvilág közötti kommunikációt egy beágyazott számítógép fogja végezni. Az információ bemenete az Ethernet port lesz, így egyszerűen lehet fogadni csomagokat TCP/IP protokoll használatával.

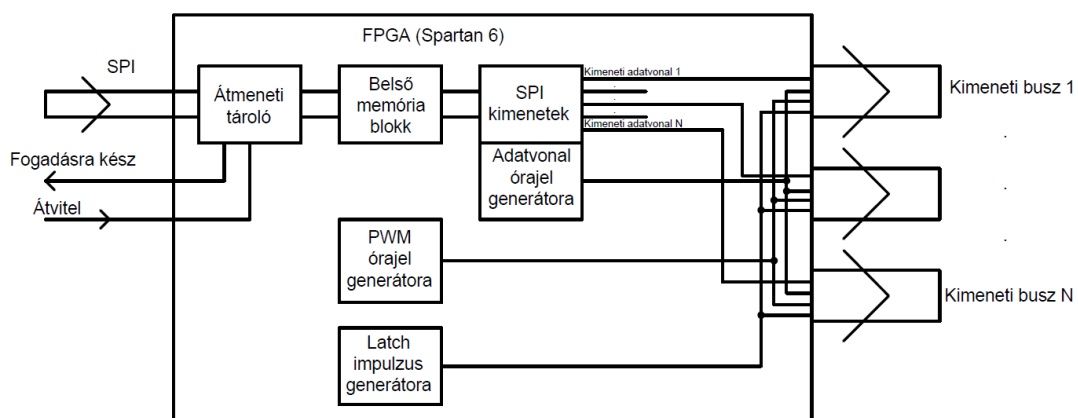
Az FPGA JTAG-en keresztül (Joint Test Action Group) programozható fel. Az FPGA-hoz illesztve lett egy SPI-on keresztül elérhető Flash memória, amibe a program letölthető, hogy bekapcsoláskor innen induljon a működés.

4.1.3 Az FPGA hardverterve

A vezérlő automatikus fog indulni a rendszer bekapcsolásával. Legfontosabb feladata a kívülről érkező adatok feldolgozása és továbbítása az FPGA felé. Induláskor inicializálni kell a bemeneteket és a kimeneteket. Az adatküldés SPI használatával történik majd, a vezérlő pedig az FPGA-t egy külső FIFO (First In, First Out) tárolóként fogja látni. A kommunikációért felelős program végtelen ciklusban fog futni és emiatt szükséges az újonnan bejövő információt egy tömbben eltárolni.

Az FPGA belső felépítése:

A kezdeti elgondolás a következő volt. A kommunikáció állapotának meghatározásához szükséges valamilyen handshake. Az FGPA a „fogadásra kész” kimenetén jelzi a mikrokontroller felé, hogy küldheti a következő képi információt. Ezután a beágyazott számítógép az „átvitel” vonalon jelzi, hogy indítja az átvitelt, ezzel egyidejűleg az FPGA lekapcsolja az „fogadásra kész” portját és addig nem kapcsolja vissza, amíg a teljes képet át nem töltötte a belső memória blokkba. Amint a mikrokontroller végzett az átvittel lekapcsolja az „átvitel” lábat, ezzel jelezve, hogy a küldési folyamat véget ért, az átmeneti tároló tartalma áttölthető a belső memóriába a megfelelő időben.



11. ábra Az FPGA belső blokkvázlatának terve

Ez a blokkvázlat annyiban módosult az implementáció során a tervezethez képest, hogy az átvitel bemenet használata nem volt indokolt. Ha a fogadásra kész vonal logikai 0 értékű, az FPGA végzett az előző kép kiküldésével.

4.1.4 Az FPGA belső rendszerterve

A mikrovezérlő az FPGA-t egy FIFO-ként fogja látni. A LED-vezérlő áramkör felépítéséből adódóan, amikor nem kap új képinformációt, akkor a legutoljára kapott adat marad megjelenítve. Ebből következik, hogy a kijelző összes LED-jének kikapcsolása egy „üres” képkocka kiküldésével valósítható meg. A beágyazott számítógép szoftverének cél egy tárolóból kiküldeni az adatokat az FPGA-ba. A tároló feltöltése kívülről történik, ez már nem része ennek a projektnek.

A mikrovezérlő egy Raspberry Pi B+, amely elég erős hardverrel rendelkezik egy operációs rendszer futtatásához. Én a Raspbian-t telepítettem fel rá. Ez egy ingyenesen használható Debian alapú rendszer kifejezetten ehhez a beágyazott számítógéphez optimalizálva.

A szoftver feladat egy végtelen ciklusban vizsgálni, hogy érkezett-e új képinformáció. Amikor érkezett ellenőrizni, hogy az FPGA készen áll-e a fogadására, majd SPI-on keresztül átküldeni a FIFO-ba.

5 Hardvertervezés

5.1 Felhasznált hardverkomponensek

5.1.1 Spartan 6 FPGA

Az általam választott FPGA a Xilinx gyártmányú Spartan 6-os család XC6SLX9 típusú egysége. Az integrált áramkör főbb jellemzői:

- Mag tápfeszültsége: 1.14 – 1.26 V
- Ki- és bemenetek tápfeszültsége: 3.3 V
- Tokozás: TQFP-144
- Ki- és bemeneti lábak száma: 102
- Logikai blokkok száma: 1430
- Makrocellák száma: 9152
- Maximális működési frekvencia: 375 MHz
- Teljes memória mérete: 576 kB

Korábbi tanulmányaim során már használtam ezt a családot, igaz az akkori darab egy kisebb teljesítményű Spartan 3-as áramkör volt, de a programozása nagyon hasonló a két IC-nek és a fejlesztőkörnyezet pedig ugyanaz.

A Spartan 6-os család széles skálával rendelkezik erőforrások terén. A projektemhez én az LX9-es verziót választottam. Ez a család második legkisebb tagja (az LX4-es típus után). A nevében lévő 9-es azt jelenti, hogy közel 9000 logikai cellát tartalmaz. Az LX4-es előzetes becslés alapján kevés erőforrást tartalmazott így ez nem jött szóba. Nagyobbat viszont azért nem volt érdemes használni, mert azok BGA tokozásúak, így beültetésük és cseréjük komplex feladat.

5.1.2 Raspberry Pi B+

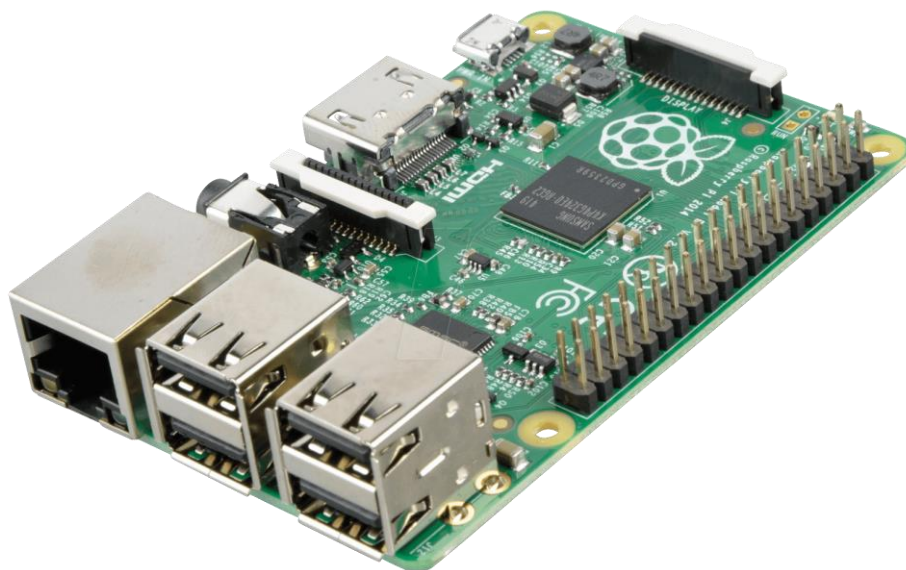
A mikrovezérlő kiválasztása előtt összegyűjtöttem mik az elvárások a vezérlőtől. Legyen egy erős processzora, ami elég egy komplexebb operációs rendszer (pl. Linux) futtatásához. Rendelkezzen SPI interfésszel, hogy egyszerűen megoldható legyen a kommunikáció az FPGA-val. Szükség van néhány GPIO lábra. Ezek alapján a Raspberry Pi egy optimális választás- Rendelkezik saját Linux disztribúcióval, megfelelő hardverelemekkel és az ára is kedvező.

A Raspberry Pi B+ a B verzió továbbfejlesztése. Az átervezés során lecserélték a 26 lábás tűkesort 40 lábásra, de az első 26 láb kiosztása nem változott, így kompatibilis maradt az elődjével. Az előző verzió 2 USB port-ja helyett 4-el szerelték fel. A háttértárként használt SD memóriát microSD-re cserélték. Az átervezés során a lineáris feszültségszabályzókat lecserélték kapcsolóüzemű tápokra. Ezzel 0.5-1 W-tal sikerült csökkenteni a fogyasztást.

Az eszköz főbb technikai paraméterei:

- System on Chip: Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, és egy USB port)
- CPU: 700 MHz ARM1176JZF-S core (ARM11 család, ARMv6 utasításkészlet)
- GPU: Broadcom VideoCore IV
- SDRAM: 512 MB (megosztva a GPU-val)
- 4x USB 2.0 port
- MIPI kamera csatlakozó
- Video kimenetek: HDMI (rev 1.3 & 1.4) felbontás 640×350-tól 1920×1200-ig PAL és NTSC szabvány, kompozit videó (PAL és NTSC) a 4 pólusú TRRS jack-csatlakozón az analóg hanggal megosztva
- Audió kimenet: HDMI digitális és sztereó analóg a 4 pólusú TRRS jack csatlakozón
- Belső tároló: MicroSD

- LAN: 10/100 Mbit/s Ethernet (8P8C)
- GPIO: 17× GPIO, UART, I²C, SPI bus 2 chip select-tel, I²S audio, +3.3 V, +5 V, föld
- Fogyasztás 3W (600mA)
- Méret: 85,60×56,5mm
- Súly: 45g



12. ábra Raspberry Pi beágyazott számítógép

5.2 LED-panelek tervezése

Fogyasztás:

A tervezet szerint a LED-ek 5 V-os tápról fognak működni. Ilyen feszültség mellett a LED-meghajtó IC maximum 32 mA-t tud rajtuk kapcsolni. Worst-case esetben, ha minden LED maximális kitöltési tényezővel működik, egy panel fogyasztása a következő egyenlettel számítható:

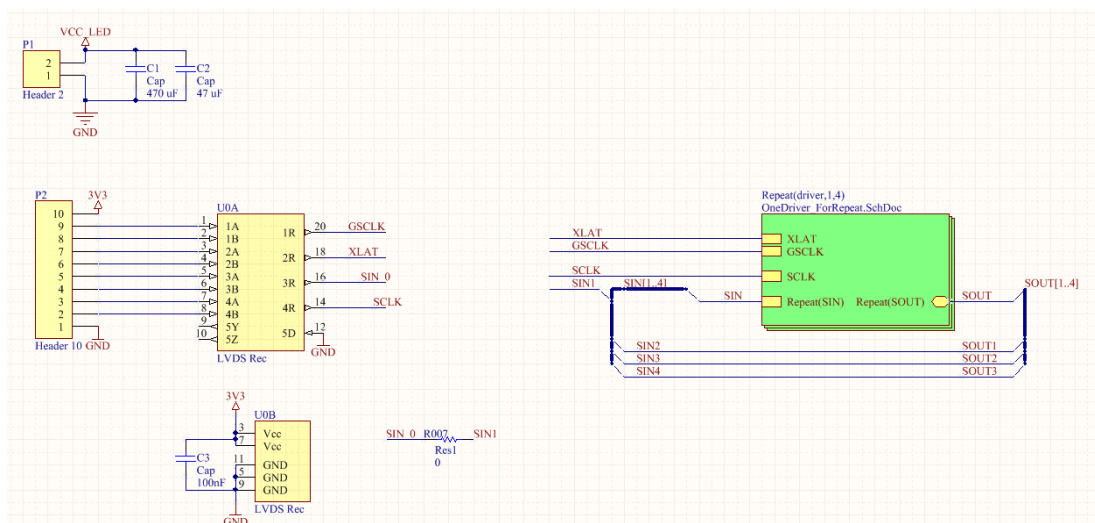
$$P_{total} = U \cdot \sum I_{max} = 5 \cdot 32 \cdot 10^{-3} \cdot 4 \cdot 48 = 30.72 \text{ W}$$

A LED-meghajtó IC-k külön tápról működnek, de a 3 V-ról legfeljebb 20 mA-t vesznek fel, ez panelenként 0.36 W.

A teljes kijelző maximális fogyasztása: $25 \times (0.36 + 30.72) = 777 \text{ W}$

Ebbe még nincs beleszámolva a vezérlő panel teljesítmény felvétele, bár ez jelentősen kisebb lesz, mint amennyi a LED-ek működtetéséhez szükséges.

Az alábbi kapcsolási rajzok az Altium Designer-ből lettek exportálva.



Az azonos kapcsolási modulokból elég egy darabot megrajzolni, majd megadni, hogy hány darabra lesz szükségünk és a NYÁK-terv készítésekor a többi automatikusan legenerálódik. A LED-vezérlők Daisy chain-be lesznek felfűzve, ezért össze kellett kötni az IC-k kimenetét a következő bemenetével. Erre sajnos nem találtam elegánsabb megoldást a fejlesztőkörnyezetben, mint direktbe összehuzalozni a blokkon kívül.

LED panel tervezése, első verzió:

A fenti ábrán nem szerepel mind a 48 LED (de a kapcsolási rajz nem látható részén igen), hogy a feliratok olvashatóak legyenek. A panelek NYÁK-tervének elkészítésénél nem a kapcsolási rajz összeállítása jelentette a nehézséget, hanem a huzalozás megvalósítása. Egy oldalas NYÁK-hoz képest nagyon sűrűn voltak az alkatrészek és a vezetékek a LED-vezérlő IC környékén, így az egész tervezést innen közelítettem meg. Az alkatrészek elhelyezésének rövid átgondolása után arra jutottam, hogy elég, ha a LED-ek kivezetéseinek számára van elegendő hely horizontális irányban, az adatvonalak bekötése pedig vertikálisan történik a szintén horizontális irányú adatbuszra. Ez a módszer önmagában még nem lett volna elég, hogy a teljes rajzolat huzalozásához. Egy síkban felhasználtam néhány 0 Ohm-os ellenállás, hogy egy extra réteget nyerjek a vezetékek kereszteződéseinél. Ezek 2512-es méretű SMD alkatrészek, hogy a lehető legtöbb huzalt lehessen átvinni alattuk.

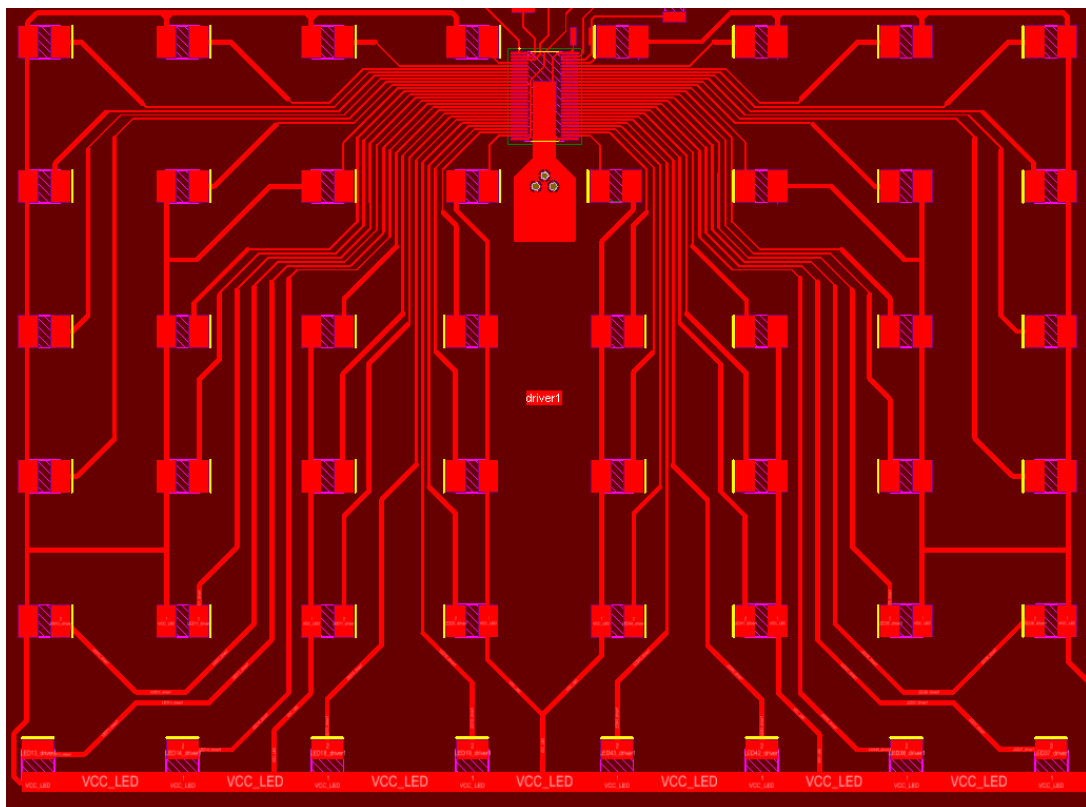
A következő ábrán látható egy darab LED-meghajtó és a környezetének első verziós huzalozása. A teljes panel A3-as nagysága miatt nem illeszthető be ide jelentős kicsinyítés és a látvány minőségének romlása nélkül. A teljes A3-as NYÁK-terv a függelékben található.

Az ábrán felül látható az adatbusz, ami a panel közepén halad és erről lecsatlakoznak a vezérlő jelek minden IC-hez. Látható, hogy a LED-ek négyzetláncosan vannak elhelyezve, mert így a megjelenítendő kép megtartja az eredeti arányait. A LED-meghajtó lábaitól maximum 0.3 mm-es vezetékeket tudtam húzni, viszont, amint lett hely megnöveltem a szélességet 0.8 mm-re. A panel külső szélén húzódik a LED-ek tápvonala. A panel nem lett teljesen egy oldalas, mert az alsó rézréteg csak földpotenciálnak van használva. Ez az ábra közepén található kitöltés és rajta keresztül a LED-meghajtó földpontjával galvanikusan össze van kötve. Sajnos ezt csak furattal lehet megoldani, de panelenként néhány furat még kezelhető mennyiség.

LED panel tervezése, második verzió:

Ahhoz, hogy az összes LED számára biztosított legyen a tápellátás 0 Ohm-os átkötő ellenállások vannak az első verzióba betervezve. Ez azért szükséges mivel a panel hátoldalán nincs rajzolat. Ennek a verziónak a nagy hátránya volt, hogy a LED vezérlő és az adatbusz közötti 2 db LED-hez a szomszédosról kellett volna egy külső vezetékkel bekötni a tápot. Ez az utólagos munkálat 200 db LED-et érintett volna (25 panel esetén, panelenként 4 IC-vel és LED-vezérlőnként 2 darab LED-del számolva). Ez után készült egy optimálisabb huzalozás, ahol a tápvezetékek fésűszerűen vannak bekötve LED-vezérlő kimenetei között. Ezzel a megoldással elegendő egy 0ohm-os ellenállás felhasználása a tápvezetékeken, ami egyszerűbb gyárthatóságot biztosít. Erre az egy darabra az FPGA felől érkező adatbusz csatlakozója miatt van szükség. Az átdolgozott verzióban a LED driver IC közelében lévő huzalok egyforma távolságra lettek elhelyezve. Amellett, hogy esztétikusabb az elrendezés megkönnyíti a gyártási hibák javíthatóságát is.

További módosítása történt a LED-ek footprint-jén is. Szélesebbre rajzoltam őket, mivel az első verzióban túlságosan is pontosan illeszkedett a LED fizikai méretéhez, ami megnehezítette a kézi forrasztást.



14. ábra LED-panel huzalozása, 2. verzió

Az áramkör belső regisztereinek speciális sorrendje van, az első 16 bit adat nem az első kimeneti lábat vezérli. Ehhez jön még az elrendezésből adódó átrendeződés. Ezek az okok miatt szükséges volt egy eltolási tábla meghatározása, amely megmondja, hogy egy LED fényerejét hányadik 16 bit-es csomag adja meg. A következő táblázat ezt foglalja össze:

Oszlop Sor	1	2	3	4	5	6	7	8
1	46	35	36	34	22	24	23	10
2	48	33	32	26	14	20	21	12
3	47	43	28	27	15	16	7	11
4	31	45	30	25	13	18	9	19
5	40	44	29	38	2	17	8	4
6	42	41	37	39	3	1	5	6

A táblázatból leolvasható, hogy az MSB után érkező első 16 bit a hatodik oszlop hatodik sorában található LED fényerejét adja meg.

Látható, hogy a LED-ek fizikai helye és az adatsorban lévő helye között nem áll fenn egyszerű matematikai kapcsolat. Ennek alapján úgy döntöttem, hogy az adatok megfelelő sorrendbe helyezését a beágyazott számítógépnek kell elvégeznie, ezért is volt elvárás, hogy nagy teljesítményű processzorral rendelkezzen. Ez a táblázat csak egy LED-vezérlő áramkörre vonatkozik, a teljes panel eltolási táblája a függelékben található. A kommunikációs vonalak csatlakozója az első oszlopban található a hatodik és hetedik sor között.

Egy teljes panel áramfelvétele teljes terhelés esetén nem elhanyagolható, így a tápvezetékek huzalszélességénél ezt figyelembe kell venni. Ehhez az IPC-2221-es szabványban rögzített adatokat használtam. Külső rétegre a szabvány a következő képlettel definiálja a szükséges vezeték szélességeket:

$$I = 0.024 \times dT^{0.44} \times A^{0.725}$$

I: maximálisan megengedett áram [A]

dT: hőmérsékletemelkedés a környezethez képest [°C]

A: huzal keresztmetszet [mil²]

A fenti képlet használata kézi számolás esetén bonyolult módszer, főleg mivel angolszász mértékegységeket is használ. Sokkal praktikusabb egy, a fenti képlet alapján

készített kalkulátor igénybevétele. Amennyiben adott áramértékhez szeretnénk huzalszélességet számolni, a kapott érték a minimális vezetékszélesség lesz. Tápvezeték esetén érdemes a minimálisnál szélesebb huzalt alkalmazni, ahol van rá lehetőség. A szélesebb huzal ellenállása kisebb, így a rajta keletkező veszteség és feszültségesés is kevesebb.

Legkritikusabb pont a tápcsatlakozó és a környéke, mivel itt a legnagyobb a Worst-case áram. Ennek az értéke $192 \text{ LED-re } 192 \times 32 \text{ mA} = 6.144 \text{ A}$. Ezt az áramot nem egy vezetéknek kell elszállítania, hiszen rögtön a csatlakozótól két irányba elágazik, így az ágakra 3.072 A terhelés jut (96 LED árama). A fenti képlet alapján $35 \mu\text{m}$ -es rézvastagsággal (ez felel meg a standard 1 uncia/négyzetláb, oz/ft² értéknek) és maximum 20 fokos hőmérsékletemelkedéssel kalkulálva 3.67 mm minimális huzalvastagságot kaptam. Ez alapján én a 4 mm -es vezetékvastagságot választottam.

Egy másik szűk keresztmetszet a saroknál bekanyarodó rész. Itt már 13-mal kevesebb LED árama folyik, ami $83 \times 32 = 2.656 \text{ A}$. Ehhez az áramértékhez 3.01 mm a szükséges minimum vastagság. Mivel folyamatosan csökken a maximális áramigény a tápvonalon előre haladva és a gyártási technológia miatt nem ajánlott a NYÁK szélén huzalt vinni, ezért a panel hosszabbik oldalán 3 mm vastag a fő tápvonal.

Ezt az áramot vissza is kell vezetni a földpontba. A földkitöltés a panel alsó oldalán található, ezért egy vezérlőhöz praktikusabb több kisebb vián keresztülvinni az áramot egy nagy helyett. Én három darab 1 mm átmérőjű viát használok erre a célra.

5.3 Vezérlő board tervezése

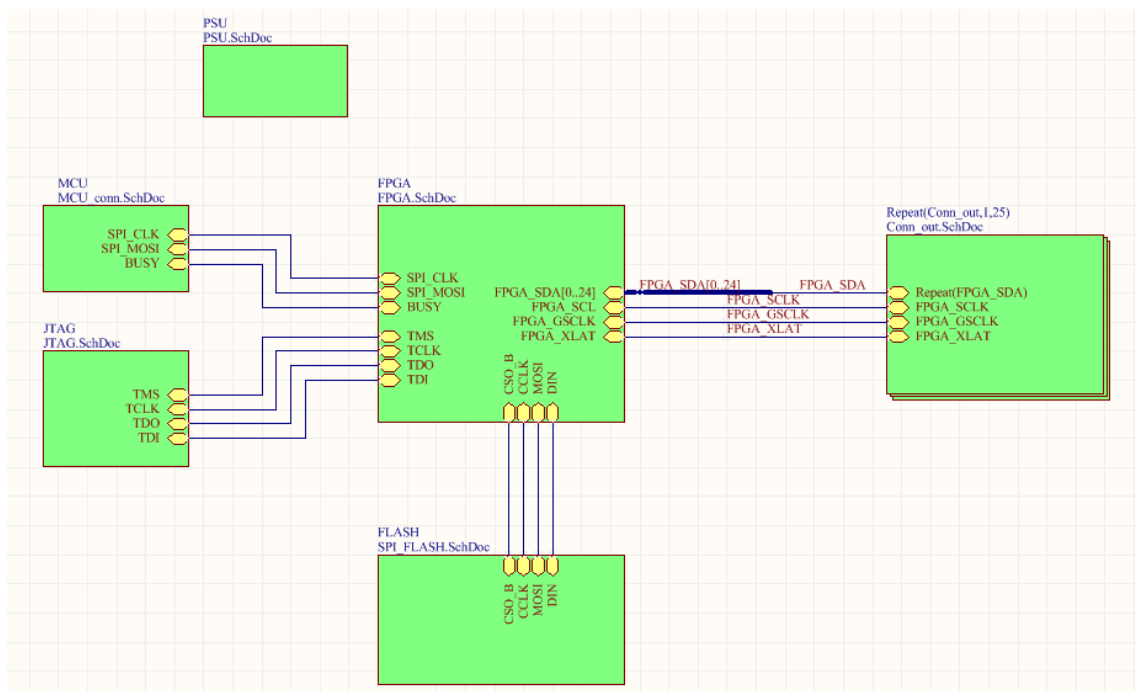
Ez a NYÁK lesz a központi elem, amely összekapcsolja a LED paneleket a FPGA-val és a külvilággal. Az egység tartalmazza a Spartan 6-os FPGA-t, az FPGA működéséhez szükséges tápáramkört, a kimenetei csatlakozókat, a kimenetek meghajtásához szükséges LVDS áramköröket, az FPGA programozó interfészét (JTAG) és a Raspberry Pi csatlakozóját.

A Raspberry Pi úgy lett kialakítva, hogy felülről lehessen rá egyszerűen illeszteni a kiegészítő áramköröket és csatlakozókat. A központi NYÁK viszont jóval nagyobb méretű lesz ezért célszerű, hogy az legyen alul. A Raspberry Pi-on található USB- és Ethernet csatlakozójának fémháza magasabb, mint a tűskesor, amire az SPI van kivezetve. Mechanikailag kivitelezhető megoldás lehetne, hogy a fémházas csatlakozók

ne a NYÁK felett legyenek, de én egy másik megoldást választottam, néhány perces munkával megfordítható a tükörsor. Ebben az esetben nem lóg le a NYÁK-ról a Raspberry Pi és a power LED-je is felfelé világít.

A külső táp egy 5 V-os hálózati adapter 2.5 x 10 mm-es jack csatlakozóval. A választást az indokolta, hogy számos helyen használják, egyszerűen csatlakoztatható és helyettesíthető.

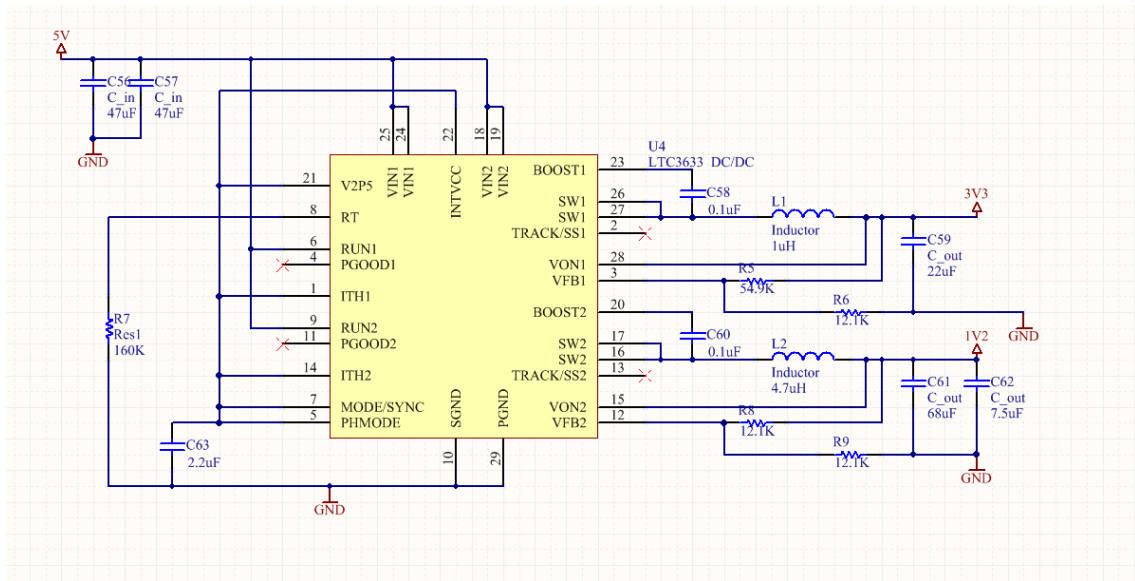
A vezérlő board blokkvázlata az alábbi ábrán látható:



15. ábra a vezérlő board blokkvázlata

5.3.1 Tápáramkör

A FPGA működéséhez két különböző feszültségű tápra van szükség. 3.3 V-ról működnek a kimenetek és 1.2 V szükséges a mag ellátásához. Elterjedt megoldás egy kapcsolóüzemű táp beépítése egy lineáris táppal szemben. Csak feszültség csökkentésre van ugyan szükség, de kapcsoló üzemű táp jobb hatásfokkal rendelkezik (kivétel ez alól, ha alacsony terhelő áram van, de ebben az esetben ez nem üzemszerű állapot). Alkalmazható két külön álló áramkör is a tápellátásra, akár kaszkádosítva is. Én egy olyan megoldást választottam, ahol a két táprész egy IC-be van integrálva. A kapcsolási rajz az következő ábrán látható:



16. ábra az FPGA tápáramkörének kapcsolási rajza

A bemenet a fentebb is említett 5 V-os feszültség szint, ez csatlakozik az IC V_{IN1} és V_{IN2} lábaira. A $PGOOD_{1,2}$ kimeneti láb, azt jelzi, hogy a feedback láb a referencia értékhez képest (0.6 V) 8%-on belül van-e. Kimeneti láb, aminek nem használok fel semmire, ezért nem szükséges fixen bekötni. A RUN lábak a két csatorna bekapcsoló bemenetei. Akkor engedélyezi a csatornákat az legalább 1.22 V feszültség esik rajta. Ebben a projektben ez az FPGA fő tápja és folyamatosan működik. Energiatakarékosan működő, például akkumulátoros, rendszerek esetén kívülről vezérelhetőek, így az inaktív eszközök tápellátását lekapcsolva csökken a rendszer fogyasztása. Ebben a projektben ez a funkció nem releváns, ezért célszerű volt bekötni a fixen az 5 V-os tápfeszültségre. Ez biztosítja, hogy amint a fő táp bekapcsol az FPGA is elinduljon. Az $ITH_{1,2}$ és az $INTVCC$ lábak a szabályozókörhöz szükségesek. Lehetőség van a külső szabályozó kör elhagyására, ha elegendő a beépített szabályozás. Én csak a beépített szabályzást használok, ehhez össze kell kötni ezeket a lábakat és egy kondenzátorral a földre kapcsolni. A PHMODE (0 vagy 180° fázistolás legyen a két kimenet között) és MODE/SYNC (külső szinkronizáló input) lábak funkcióira sincs szükség, mivel bemeneteket nem célszerű lebegve hagyni, ezért a szabályozó kör kapacitásán keresztül ők is a földhöz vannak kötve.

Az RT láb a belső oszcillátor frekvenciáját beállító láb. A szükséges ellenállás értéket úgy lehet megkapni, hogy $3.2 \times 10^{11} / \text{kapcsolási frekvencia}$. Én az adatlapon javasolt 2 MHz körüli értéket állítottam be, ehhez szükséges egy 160 k Ω -os ellenállás. 1%-os tűrésű ellenállásból elérhető a 160 k Ω , így ezt választottam.

Az SW_{1,2} láb az impulzusmodulált kimenet, amely a tekercset és kimeneti kondenzátort táplálja. A BOOST_{1,2} bemenetre az adatlap egy 0.1 µF-os boost kondenzátor bekötését javasolja, amelyik másik lába az SW_{1/2} kimenetre van kötve, lehetőség szerint minél rövidebb huzalozással. A VON_{1,2} lábakat a tekercs kimenetére kell kötni, a bekapcsolt periódusban ezen keresztül tudja az IC visszamérni a kimeneti feszültséget.

TRACKSS_{1,2} a soft start-hoz felhasználható lábak, ennek a funkciónak a használata opcionális, ezért én nem kötöttem be őket.

A kapcsolóüzemű tápok rendelkeznek egy minimális áramhullámossággal a működési elvükből következően. Ezt befolyásolja a be- és a kimeneti feszültség, a kapcsolási frekvencia és az induktivitás nagysága a következő képlet szerint:

$$L = \left(\frac{V_{OUT}}{f \times \Delta I_{L \max}} \right) \cdot \left(1 - \frac{V_{OUT}}{V_{IN}} \right)$$

A 3.3 V-os kimenet esetén 3 A maximális áramterhelésnél 40%-os áramhullámossággal és 2 MHz-es kapcsolási frekvenciával számolva 1 µH-s tekercs szükséges. Ahhoz, hogy a mag tápellátásánál kicsi legyen az áramhullámosság, érdemes nagyobb tekercset alkalmazni. 1.2 V-os kimenetnél 1 A maximális áramterhelés esetén viszont 10% áramhullámossággal kalkulálva 4.7 µH-s tekercsre esett a választás.

A kimeneti puffer kapacitás értéke befolyásolja a kimeneti feszültség hullámosságot.

$$C_{OUT} = \frac{3 \cdot \Delta I_{OUT}}{f \cdot V_{DROP}}$$

A 3.3 V-os kimenetre kiszámolva 200 mV feszültség esésre méretezve 22.5 µF-os, az 1.2 V-os kimenetre 20 mV-os feszültség esést kalkulálva 75 µF-os kapacitás szükséges. A mag tápfeszültsége érzékenyebb ezért méreteztem ez sokkal stabilabbra, a szűrőkondenzátorok használata mellé.

5.3.2 FPGA kapcsolási rajzának terve

A teljes kapcsolási rajz a helyigénye miatt a függelékben tekinthető meg. Az FPGA felprogramozására többféle lehetőség is van. Ebben az esetben fontos volt, hogy a rendszer bekapcsolásakor ne egy kezelőnek kelljen elvégezni bitstream letöltését, hanem ez egy automatikus folyamat legyen. Én azt a lehetőséget választottam, hogy FPGA bitstream-je egy SPI-os Flash memóriában van. Ehhez 4 lábat szükséges bekötni. Az áramkör órajelét a 70-es lábra, az MOSI-t a 64-es lábra, a DIN-t a 65-ös lábra és a CSO_B-t a 38-as számú lábra. A kialakítás az FPGA adatlapja alapján történt. Szintén ez alapján kell a helyes működéshez az INIT_B lábat (39), a DONE lábat (71) és a PROGRAM_B lábat (37) egy felhúzó ellenálláson keresztül tápra kötni, a HSWAPEN lábat (144) pedig közvetlenül a földhöz csatlakoztatni.

Ez még önmagában nem elég a felprogramozáshoz. Szükség van a JTAG bekötésére is, mert az Xilinx fejlesztőkörnyezetében ezen keresztül történik a letöltés. Itt kell hozzáadni a SPI memória használatát is és a felprogramozás során az FPGA-n keresztül töltődik fel a Flash memória. A JTAG-ből szükséges lábak a TDO, a TDI, a TMS és a TCK. Fizikailag a TDO-t a 106 lábhoz, a TDI-t a 110-es lábhoz a TCK-t a 109-es lábhoz és a TMS-t pedig a 107-es lábhoz kell kapcsolni.

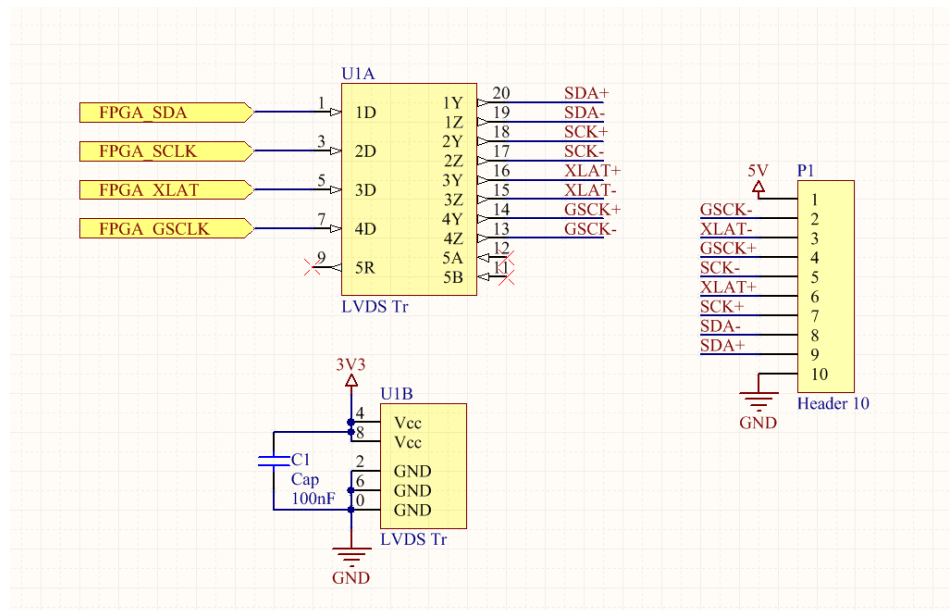
Az FPGA tápellátása 6 csoportra osztható. Külön van egy-egy csoport a 4 bank-nak (VCCO_0,1,2,3), egy kiegészítő táp (VCCAUX) és egy csoport, ami a mag tápellátását biztosítja (VCCINT). Fontos a stabil tápfeszültség, ezért a csoportokat szűrőkondenzátorokkal kell ellátni. A csoportok felépítése azonos. Ajánlott minden csoportra a becsatlakozási pontjára elhelyezni egy 4.7 μ F-os és egy 470 nF-os kondenzátort, úgy hogy a 4.7 μ F legyen a tápegység felől. Ezen kívül minden láb mellé, minél közelebb a lábhoz, szükséges egy 47 nF-os és egy 10 nF-os szűrőkondenzátor, a lábhoz közelebb a 10 nF. Minél kisebb egy kondenzátor, annál nagyobb frekvenciás zavarokat tud hatékonyan szűrni, ezért szükséges, hogy a legkisebb értékű legyen a legközelebb egy áramkör lábához. A hatékony szűrés érdekében a kondenzátorokat Y-kapcsolásban érdemes bekötni.

A LED-panel SPI vonalait a 0-s és 1-es bankon helyeztem el a 102-es és a 142-es láb közötti I/O lábak felhasználásával. A programozó lábak fixen a 2-es bankban vannak, ezért volt célszerű a kimeneti SPI-ok elhelyezése az IC másik oldalára. A 2-es

bank-ban került elhelyezésre a Raspberry Pi 3 csatlakozó vonala is. Ez azért előnyös kialakítás, mert a NYÁK-terven elkülöníthetőek a funkciók és a huzalozás.

5.3.3 Kimeneti csatlakozó a LED panelek felé

Minden LED-panel számára külön csatlakozó található. A repeat sheet funkció használatával elegendő egyet megrajzolni és tetszőleges számú példányt lehet belőle generálni.



17. ábra Kimeneti csatlakozó és LDVS áramkör kapcsolási rajza

LDVS adó áramkörnek a Texas Instruments SN65LVDT41 típusú IC-jét használtam. Ezt az IC SPI busz meghosszabbításhoz ajánlották, ennek megfelelően képes egy Master és egy vagy két Slave egység között biztosítani a kommunikációt. Egy áramkör a Master felől 4 vonalat képes átvinni (MOSI, SCLK, CE0, CE1) és a MISO vonalat fogadni a slave egységtől. Én Chip Enable jelek helyett a GSCLK-t és az XLAT vonalakat kötöttem be a bemenetre, a számomra nem releváns MISO vonalat fogadó részt bekötetlenül hagytam. Ez az IC pontosan annyi vonal átvitelére képes, mint amennyit a projekt igényel, ezért sokkal optimálisabb megoldás, mint 1 vagy 2 csatornás LVDS áramkörök használata. 2 táp lába van ezért EMC szempontból előnyösebb lenne egy-egy dedikált szűrőkondenzátor használata mindkét bemenetre, de ezek közel helyezkednek el egymáshoz, ezért az egyiket elhagytam.

Csatlakozónak a 10 lábás Minifit típust választottam. Ez egy elterjedt típus, masszív és akár több A-t is képes átvinni.

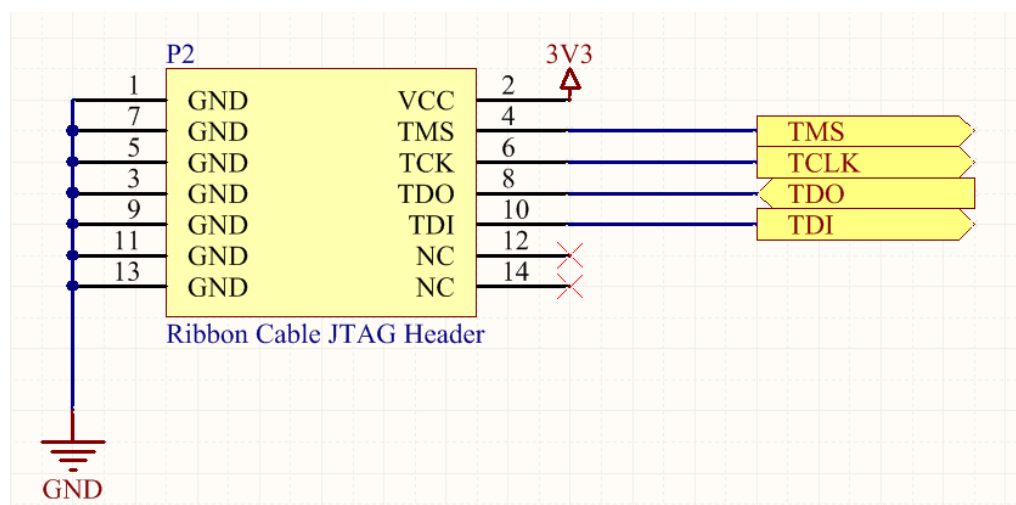
5.3.4 JTAG (Joint Tag Action Group) csatlakozó

Ez egy szabályos interfész, használata elterjed a beágyazott rendszerekben. Kétirányú soros kommunikációt valósít meg. Előnye, hogy egy interfészen keresztül lehetőség van programozásra, tesztelésre és debuggolásra is.

A táp és a föld lábakon kívül 4 láb bekötése szükséges, ahhoz hogy az FPGA felprogramozható legyen:

- TMS (Test Mode Select)
- TCK (Test Clock)
- TDO (Test Data Out)
- TDI (Test Data In)

A lábkiosztása az alábbi ábrán látható.



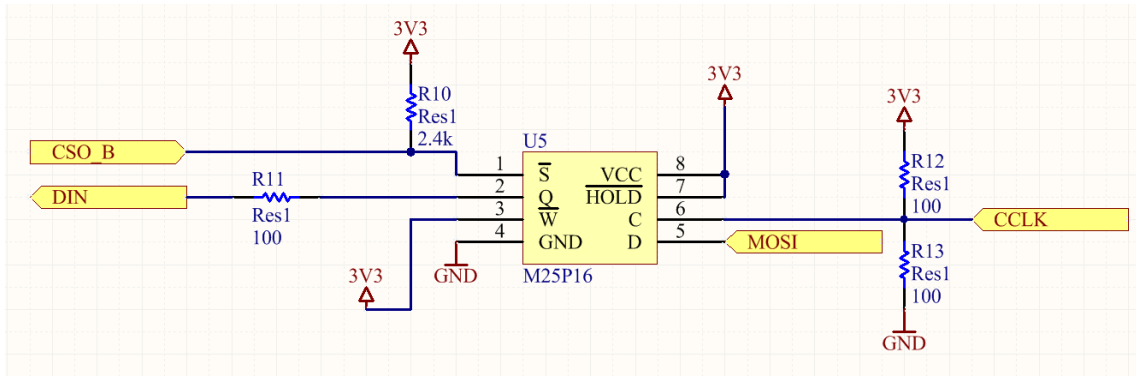
18. ábra JTAG csatlakozó

5.3.5 SPI FLASH memória

Ez a memória tárolja az FPGA felprogramozásához szükséges információt. A felprogramozás minden rendszerbekapcsoláskor automatikusan megtörténik.

A memória SOP8-as tokozású IC, amelyben 16 Mbit adat tárolható maximálisan. Az 1-es lába a chip select, a 2-es a sorod adat kimenet, a 3-as az írás

engedélyező alacsony aktív bemenet, az 5-ös a soros adat bemenet, a 6-os az soros órajel és a 7-es láb lehúzásával pedig a kommunikációt lehet ideiglenesen szüneteltetni.

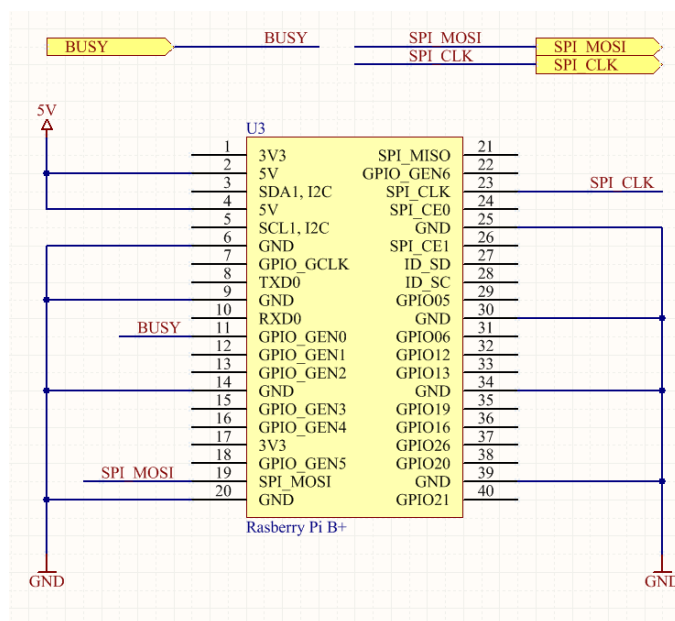


19. ábra az SPI Flash memória kapcsolási rajza

Az kapcsolási rajz a Spartan 6 adatlapja alapján történt. Az órajel huzalozásához 50 Ω -os hullámimpedanciájú vezetékét ajánlanak. A memóriához közel javasolnak egy 2 hullámimpedancia nagyságú fel- és lehúzó ellenállást. A HOLD és W lábak funkcióira nincs szükség, direktbe rá lehet kötni őket a tápra.

5.3.6 Raspberry Pi B+ csatlakozó

A Raspberry Pi B+ csatlakozója egy 2.54 mm-es tűkesor anyacsatlakozója. A teljes csatlakozó 2x20 pin-es Ezek közül én 3 jelvezetékét használok. Az SPI órajel (SPI_CLK) és az SPI MOSI fix helyen vannak. Az FPGA Busy kimenetét bármelyik GPIO lábra rá lehet kötni.



20. ábra Raspberry Pi B+ csatlakozója

5.4 FPGA hardverterve

Az FPGA-nak a következő feladatokat kell ellátnia:

- Adatok fogadása SPI-on
- Kapott információ tárolása a képinformáció megjelenítésének végéig
- LED driver áramkörök inicializálása
- LED driver áramkörök periodikus feltöltése kapott képinformációkkal és a vezérlőjelek kiadása

5.4.1 Az SPI interfész bemutatása

A soros periféria interfészt (SPI) a Motorola fejlesztette ki. A kommunikáció full-duplex (mindkét irányba történhet kommunikáció egy időben) és Master-Slave felépítésű. Elsődlegesen beágyazott rendszerekben használják, ahol nincs szükség nagy távolságú adatátvitelre. Az átviteli sebesség elérheti akár az 50 Mbps-ot is. Az SPI interfész 4 vezetéssel dolgozik:

- MOSI (Master Out Slave In)
- MISO (Master In Slave Out)
- SCLK (soros órajel)
- SS (Slave Select)

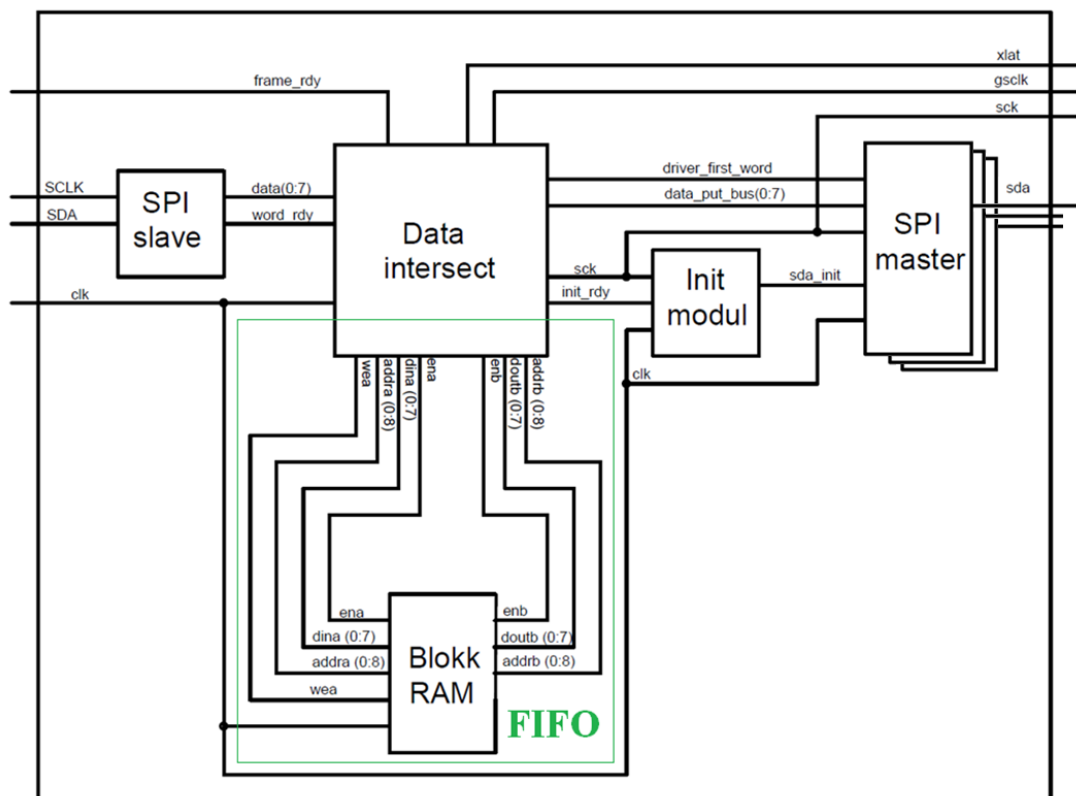
A kommunikációhoz a Master egység biztosítja az órajelet, ezért tervezéskor figyelembe kell venni mind a Master mind a slave egység képességeit. A Master a Slave Select alacsony aktív jel lehúzásával jelzi a Slave egységnek, hogy vele kommunikál. Az órajel élére (módfüggő, hogy felfutó- vagy lefutó élre) történik az adat mintavételezése a MOSI és a MISO csatornán, a következő élre pedig az adatvonal értékének esetleges megváltozása. Az átvitel általában 8 bites szavakban történik, de ez nem rögzített szabály. Lehetőség van a Slave Select vonal elhagyására és az egységeket úgynevezett Daisy Chain-be felfűzni. Ez azt jelenti, hogy minden egység MISO vonala a következő egység MOSI vonalára csatlakozik. Ezzel egy hosszú körláncot kapunk, amit továbbra is a Master vezérel, de a kiküldött adat ekkora körbe megy minden egységen.

Az SPI interfésznek 2 paramétere van. Az egyik a CPOL (Clock Polarity), ami meghatározza, hogy inaktív állapotban a busz órajele logikai alacsony vagy magas szinten van. A másik a CPHA (Clock Phase), ami definiálja, hogy adatbuszon kint lévő információ mintavételezése az órajel felfutó (CPHA = 0) vagy lefutó (CPHA = 1) élére történik. Ezek alapján 4 féle működési mód létezik. Ezeket az alábbi táblázat foglalja össze.

SPI Mode	Clock Polarity	Clock Edge
0	0	1
1	0	0
2	1	1
3	1	0

5.4.2 A fejlesztés folyamata

Az alábbi ábrán látható az FPGA belső blokkvázlata.



21. ábra: Az FPGA belső felépítése

Az FPGA teljes belső működése az órajel felfutó élére szinkronizálódik.

Első lépésben az SPI ki- és bemeneti moduljait programoztam le. Ezek egyszerű felépítésű modulok és található hozzájuk kiindulási alap az interneten. Az első verzióban a kimeneti modul tartalmazta a LED-vezérlő IC-k felfelprogramozásához szükséges 769 bit-es adatot, de később ezt kivettem belőle, mivel több kimeneti modulra volt szükség és erőforrás pazarlás ezt az adatot minden modulban külön tárolni. A modul tartalmaz egy 17 bit-es adatpuffer-t, egy 6 bit-es számláló regisztert, egy-egy regisztert, amiben a kimeneti SPI adatvonal és órajel értéke van eltárolva és egy 1 bit-es állapotjelző regisztert, ami a központi modul felé jelzi, hogy folyamatban van-e adatátvitel. A kész modult az ISE Project Navigator beépített szimulációs eszközével teszteltem. A teszt során az előző részben bemutatott dobozra lehet időzített bemeneteket adni. Ezután lehet megfigyelni az input-ra adott kimeneti jeleket és a belső állapotokat is.

Az SPI slave modul tartalmaz egy 8 bit-es adat puffert, egy 5 bit-es számláló regisztert, egy 2 bit-es regisztert a külső SPI órajel mintavételezéséhez és egy státuszregisztert, amivel jelzi, hogy ha egy teljes csomag megérkezett.

A blokk memória generálása egy beépített tool segítségével történt, ennek használatáról részletes leírást lehet találni a Xilinx honlapján. Első lépésben el kellett dönteni az eszköz interfészének típusát. Két lehetőség közül lehet választani, vagy natív interfésszel fog rendelkezni a blokk RAM, vagy AXI4 buszon keresztül is lehet hozzá csatlakozni. Mivel az FPGA belső felépítését teljesen én terveztem és nem tartalmazott soft-core processzort az AXI4-es busz csak felesleges bonyolítást jelentett volna az adatok eltárolásánál és visszatöltésénél, ezért én a natív interfészt választottam. A következő lépésnél kell megadni a generálandó memória típusát. Itt ki lehet választani, hogy RAM-ra vagy ROM-ra (Read Only Memory, csak olvasható memória), illetve hogy single-, simple dual- vagy true dual port memória legyen. Ennél a projektnél egyértelműen írható memóriára volt szükség, ezek közül pedig a simple dual port memóriát választottam, ahol a külön adat-, cím- és órajel bement van az íráshoz és az olvasáshoz. Értelmszerűen az olvasásért felelős rész adatcsatornája kimenet és ehhez a port-hoz nem tartozik írás engedélyező bemenet se. A blokkmemória méretét az elérhető maximálisra állítottam, ami 2 x 8 kB. Ekkora méretű egységekből egy képkocka tárolásához ($8 \times 4800 = 38400$ bit) 4 db ilyen blokkra van szükség. A Spartan 6 LX9-es FPGA-ban 32 darab 16 kB-os blokk áll rendelkezésre így ez nem lesz szűk keresztmetszet. A harmadik lépésnél lehet kiválasztani, hogy legyen-e chip select

bemenet, valamint itt szükséges megadni a bemeneti adatbusz szélességét és a 16 kB-os blokkból használható rész méretét. Egyszerre egy SPI üzenet tartalmát kell tárolni és visszaolvasni így az adatbuszt az írási és olvasási oldalon 8 bit-esre állítottam. Ehhez a beállításához 2048 adatrekesz és ebből következően 11 bit-es címtartomány tartozik. A chip select használata kötelező, mivel csak egy adatbusz lesz a központi modulban, viszont erre több blokk RAM is csatlakozik, így az aktív egység kiválasztására ez a legcélszerűbb módszer.

Legutoljára következett a központi modul fejlesztése. Azért hagytam ezt a végére, mert tudtam, hogy ez lesz a komplexebb. A többi modulban kézenfekvő felépítés volt először a regiszterek definiálása, majd az always blokkok megadása, ahol az órajelre szinkronizált logika van leírva, végül pedig a kimenet értékadása az assign utasítással. Ezt a szemléletet próbáltam először a központi modul megalkotásakor is követni, ám a folyamatos alakítások egyre nehezebben voltak követhetőek. Néhány, az interneten elérhető forráskódú, kész modult tanulmányozva találtam rá egy másik konvencióra. A szorosan összetartozó részek a fent leírt sémát követik, ám a különböző részek egymástól elkülönítve találhatóak, egy rövid kommentmezővel elválasztva. Ez módszer sokkal áttekinthetőbbé tette a Verilog kódot és könnyebb volt követni a módosítások hatásait. Ez alapján a központi modul kódja 5 szekcióra osztható. Az első rész a felelős a kommunikáció időzítéséért. Folyamatosan mintavételezi a BUSY és a WORD_RDY vonalakat, amik jelzik, hogy érkezett-e adat vagy elkészült-e a kiküldés. A második szekció rész végzi a blokkram-ba írást. Ennek a megvalósítása a következő módon történik:

```
always @(posedge clk) begin
    if(word_rdy_reg[0] && ~word_rdy_reg[1]) begin
        data_input <= data_in;
    end
end
assign dina = data_input;

always @(posedge clk) begin
    if(word_rdy_reg[1] && ~word_rdy_reg[0]) begin
        if(frame_rdy[0] == 0) addra_cntr <= addra_cntr + 1;
    end
    if(addra_cntr == (`driver_per_panel * `led_per_driver) )
        addra_cntr <= 0;
    end
end

always @(posedge clk) begin
    if(word_rdy_reg[1] && ~word_rdy_reg[0]) wea_reg <= 1;
    else wea_reg <= 0;
```

```

end
assign wea = wea_reg;

always @(posedge clk) begin
    if(word_rdy_reg[1] && ~word_rdy_reg[0]) ena_reg <= 1;
    else ena_reg <= 0;
end
assign ena = ena_reg;

always @(posedge clk) begin
    if(word_rdy_reg[0] && ~word_rdy_reg[1]) begin
        data_input <= data_in;
    end
end
assign dina = data_input;

```

A harmadik rész a felelős a memóriában tárolt adatok kiolvasásáért. Felépítése nagyban hasonlít az írásnál vázolt kódrészlethez. A negyedik szekció a legrövidebb, a feladata csak annyi, hogy bekapcsolás után megvárja az inicializáció befejeződését. Az ötödik rész végzi a vezérlőjelek (SPI órajel, latch jel, Grayscale órajel) generálását. A legkomplexebb feladat a frame_rdy jel előállítás, amely FIFO_full vonalként is funkcionál. Ez az alábbi módon lett implementálva:

```

always @(posedge clk) begin
    if(init_status_reg == 2'b11) begin
        if (addra_cnr == `driver_per_panel * `led_per_driver )
            frame_rdy[0] <= 1;
        else if (addrb_cnr == (`driver_per_panel * `led_per_driver
            +1) && (busy_negedge[1] && ~busy_negedge[0]))
            frame_rdy[0] <= 0;
        end
        else if(init_status_reg == 2'b01) frame_rdy[0] <= 0;
        else frame_rdy[0] <= 1;

        frame_rdy[1] <= frame_rdy[0];
    end

    assign frame_ready = frame_rdy[0];
end

```

Annak érdekében, hogy a fejlesztés folyamatosan tesztelhető legyen először az adatfogadásért felelős rész készült el. Ezt követte a blokk ram kezelése, elsőként az írás, majd pedig az olvasás. A folyamat végén lett megalkotva a kimeneti része a központi modulnak. A fejlesztés során folyamatosan lett összekapcsolva és együtt tesztelve a központi modul a már korábban megírt modulokkal.

A fejlesztés során megtapasztaltam a FPGA belső működésének időkritikusságát. A modulok összekapcsolásánál jött elő a probléma. Minden modul önállóan le lett tesztelve, de a tesztbemenet előre generált volt nem pedig egy dinamikus jelfolyam. Először csak a központi modul kódját módosítva próbáltam megoldani a

problémát, de a komplexitása miatt egy jel 1-2 órajellel történő eltolása több helyen is hatással volt a modul egészének működésére. Célszerű volt a már megalkotott modulokat rendszer szinten vizsgálni és fokozatosan kijavítani az egyes hibákat.

A hibajavítás közben jött az ötlet az inicializáció modul megalkotásához. A hibátlan működéshez ugyan nem feltétlen szükséges, de erőforrást lehet vele megtakarítani.

Miután a teljes szimuláció az elvárásoknak megfelelően működött következhetett a hardveres tesz. Ennek során egy fejlesztő kit-re, amely ezt a típusú FPGA-t tartalmazta, fel lett töltve a Verilog kódból generált bitstream. Az FPGA kimenetei össze lettek kötve a teszt LED-panellel. A teszt sikeresen zárult.

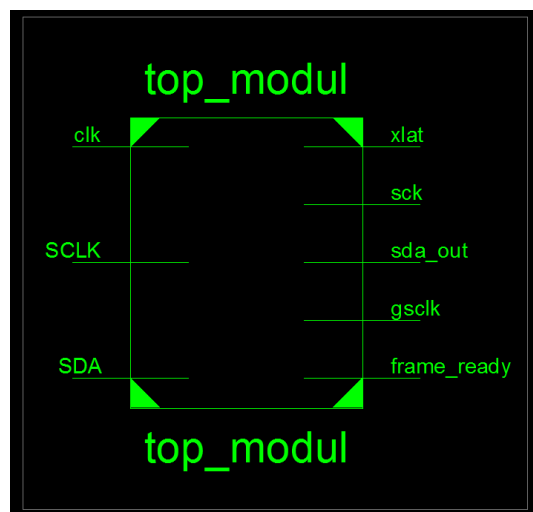
5.4.3 FPGA modulok tervezése

Az FPGA belső felépítése moduláris. Ennek a módszernek az előnye, hogy amint a LED panel vezérlőmodul elkészül tetszőleges számút lehet beilleszteni belőle a tervbe. Az FPGA-n belül 5 féle modul található, melyek az alábbiak:

- inicializációs modul
- SPI_slave modul
- SPI_Master modul
- központi modul
- blokk RAM (Random Access Memory, véletlen elérésű memória)

5.4.3.1 Blokkdiagram tervezése

A FPGA-ban található modul így néz ki a külvilág felől:



22. ábra Az FPGA doboza a külvilág felől

A fent látható képen, a jobb oldalon találhatóak a bemenetek és a bal oldalon a kimenetek. Az SCLK és az SDA vonalakon keresztül jön a képinformáció a mikrovezérlő felől. A CLK nevű bemenet a külső 50 MHz-es órajel, erre fog szinkronizálni az egész rendszer. Fontos megjegyezni, hogy a mikrovezérlő felől érkező SPI Clock nincs szinkronban a rendszer órajelével, így ennek megfelelően kell majd mintavételezni.

Az 5 kimeneti vonalból 4 megy a LED panelek felé, a FRAME_READY pedig a mikrovezérlőnek jelzi, hogy az összes információ továbbítva lett a LED panelek felé és az FPGA készen áll az új adatok fogadására. Az sck a kimeneti SPI órajel, az SDA_OUT a kimeneti SPI adatvonal, a GSCLK a LED driver áramkörök Grayscale órajele, az XLAT pedig a latch jel, amelynek felfutó élére töltődik a shiftregiszterek értéke a belső tárolókba.

5.4.3.2 Központi modul kialakítása

Ez a modul, ami data_intersect néven lett implementálva, az adatok mozgását végzi. Ez a központi modul a legbonyolultabb és a legtöbb kódsort tartalmazó is egyben. Az SPI-slave modultól kapott 8 bit információt eltárolja a blokk memóriában, majd amikor a teljes kép bejött, megkezdí a továbbküldés irányítását a LED-panelek felé. Amint végzett a FRAME_RDY vonalon jelzi a mikrovezérlőnek, hogy készen áll a következő képkocka fogadására.



23. ábra az FPGA központi moduljának logikai diagramja

A DATA_IN bemeneti busz az SPI-on kapott 8 bit-es adatcsomagot tartalmazza, értéke a WORD_RDY vonal felfutó élére érvényes. A kapott adatot ezután a blokkram-ban kell elhelyezni. Ehhez szükség van az ADDRA (address, „a” port), WEA (write enable), ENA (chip enable, „a” port) és a DINA (data input, „a” port) vonalakra. A

modul automatikusan inkrementálja a címet, amelyet az addra vonalon ad ki, majd új képkocka érkezésekor nullázza. A blokkmemória olvasásához az ADDR_B (address, „b” port), az EN_B (chip enable „b” port) kimenet és a DOUT_B (data output, „b” port) bemenet szükséges.

Annak érdekében, hogy az egész rendszer egy ütemben működjön, ez a modul generálja a vezérlő jeleket és az SPI órajelet is az összes LED-panel számára. Ezek a vezérlőjelek a már korábban is említett GSCLK és XLAT.

Amíg a LED meghajtó áramkörök inicializációja nem történt meg, a modul nem fogad adatot és nem is küld tovább. Ez az inicializáció automatikusan lefut minden bekapcsoláskor.

5.4.3.3 SPI_slave modul kialakítása

Ez a modul fogadja az adatokat a mikrovezérlő felől. Ahogy fentebb is említettem a bejövő SPI órajel nincs szinkronban az FPGA belső órajelével és mivel szinkron sorrendi hálózat tervezése a cél, így nem megfelelő. Erre jó megoldás, hogy az SCLK értékét a belső órajel felfutó élére mintavételezem és eltárolom egy kétbites regiszter 0. bitjében és ezzel párhuzamosan a 0. bitet pedig tovább shift-elem az 1. bitbe. Ez a következőképpen van megvalósítva:

```
always @(posedge clk) begin
    posedge_sck_detect[0] <= SCLK;
end
always @(posedge clk) begin
    posedge_sck_detect[1] <= posedge_sck_detect[0];
end
```

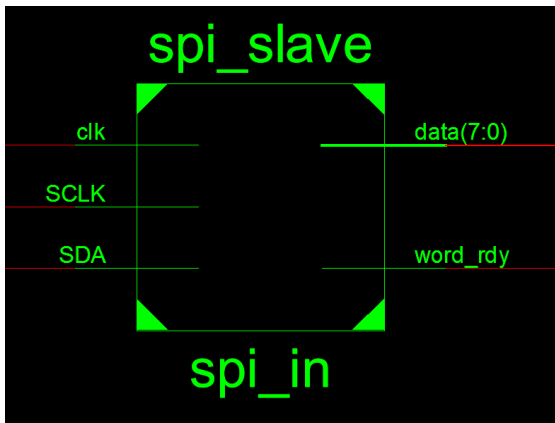
A regiszter értékét már lehet vizsgálni a belső órajel felfutó élére és amint a tartalma 2b’01-el egyenlő az jelzi, hogy felfutó él volt az input vonalon. Hasonló a helyzet lefutó él keresés esetén, ekkora a regiszter tartalma 2b’10. Mindkét eset csak egy órajelig áll fenn, hiszen a következő mintavételre már 2b’11 vagy 2b’00 lesz a regiszter tartalma, így hatékonyan lehet használni élek detektálására. Ez a módszer nem használható, ha az input frekvenciája nagyobb, mint a belső órajel fele. Az én esetemben ez nem fordulhat elő mert az FPGA belső órajele 50 MHz, a SPI-on az adatok folyamatos küldéséhez már 10 MHz-es frekvencia is elegendő. Ezt a funkciót leíró Verilog-kód:

```
always @(posedge clk) begin
    if(posedge_sck_detect[0] && ~posedge_sck_detect[1])
        spi_data[spi_bit_cntr] <= SDA;
end
```

```

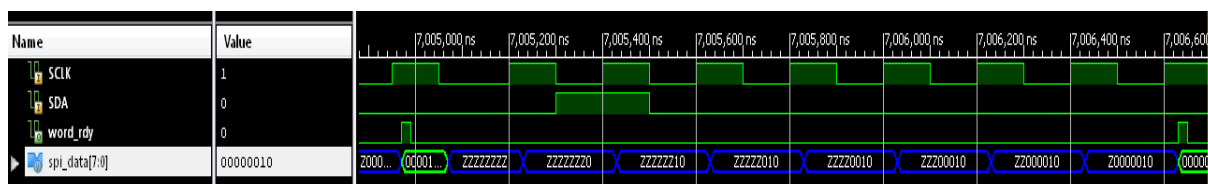
        else      if(posedge_sck_detect[1] && ~posedge_sck_detect[0]      &&
(spi_bit_cntr == 0)) spi_data <= 8'hz;
end

```



24. ábra Az SPI Slave modul logikai diagramja

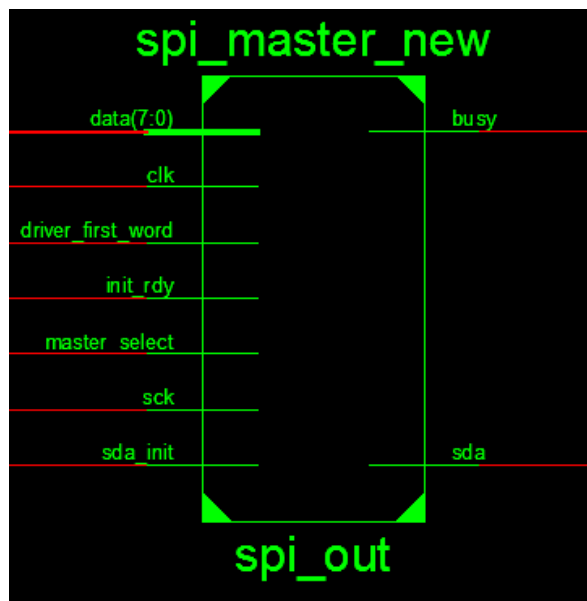
Egyszínű LED-ek esetén ennyire finom hangolási lehetőség nem kell, már a 8 bit is elegendő fényerő skálázhatóságot eredményez. Ekkora az adatok átviteléhez 50 Hz-es képfriásítás esetén 1.92 MHz szükséges. A modul a 8 bit vétele után a WORD_RDY kimeneten jelzi, hogy megérkezett egy csomag és kint van a data buszon. Az adatok további kezeléséről a data_intersect modul gondoskodik. Idődiagramon ez az alábbi módon néz ki:



25. ábra Az SPI-on történő adatfogadás idődiagramja

5.4.3.4 SPI Master modul kialakítása:

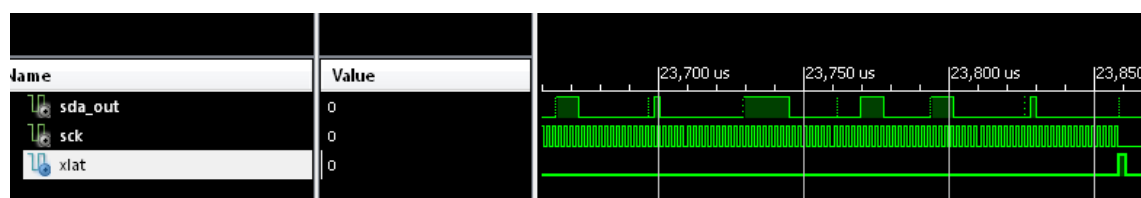
Ez a modul vezérel egy darab LED-panelt. Több LED-panelhez több modult kell egymás mellé elhelyezni és a MASTER_SELECT jellel lehet kiválasztani az éppen aktív SPI_Master modult. A MASTER_SELECT jel felfutó élére automatikusan betöltődik a modul tárolójába a DATA buszon található információ és megkezdődik az adatok soros továbbítása a LED-panel felé. Amíg az aktuálisan tárolt 8 bit átvitele nem fejeződött be a BUSY jel magas értéke jelzi a központi modul felé, hogy még folyamatban van az átvitel.



26. ábra Az SPI Master modul logikai diagramja

Minden LED meghajtó áramkörbe 769 bitet kell továbbítani, ez 48 darab 16 bites csomag és egy vezérlőbit, ami eldönti, hogy konfigurációs vagy fény információ érkezett. Amikor a DRIVER_FIRST_WORD bemenet magas értékű, a kiküldendő adatcsomag elé a modul beszúr még egy 0-t, ez a vezérlőbit, ami az MSB lesz a LED-driver áramkör shiftregiszterében. Ahhoz, hogy a kapott 8 bit adatból 16 bit információ legyen kiküldve, az alsóbb helyiértékek 0-s értékkel vannak feltöltve. Ezzel a módszerrel a teljes 0-32 mA-es tartomány egyenletesen van lefedve. A megoldás előnye, hogy kevesebb adatot kell tárolni és mozgatni és csak minimálisan romlik a fényerő skálázhatósága.

Az SPI kimeneti adatvonala (SDA) a központi modultól kapott SCK órajelhez van szinkronizálva. Az alábbi képen látható egy képkocka kiküldésének vége. Amint az utolsó adatbit is kiküldésre került lekapcsol az SPI órajele és egy rövid latch impulzust küld ki az FPGA.



27. ábra a LED-vezérlő áramköré felé kimenő jelidagram

Bekapcsoláskor ez a modul küldi az inicializáló adatokat, viszont csak az órajelet kapja a központi egységtől, az elküldendő adatot az init_modul szolgáltatja.

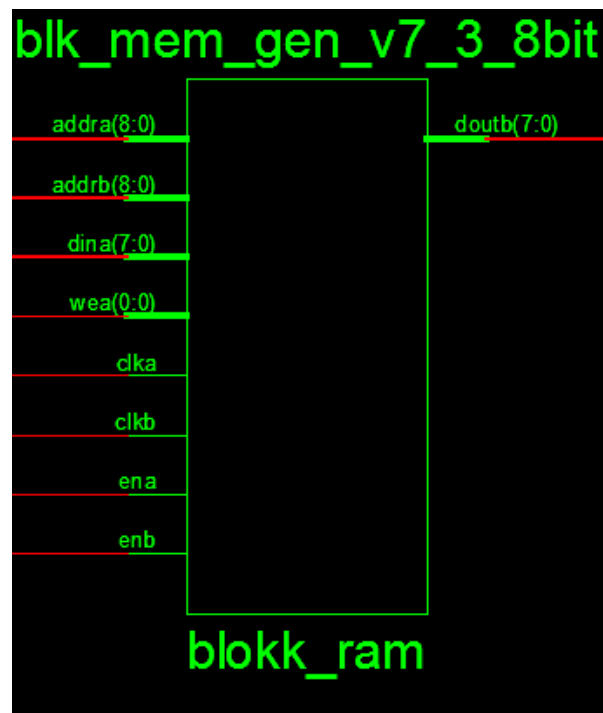
5.4.3.5 Init modul kialakítása

Ez a modul tartalmazza az inicializációhoz szükséges 769-es bitsorozatot. Bekapcsolás után ezek továbbításra kerülnek minden panel minden LED-driver áramköréhez. Ezután a LED-meghajtók képesek a LED-ek vezérlésére. Az adatok küldése a központi modul által generált SK órajelhez van szinkronizálva.

Az inicializáció végeztével az INIT_RDY vonal értéke alacsony lesz és az FPGA képes adatokat fogadni a mikrovezérlőtől, és a kapott adatokat megjeleníteni a LED-paneleken.

5.4.3.6 Blokk ram kialakítása

Ez egy előre legenerált modul, amely a beépített generátorral hozható létre.

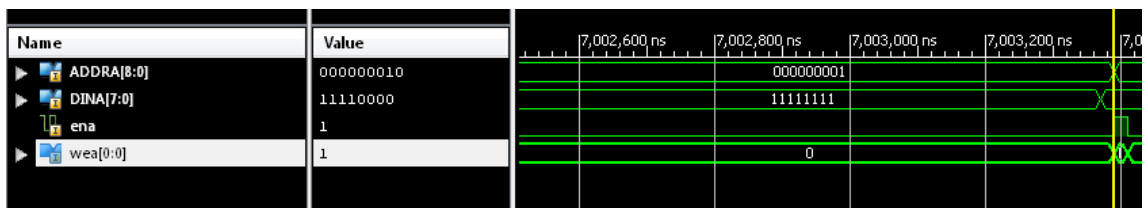


28. ábra a blokk RAM logikai diagramja

Az általam használt memória típus egy dual-port RAM, az „A” port csak írásra, a „B” port pedig csak olvasásra használható. Ez a megoldás lehetőséget ad arra, hogy a memóriát egyidejűleg lehet írni és olvasni is.

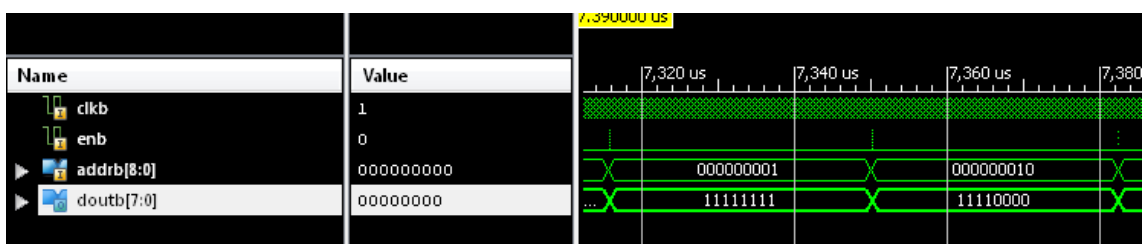
Mivel a két port képes egymástól függetlenül működni rendelkeznek egyedi cím- és engedélyező bemenettel valamint adatbusszal is. Mindkét port az 50 MHz-es rendszer órajelről üzemel. Az egyetlen különbség a két port között, hogy az „A” portnak van egy write enable (WEA) bemenete a chip select (ENA) bemenet mellé. Az írási

szekvencia érvényességéhez mindkét jel használta szükséges. Az írási ciklus idődiagramja a következő ábrán látható:



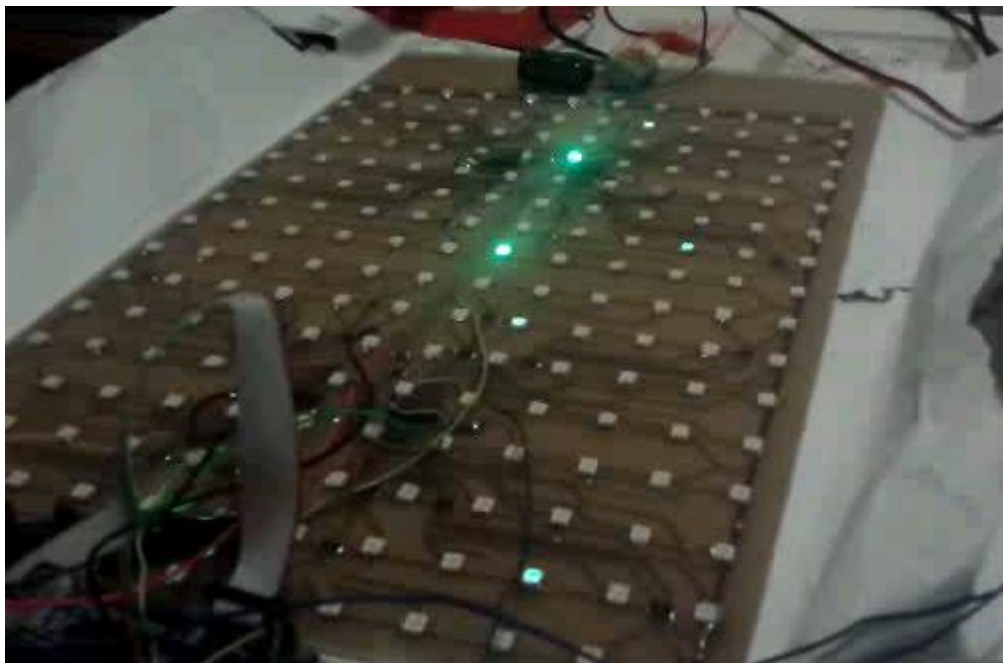
29. ábra a blokk RAM írásának idődiagramja

Kiolvasás esetén a következő jelforma látható:



30. ábra a blokk RAM olvasásának idődiagramja

5.4.4 Panelvezérlés tesztelése



31. ábra Tesztelés

5.5 Kiegészítő elemek tervezése

5.5.1 Tápellátás tervezése

Az előző fejezetekben már volt kalkuláció a fogyasztásról. A LED-ek tápellátása 5 V-ról célszerű. Ezen a feszültségen már teljesen kinyit a zöld dióda, de nem szükséges előtét ellenállás ugyanis a LED-vezérlő áramkörön áramgenerátor kimenetű. A teljes kijelző Worst-case fogyasztása 777 W (~150 A). Ez abban az esetben van, ha minden LED maximális kitöltéssel (fényerővel) világít. Üzemszerűen ez az állapot nem általános, de szükség esetén ezt a terhelést el kell bírni. A fogyasztás csökkentése érdekében lehetne szoftveres áramkorlátozást bevezetni. Ebben az esetben a beágyazott számítógép figyelne a felkapcsolt LED-ek számát és áramát, de a tervezés során a céges konzulenssel ezt a lehetőséget elvetettük.

Lehetőségként felmerült asztali számítógépek tápjának felhasználása. Ezekből rendelkezésre is állt néhány darab. Az áruk mérsékelt és könnyen cserélhetőek. Sajnos hiába van néhány száz W-os teljesítményük az 5 V-os ágakon ennek csak töredékét képesek leadni, így ez nem vezetett gazdaságos megoldáshoz.

Saját tervezésű tápegység is szóba került, de ennek a megtervezése és megépítése több akadályba ütközött, például hogy sok időt vett volna igénybe. Információim ugyan voltak a kapcsolóüzemű tápegységek működéséről és tervezéséről, de ekkora teljesítmény és áramszükséglet esetén egy stabilan működő egység megépítése, amiből mindössze 1 darabra van szükség, nem veheti fel a versenyt gazdaságosság szempontjából a piacon található termékekkel.

Ebben a teljesítmény tartományban AC/DC átalakítók között a legtöbb típus 12- vagy 24 V_{dc}-os kimenettel rendelkezik, én a Mean Well céget találtam, akik 5 V-os szekunder oldalú terméket gyártanak. A figyelembe vett megoldásokat az alábbi táblázat tartalmazza:

Teljesítmény (W)	Ár	Szükséges darabszám	Teljes ár	Teljes ár (HUF)
150	36 EUR	6	216 EUR	~68000
350	52 EUR	3	156 EUR	~49000
600	85 USD	2	170 USD	~48500
1000	200 EUR	1	200 EUR	~62500

A feltüntetett árak tájékoztató jellegűek, a forgalmi adót és a postaköltséget nem tartalmazzák.

A táblázat alapján látható, hogy nem érdemes egy nagy vagy több kicsi tápegységet választani. A legköltséghatékonyabb választás két darab 350 W-os és egy darab 150 W-os táp beszerezése. Ez nagyságrendileg 44000 Ft-ba kerül.

5.5.2 Kábelezés kialakítása

A vezérlő board nem közvetlen a LED-panelek mellett fog elhelyezkedni és a kiterjedését is számításba kell venni az adatátvitel során. Beágyazott rendszer esetén a 2-3 m-es távolságnál a jelátvitel külön figyelmet igényel. A különböző vezetékeken a jelterjedési időnek meg kell egyeznie, mert ennek hiányában különböző fázisban lesznek a képernyő távolabbi részei a közelebbiekhez képest. Ennek érdekében közel ugyanakkora vezetékkel kell bekötni az adatvezetéseket.

Az 5 V-os táp, a föld és a 4 csatorna differenciális átviteléhez 10 eres kábel szükséges. Ehhez kevés a 8 eres UTP kábel így más típusra van szükség. Először az YSLY-JZ 10X0.75 típust vizsgáltam. Ez egy kettős szigetelésű ipari kábel, tipikusan jelző és működtető funkciókat lát el. Az erek számozottak a megfelelő bekötés elősegítéséhez.

A másik típus az ASRA 10X0.22-es kábel. Ez is kettős PVC szigeteléssel rendelkezik. Az erek nem számozottak, de a könnyű megkülönböztetéshez a belső szigetelés erenként különböző színű. Kisebb átmérője miatt könnyebb hajlítani. Jellemzően a biztonságtechnikában használják.

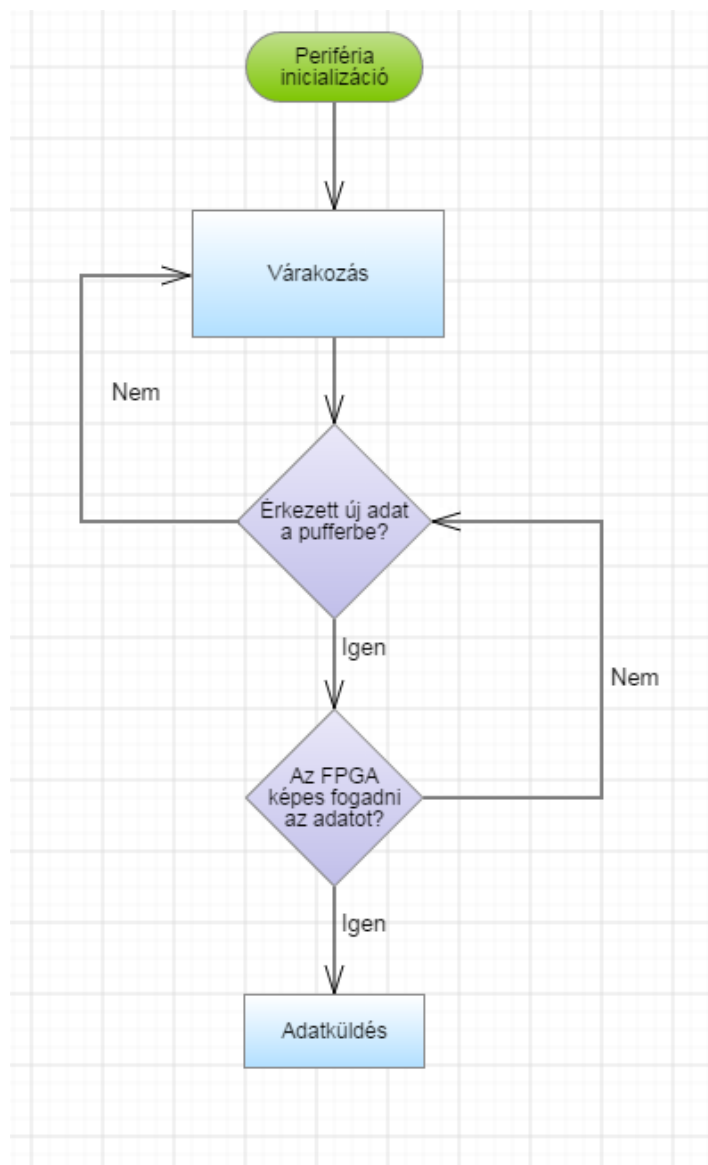
A két típus közül én az ARSA 10X0.22-es típust választottam. Bár az YSLY-JZ 10X0.75 jobb szigeteléssel és nagyobb keresztmetszettel rendelkezik emiatt magasabb az ára. Ebben az alkalmazásban nem indokolja semmi a nagyobb és drágább vezeték alkalmazását.

A kapcsolóüzemű tápegységek 5 V-os tápellátásához megfelelő a háztartásokban is használt 2 eres kábel. Ami fontos, hogy a LED-panelek ne sorban legyenek rákötve a tápegységre, hanem csillagpontosan. Ehhez ugyan több kábel szükséges, de ez a megoldás hibatűrőbb és egyenletesebb feszültséget biztosít minden panel számára.

6 Szoftvertervezés

A Raspberry Pi alapsomag nem tartalmaz SD kártyát, se az operációs rendszert. Az operációs rendszer telepítője elérhető a Raspbian honlapján.^[4] Telepítés után az operációs rendszer még nem tartalmazza a Boardcom gyártmányú SoC (System on Chip) illesztő szoftverét. Ez kötelező telepíteni, mert nélküle nem lehet használni a szükséges perifériákat (GPIO, SPI, stb.) . A telepített csomag tartalmaz néhány példakódot a különböző perifériák használatához.

A Raspberry Pi-on az alábbi állapotgéppel leírható program fut:



32. ábra a Raspberry Pi szoftverének folyamatábrája

Az inicializáció során a program felkonfigurálja az SPI vonalat. Az átvitelhez MODE0-s kommunikációt használok (lásd 5.4.1-es fejezet). A sebesség beállításánál több opció közül lehet választani. A sebesség az órajel 2 hatványával leosztott értéke lehet. 64-es leosztás esetén 3.9 MHz lesz az órajel. Ezzel a sebességgel már megvalósítható a megfelelő kommunikáció.

Az adat puffer feltöltése kívülről történik. A puffer egy 4800 elemes 8 bites tömb. Ehhez tartozik egy státuszváltozó, amely logikai igaz, ha új információ érkezett a pufferbe. Az utolsó adat kiküldése után a státusz értéke logikai hamisra változik.

A szoftvertervezés célja egy felület kialakítása volt, amellyel el lehet érni a kijelzőt. A bemeneti tároló feltöltéséhez rendelkezésre áll egy operációs rendszer, Ethernet csatlakozó és soros port is. Ezek kezelése kívül esik a dolgozat keretén.

7 Összefoglalás

Az általam végzett projekt célja egy egyszínű LED-es mátrixkijelző kijelző tervezése és dokumentálása. Ez a kijelző egy erőmű külső oldalán kerül elhelyezésre.

Bemutatásra került a LED technológia fejlődése, a korszerű LED meghajtó áramkörök és felhasználási területeik.

Specifikáltam a megtervezett LED-mátrix kijelző paramétereit. A rendelkezésre álló hely, pixelszám és költségkeret figyelembevételével döntés született a LED kijelző felbontásáról. A felbontás 60 x 80 pixel lett és ez optimálisan 1.49 x 1.98 m nagyságú kijelzőt eredményezett. A specifikáció szerint a kijelző monokróm, grafikus és karakteres adatokat képes megjeleníteni.

Meghatároztuk a rendszer funkcionális felépítését. Több lehetőség figyelembevételével megterveztük a rendszer főbb alkotórészeit, a vezérlő egységeket és a kommunikációs interfészeket. A végleges beágyazott számítógép egy Raspberry Pi B+ lett, amely Etherneten fogadja a kívülről érkező adatokat és SPI-on keresztül küldi tovább őket a Spartan 6-os FPGA-nak. Egy 297 x 396 mm-es panelen 4 darab LED vezérlő IC található és ezekhez 192 darab zöldszínű LED tartozik. A teljes kijelző 25 darab panelből áll.

Az FPGA belső felépítése moduláris. Ennek a módszernek az előnye, hogy amint a LED panel vezérlőmodul elkészül, tetszőleges számút lehet beilleszteni belőle a tervbe. Öt modult készítettem el: az SPI Master modult, az SPI Slave modult, a központi modult, az inicializáló modult és a blokk RAM-ot. Ezek a modulok valósítják meg az adatátvitelt a beágyazott számítógép és a LED panelek között. Elkészítettem egy szoftvert a Raspberry Pi-ra, amely egy pufferből átküldi az adatokat az FPGA-ba.

Elkészült a LED kijelző tesztmodellje, ami lehetőséget biztosított a hardver és szoftver komponensek működésének tesztelésére. A tesztmodell segítségével sikerült a rendszer működőképességét igazolni. Előzetes költségbecslés alapján ezzel a módszerrel a piacon elérhető LED falaknál olcsóbban lehet előállítani nagyméretű monokróm LED mátrix kijelzőket.

A kijelző tervezésénél fontos volt az elérhető legnagyobb méret, de véges erőforrások miatt megjelenítési képességei korlátosak.

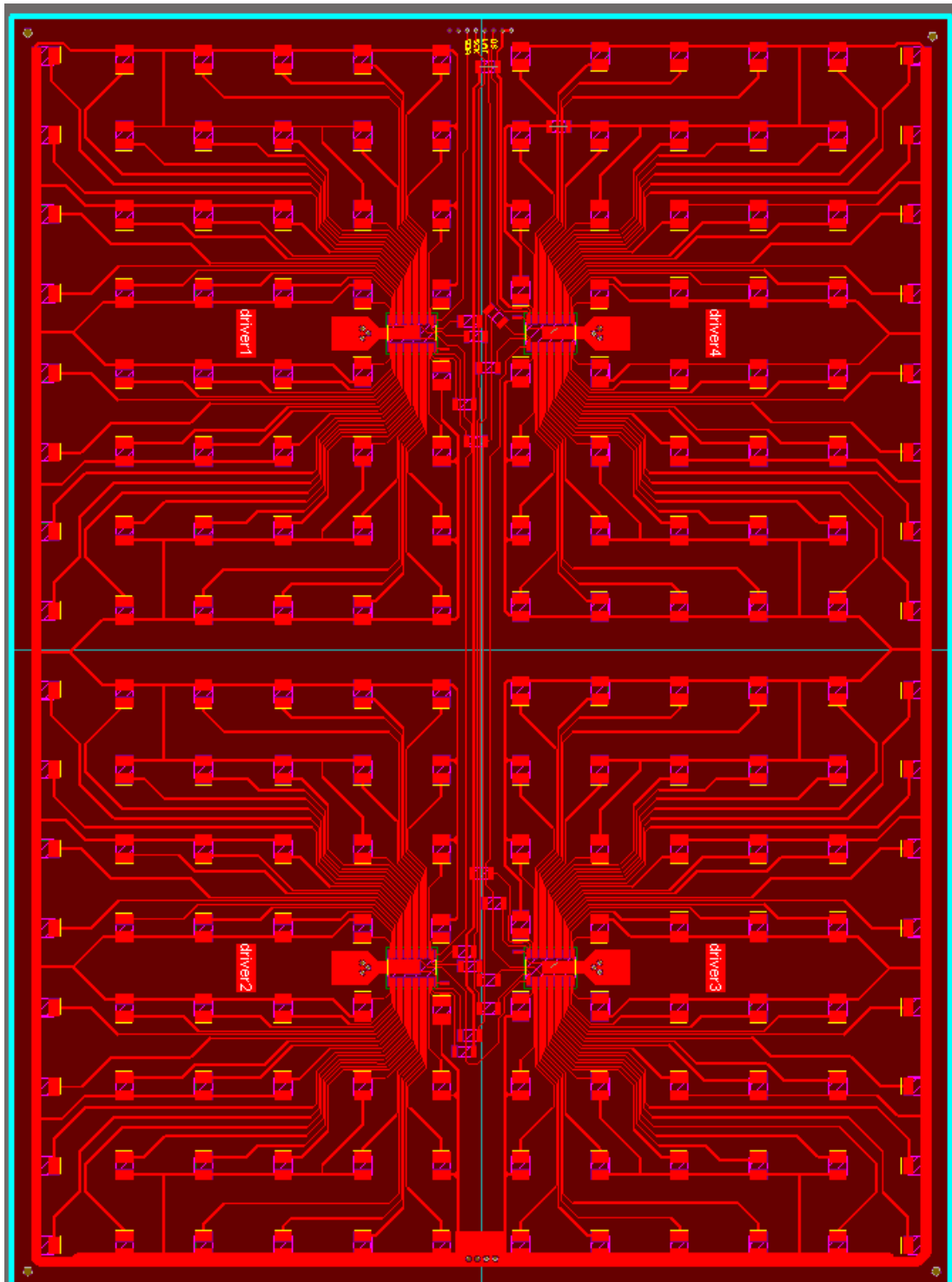
A kijelző megjelenítő képessége továbbfejleszthető színes LED-ek alkalmazásával. Ennek megvalósításához komolyabb hardveres átalakítás szükséges.

A beágyazott számítógép és a hozzá tartozó operációs rendszer lehetőséget biztosít a rendszer szoftveres továbbfejlesztésére. A hardvertervezés során beépítésre került egy Ethernet csatlakozó, aminek segítségével a rendszer bemeneti pontján elhelyezkedő Raspberry Pi interneten elérhetővé tehető. Az operációs rendszerhez tetszőleges megjelenítő szoftver tervezhető a karakterek kiírásától egészen az bonyolult ábrákig.

8 Irodalomjegyzék

- [1] http://www.electronics-tutorials.ws/diode/diode_8.html
- [2] <http://www.ti.com/product/TLC5955/datasheet>
- [3] <http://www.projektor.hu/lumen+lux+kandela.html>
- [4] <https://www.raspberrypi.org/downloads/raspbian/>

9 Függelék



Sor	Oszlop															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	150	149	145	147	183	181	185	186	102	101	97	99	135	133	137	138
2	148	152	161	146	182	173	188	184	100	104	113	98	134	125	140	136
3	163	153	162	157	169	174	189	175	115	105	114	109	121	126	141	127
4	155	151	160	159	171	172	187	191	107	103	112	111	123	124	139	143
5	156	165	164	158	170	176	177	192	108	117	116	110	122	128	129	144
6	154	167	168	166	178	180	179	190	106	119	120	118	130	132	131	142
7	46	35	36	34	22	24	23	10	94	83	84	82	70	72	71	58
8	48	33	32	26	14	20	21	12	96	81	80	74	62	68	69	60
9	47	43	28	27	15	16	7	11	95	91	76	75	63	64	55	59
10	31	45	30	25	13	18	9	19	79	93	78	73	61	66	57	67
11	40	44	29	38	2	17	8	4	88	92	77	86	50	65	56	52
12	42	41	37	39	3	1	5	6	90	89	85	87	51	49	53	54