

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

«Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті»
коммерциялық емес акционерлік қоғамы

Автоматика және ақпараттық технологиялар институты

Электроника, телекоммуникация және ғарыштық технологиялар кафедрасы

Тажен Қалмұхамед Әбубәкірұлы

«БЛИС негізінде көп арналы басқару блогын әзірлеу»

ДИПЛОМДЫҚ ЖҰМЫС

6B07112 - Electronic and Electrical Engineering

Алматы 2024

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ
МИНИСТРЛІГІ

«Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті»
коммерциялық емес акционерлік қоғамы

Автоматика және ақпараттық технологиялар институты

Электроника, телекоммуникация және ғарыштық технологиялар кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ
ЭТ ж ҒТ кафедра меңгерушісі
техн. ғыл. канд.
Е. Таштай
« 31 » 05 2024 ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы «БЛИС негізінде көп арналы басқару блогын әзірлеу»

6B07112 - Electronic and Electrical Engineering

Орындаған:

Тажен Қ.Ә.

Рецензент

Ғылыми жетекші
техн. ғыл. кандидаты,
аға оқытушы

техн. ғыл. канд.,
ҚАЗАҚ ҰЛТТЫҚ АҒРАРЛЫҚ
ЗЕРТТЕУ АКАДЕМИЯСЫ
ҚАЗАҚ ҰЛТТЫҚ ТЕХНИКАЛЫҚ
ЗЕРТТЕУ УНИВЕРСИТЕТІ
ФАКУЛЬТЕТІ
А. Б. Токмолдаев
А. Б. Токмолдаев

М. А. Абдуллаев М.А.

« 29 » 05 2024 ж.

« 24 » 05 2024 ж.

Алматы 2024

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ
МИНИСТРЛІГІ

Қ.И Сәтпаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

Электроника, телекоммуникация және ғарыш технологиялар кафедрасы

6B07112 - Electronic and Electrical Engineering

БЕКІТЕМІН
ЭТ ж ҒТ кафедра меңгерушісі
техн.ғыл.кан
Е.Таштай
«31» 05 2023 ж.



Дипломдық жұмыс орындауға
ТАПСЫРМА

Білім алушы: Тажен Қалмұхамед Әбубәкірұлы

Тақырыбы: «БЛИС негізінде көп арналы басқару блогын әзірлеу».

Университет ректорының «4» желтоқсан 2023 ж. №548 П/Ө бұйрығымен
бекітілген

Аяқталған жобаны тапсыру мерізімі «15» мамыр 2024 ж.

Жұмыстың бастапқы мәліметтері:

1) Цилиндрлік 3D дисплейдің құрамы және негізгі электронды бөлшектері:
дисплей панелі, айналмалы механизм, басқару тақтасы, қоректендіру блогы,
сенсорлар;

2) Негізгі плата ретінде NEXYS A7, Басқару бөлшегінің негізгі интерфейсы
SPI.

3) ПЛИС негізінде басқару бөлшегінің құрамы және программалау
жолдары;

4) Xilinx фирмасының Nexys Artix-7 платасы.

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

а) Цилиндрлік 3D дисплейді толыққанды жұмысқа қосу.

б) Artix-7 платасымен жұмыс жасау.

в) Айналымдарды есептеу, Циклдарды есептеу.

г) ПЛИС программалау;

д) Схема түріндегі 4-биттік сумматор сызбасын құру.

Сызбалық материалдар 15 слайдпен ppt форматында көрсетілген.


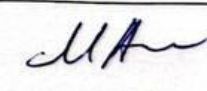
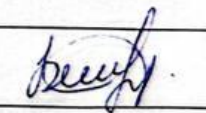
Пайдаланылган әдебиеттер тізімі:

- 1 Обзор шины SPI и разработка драйвера ведомого SPI устройства для embedded (<https://habr.com/ru/articles/123145/>) 2011 ж.
- 2 Поколения мобильной телефонии (<https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BA%D0%>)
- 3 Что такое «поколение» сетей сотовой связи? (<http://1234g.ru/1g/chto-takoe-pokolenie-setej-sotovoj-svyazi>)
- 4 Многоканальная система измерения положения на базе FPGA-PLC (<https://www.sciencedirect.com/science/article/abs/pii/S0019057821000148>) 2021 ж.
- 5 Многоканальная система сбора данных на базе FPGA для детекторов перегретой эмульсии (<https://www.sciencedirect.com/science/article/abs/pii/S0168900221004423>) 2021 ж.
- 6 Многоканальная система измерения положения на базе FPGA-PLC (<https://www.sciencedirect.com/science/article/abs/pii/S0019057821000148>) 2021 ж.
- 7 Язык описания аппаратуры Verilog HDL (<https://marsohod.org/verilog>)
- 8 Xilinx (<https://ru.wikipedia.org/wiki/Xilinx>)
- 9 Увеличение количества ШИМ контактов Arduino с помощью TLC5940 (<https://microkontroller.ru/arduino-projects/uvelichenie-kolichestva-shim-kontaktov-arduino-s-pomoshhyu-tlc5940/>) 2022 ж.
- 10 Разработка интерфейса SPI на ПЛИС (https://fpga-systems.ru/publ/raznoe/interfaces/spi_chast_1_obshhie_svedeniya/26-1-0-90) 2019 ж.

Дипломдық жұмысты дайындау
КЕСТЕСІ


Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекшіге және кеңесшілерге көрсету мерізімі	Күтілінетін нәтижелер
Теориялық бөлім	1.02.2024 - 21.02.2024	Орындалды
Негізгі бөлім	21.02.2024 - 01.03.2024	Орындалды
Есептеу бөлімі	01.03.2024 - 14.05.2024	Орындалды

Дипломдық жұмыс (жоба) бөлімдерінің кеңесшілері мен норма бақылаушының аяқталған жұмысқа(жобаға) қойған қолтаңбалары

Бөлімдер атауы	Кеңесшілер (аты, әкесінің аты, тегі, ғылыми дәрежесі, атағы)	Қол қойылған күні	Қолы
Диплом жұмысының тақырыбын талдау	Техн.ғыл.канд., ЭТЖҒТ каф. Аға оқытушы Абдуллаев М.А.	30.05.24	
Теориялық ақпарат	Техн.ғыл.канд., ЭТЖҒТ каф. Аға оқытушы Абдуллаев М.А.	30.05.24	
Норма бақылау	ЭТЖҒТ каф. ассистенті Ақылжан П.	30.05.24	

Ғылыми жетекшісі  М.А.Абдуллаев

(колы)

Тапсырманы орындауға алған білім алушы  Қ.Ә.Тажен

(колы)

Күні « 30 » 05 2024 ж.

АҢДАТПА

Тапсырма 240 та 320 пиксель кадрды шығару үшін 240 светодиодтан тұратын линейка керек. Үш топ жеті линейка. Барлығы 21 линейка. Кадр 24 секундта ауысып тұруы керек. Бір бағанадан екінші бағанаға ауысу үшін 130 микросекунд беріледі. 21 SPI интерфейсі БЛИС танысып параллелді шығуы керек. Әр қайсысының жылдамдығы 30 МГц. Сонда жалпы жылдамдық 630 МГц, жуықтап гигагерц немесе гигабит сек. Ондай микросхема жоқ. Сондықтан БЛИС қолданамыз.

АННОТАЦИЯ

Задача 240 та нужна линейка из 240 светодиодов для получения кадра 320 пикселей. Три группы по семь линий. Всего 21 линия. Кадр должен переключаться через 24 секунды. Для перехода от одного столбца к другому дается 130 микросекунд. 21 интерфейс SPI должен выходить параллельно с ПЛИС - интерфейсом. Каждый имеет скорость 30 МГц. Тогда общая скорость 630 МГц, почти гигагерц или гигабит сек. Такого чипа нет. Поэтому используем ПЛИС.

ANNOTATION

Task 240 that needs a ruler of 240 LEDs to get a frame of 320 pixels. Three groups of seven lines. There are 21 lines in total. The frame should switch after 24 seconds. It takes 130 microseconds to move from one column to another. 21 The SPI interface should exit in parallel with the Bliss interface. Each has a speed of 30 Mhz. Then the total speed is 630 Mhz, almost gigahertz or gigabit sec. There is no such chip. Therefore, we use FPGAs.

МАЗМҰНЫ

Кіріспе	8
1 Цилиндрлік 3D дисплейдің құрамы	9
1.1 ЦД сәулелі диодтардың жинағы	9
1.2 ЦД сәулелі диодтарды басқару алгоритмдері	12
2 БЛИС негізінде басқару бөлшегінің құрамы	21
2.1 БЛИС бағдарламалау жолдары	23
2.2 Тиімділік пен өнімділікті арттыруға арналған Vivado	25
3 Басқару бөлшегінің негізгі интерфейсы	28
3.1 Басқару бөлшегінің құрамы	30
4 БЛИС негізгі сұлбалары және бағдарламалау	33
4.1 Негізгі SPI логикасы	34
4.2 Басқару бөлшегін бағдарламалау	35
Қорытынды	37
Пайдаланылған әдебиеттер тізімі	38

КІРІСПЕ

Осы дипломдық жұмыста менің негізгі міндетім FPGA (БЛИС - бағдарламаланатын логикалық интегралды сұлба) толық танысып, LED шамдары бар цилиндрлық дисплей үшін көп арналы басқару блогын әзірлеумен айналыстым. Негізгі плата ретінде NEXYS A7 алынды. Бұл заманауи стандарттарға, характеристика жағынанда ең қолайлы плата. Бағдарламау үшін бірнеше нұсқалары бар: SystemVerilog немесе VHDL. Менің тандауым SystemVerilog тіліне түсті. БЛИС негізінде көп арналы басқару блогын әзірлеу тақырыбының ерекшелігі – бұл заманауи технологиялардың қолданылу аясының кеңеюімен және өндірістік процестердің автоматтандырылуымен тығыз байланысты.

Цилиндрлік 3D дисплейдің құрамы және негізгі электронды бөлшектері.

- 1) Цилиндрлік дисплейдің негізгі бөлшектері;
- 2) ЦД сәулелі диодтардың жинағы;
- 3) ЦД сәулелі диодтарды басқару алгоритмдері;
- 4) БЛИС негізінде басқару бөлшегінің құрамы және программалау жолдары;
- 5) Басқару бөлшегінің негізгі интерфейсы SPI;
- 6) Басқару бөлшегінің құрамы;
- 7) БЛИС программалау жолдары;
- 8) Басқару жүйесінің негізгі алгоритмдары және іске асыру үрдістері;
- 9) БЛИС негізгі сұлбалары және программалары;
- 10) Басқару алгоритмдері;
- 11) БЛИС негізінде басқару бөлшегін программалау.

1 Цилиндрлік 3D дисплейдің құрамы және негізгі электронды бөлшектері: дисплей панелі, айналмалы механизм, басқару тақтасы, қоректендіру блогы, сенсорлар

Цилиндрлік 3D дисплей – бұл үшөлшемді кескіндерді шынайы уақыт режимінде көрсете алатын, цилиндр пішінді дисплей. Мұндай дисплейлер көбінесе интерактивті медиа, жарнама және ойын-сауық салаларында қолданылады. Бұл дисплейлердің негізгі артықшылығы - олар көп бұрыштан көрінетін үшөлшемді кескіндерді жасай алады, бұл олардың қолдану аясын кеңейтеді.

Цилиндрлік дисплейдің негізгі бөлшектері. Цилиндрлік дисплейдің негізгі бөлшектері жалпы құрылғының функционалдылығын қамтамасыз етуге және басқаруға көмектеседі. Мұндай дисплейлер үшөлшемді көріністерді жасауға және көрсетуге мүмкіндік береді, бұл көптеген пайдаланушыларға бір мезгілде бірдей визуалды тәжірибені көруге мүмкіндік береді.

Негізгі бөлшектер:

1. Дисплей панелі: Цилиндрлік дисплейдің орталық элементі болып табылады. Біз LED технологиясың пайдаландық. Панель жан жақтан көрген кезде 360 градустық сурет немесе бейнені көруге мүмкіндік береді.

2. Айналмалы механизм: Бұл дисплейдің бірегей сипатын қамтамасыз ететін маңызды жүйе. Айналмалы механизм дисплейді үнемі немесе белгілі бір сценарий бойынша айналдыруға мүмкіндік береді.

3. Басқару тақтасы: TLC5940 және 74НС595 микропроцессорлары мен басқа электрондық құрылғылардан тұрады, олар дисплейдің барлық аспектілерін басқарады, соның ішінде кескін өңдеу, дисплейдің айналу жылдамдығы және т.б.

4. Қоректендіру блогы: Жоғары қуатты қажет ететін дисплей жүйелері үшін қуатты тұрақты және сенімді етіп беру үшін жауапты. Негізгі қорек көзі ретінде гироскутердың моторы.

5. Сенсорлар: Кейбір жағдайларда, дисплейлер қозғалыс сенсорлары немесе тағы басқа сенсорлармен жабдықталуы мүмкін, бұл олардың интерактивтілігін арттырады.

Бұл бөлшектердің бәрі бірлесіп, цилиндрлік дисплейдің функционалдылығын және жоғары деңгейдегі пайдаланушы тәжірибесін қамтамасыз етеді.

1.1 ЦД сәулелі диодтардың жинағы

Светодиодтар, немесе LED (Light Emitting Diode), электр энергиясын жарыққа айналдыратын жартылай өткізгіштер болып табылады. Оларды түрлі жарықтандыру құрылғыларында қолданады. Светодиодтар келесі компоненттерден тұрады:

1. Светодиод (LED Chip). Бұл - өте кішкентай кристалл, ток өткенде жарық шығарады.

2. Корпус (Housing): Светодиод кристалын қоршап, қорғап тұратын сыртқы қабық. Ол жарықты бағыттау және тарату үшін жасалған.

3. Оптика (Optics): Светодиодтың жарығын бағыттайтын немесе таратуға арналған линзалар немесе басқа оптикалық элементтер.

4. Металл табақша (Metal Plate): Светодиодтың шығаратын жылуын таратып, оны қызып кетуден сақтайтын элемент.

5. Байланыс сымдары (Wiring): Светодиодты электр тізбегіне қосуға арналған сымдар.

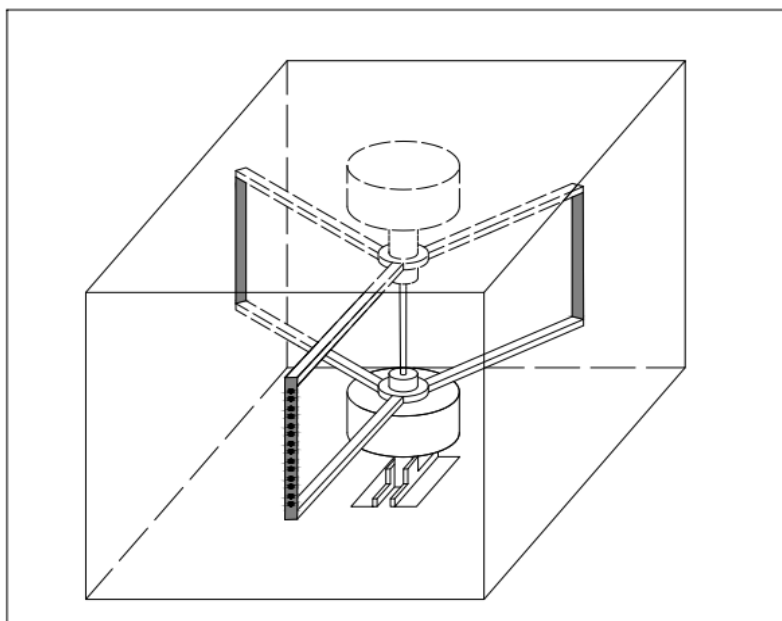
Светодиодтардың қолданылуы. Светодиодтар күнделікті өмірде әртүрлі салаларда кеңінен қолданылады, мысалы:

- Үй және көше жарықтандыруы;
- Дисплейлер мен теледидарлар;
- Автомобиль фаралары;
- Сигналдық шамдар.

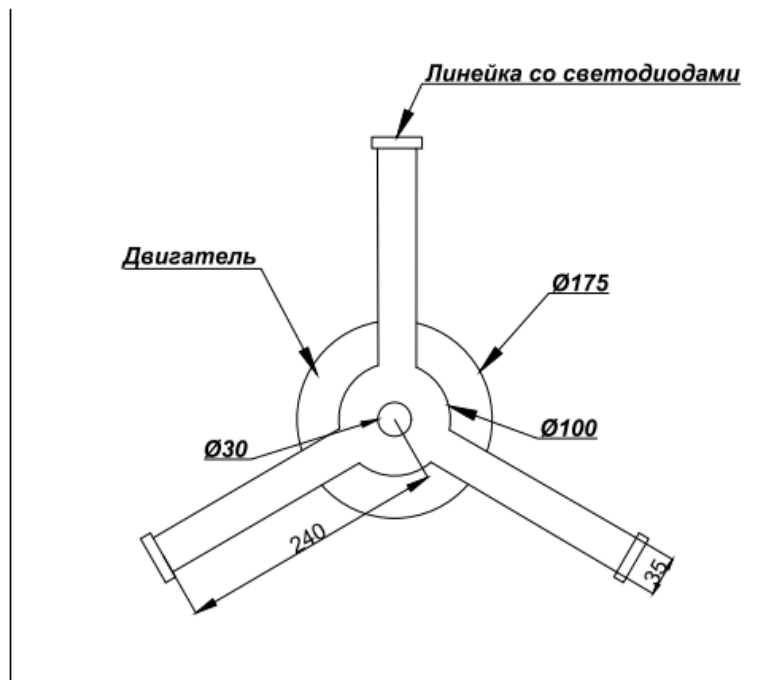
Светодиодтарды құрастыруда нені ескеру керек?

1. Электр қуаты: Светодиодтарға дұрыс кернеу мен ток беру керек. Қате қосылса, олар жұмыс істемей қалады.

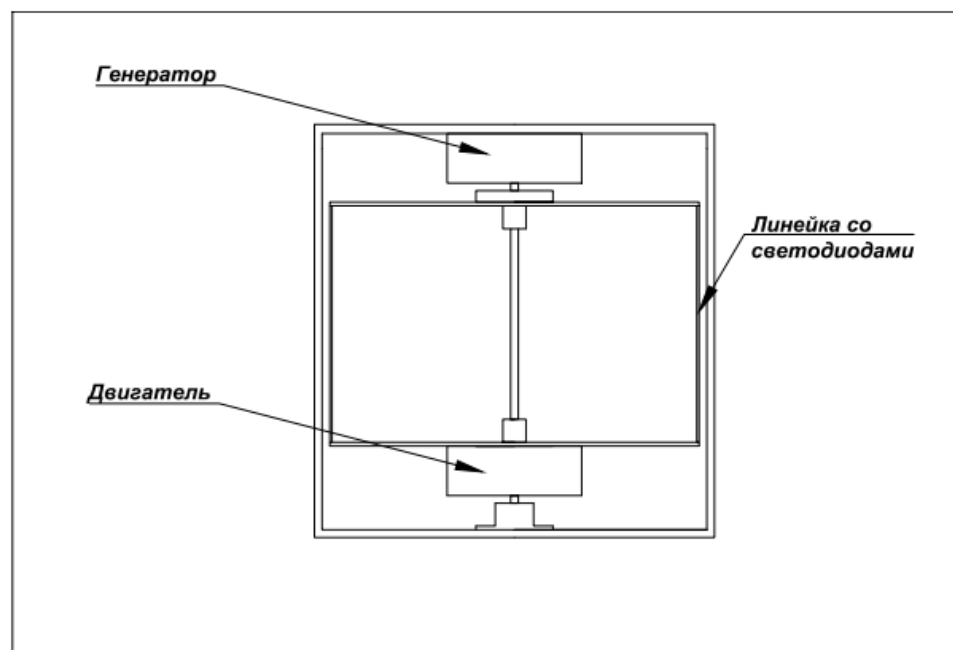
2. Салқындату: Светодиодтарды қызып кетуден қорғау үшін жақсы салқындату керек. Ол үшін металл табақшалар пайдаланылады.



1.1 - сурет – AutoCAD-та жасалған цилиндрлік интерфейс



1.2 - сурет – Цилиндрлік дисплей. Жоғарыдан көрініс



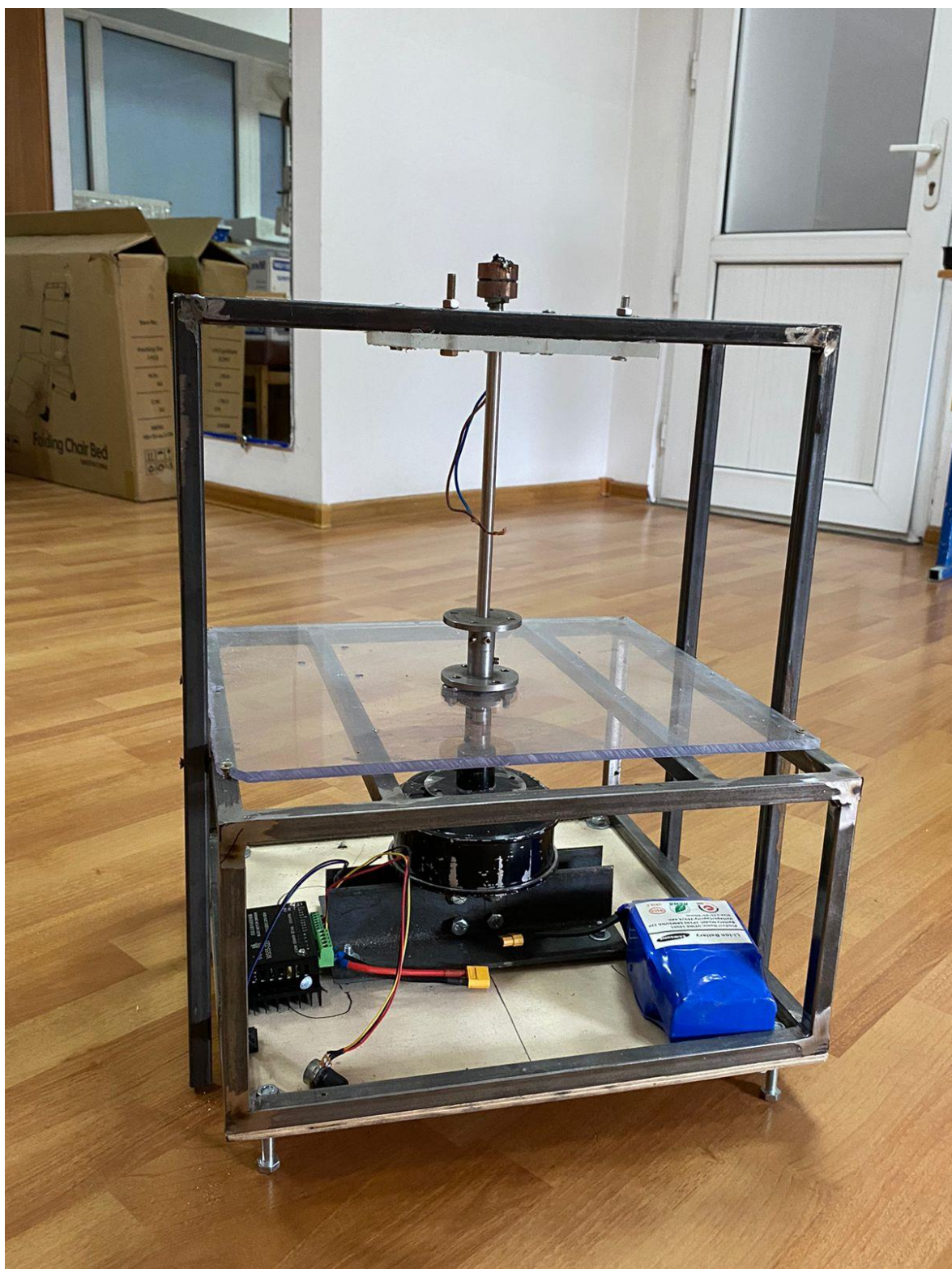
1.3 - сурет – Цилиндрлік дисплей. Бүйірлік көрініс

1.2 ЦД сәулелі диодтарды басқару алгоритмдері

Светодиодтарды (LED) басқарудың бірнеше негізгі алгоритмдері бар. Ең қарапайым әдіс - светодиодты жай қосып-өшіру, яғни электр тогын беру немесе алу арқылы оны жандыру немесе сөндіру. Жарықтықты басқару үшін PWM (импульсті-ендік модуляция) әдісі қолданылады, мұнда токтың қосылып-өшіп тұруының ұзақтығын өзгерту арқылы жарықтың интенсивтілігі реттеледі. PWM әдісі арқылы светодиодтың қаншалықты ұзақ қосулы болатынын өзгерте отырып, оның жарықтығын арттырып немесе азайтуға болады. RGB светодиодтарды басқару үшін әр түсті (қызыл, жасыл, көк) жеке-жеке PWM арқылы реттеп, түрлі түсті жарық эффектілерін жасауға мүмкіндік бар. Осылайша, светодиодтарды басқарудың негізгі алгоритмдері оларды қосу/өшіру, жарықтықты PWM арқылы реттеу және RGB светодиодтардың түстерін басқару болып табылады. Эксперименттік 48-ден 384-ке дейін. Әрқайсысында 7 сызғыштан тұратын үш жапырақ. Барлығы 21 сызғыш, 1008 жарықдиодты және 188 Чип. Ардуино нано 3. Ардуино уно 1.

Светодиодтарды (LED) басқарудың негізгі әдістері бар. Ең қарапайым әдіс - светодиодты қосу немесе өшіру, яғни электр тогын беру арқылы оны жандыру немесе токты алып тастау арқылы сөндіру. Бұл әдіс тек светодиодтың жанып немесе сөнуін қамтамасыз етеді.

Күрделі әдіс - PWM (импульсті-ендік модуляция) арқылы жарықтықты басқару. Мұнда токтың қосылып-өшіп тұруының жылдамдығын өзгерту арқылы светодиодтың жарықтығын реттейді. PWM көмегімен светодиодтың жарықтық деңгейін қажетті мөлшерде арттыруға немесе азайтуға болады. RGB светодиодтарды басқару үшін әр түсті (қызыл, жасыл, көк) жеке-жеке PWM арқылы реттеп, түрлі түсті жарық эффектілерін жасауға мүмкіндік бар.



1.4 - сурет – Проектың қаңылтыры

Электромеханикалық бөлігі. Гироскоптың қозғалтқышы алғаш рет эксперименттік 3D бейнелеу үшін қолданылады. Реттеу айнымалы резистор немесе қосқыш арқылы жүзеге асырылады. Үш жылдамдық. 1 айналу минутына 360, 480 және 720. Сенсор арқылы дисплей айналу жылдамдығын көрсетеді. Бүкіл құрылым 50 см жақтары бар профиль құбырының текшесінің ішіне орналастырылған. Қуат 36 вольтты аккумулятордан алынады. Қуат көзін негізгі қуат көзінен алуға болады.

Гироскоптың қозғалтқышы және 3D бейнелеу. Гироскоптың қозғалтқышы алғаш рет эксперименттік 3D бейнелеу үшін қолданылды. Қозғалысты реттеу айнымалы резистор немесе қосқыш арқылы жүзеге асады. Құрылғы үш түрлі жылдамдықта жұмыс істей алады: 360, 480 және 720 айн/мин. Дисплейде айналу жылдамдығын көрсету үшін сенсор қолданылады. Бүкіл құрылым 50 см жақтары бар профиль құбырының текшесіне орналастырылған. Құрылғы 36 вольтты аккумулятордан қуат алады, бірақ оны негізгі қуат көзінен де қамтамасыз етуге болады. Қуат көзін микроконтроллерден, LED драйвер чіпінен және жарық диодтардан тұратын айналмалы платформаға беру қажет.

Құрылғының дұрыс жұмыс істеуі үшін бірнеше әдіс қарастырылды. Бірінші әдіс - контактілі сақина тартқышты қолдану. Екінші әдіс - сымсыз қуат беру, мұнда екі катушка қолданылады, олардың біреуі тісті доңғалақтың көмегімен айналады.

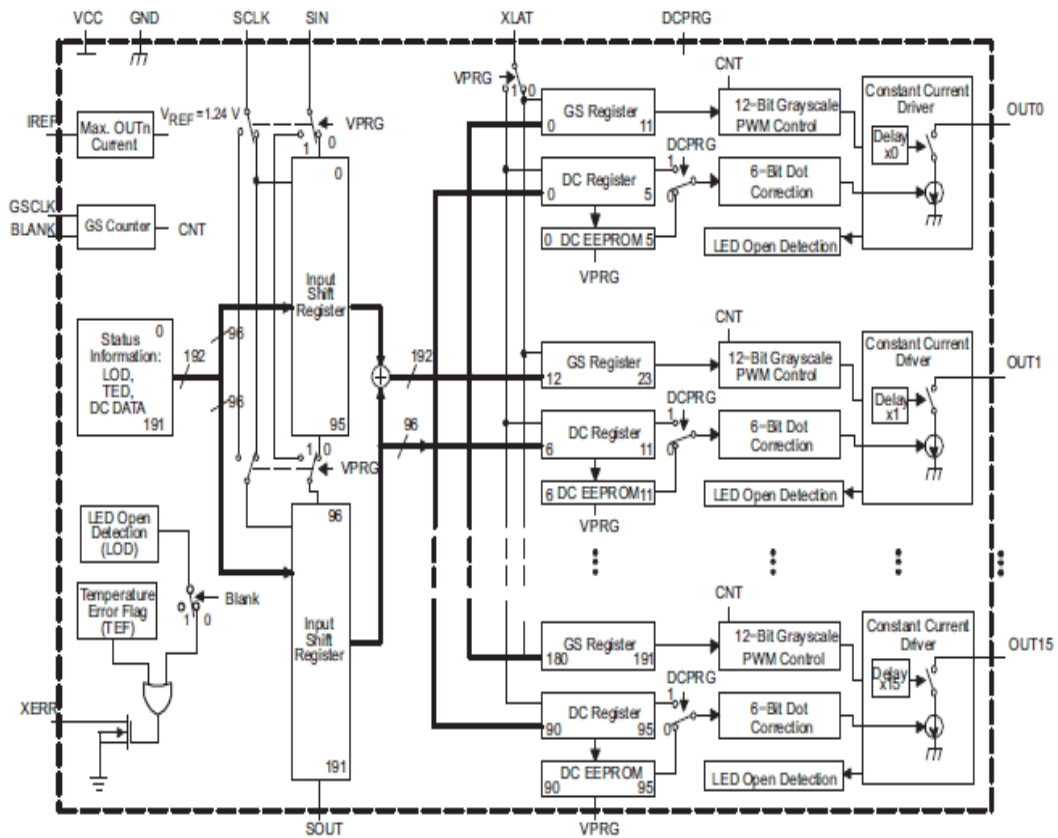
Электроника бөлімі. TLC5940 негізгі диаграммасы келесідей: әрбір TLC5940 чіпі 16 арналы жарық диодтарды басқару үшін 12 биттік регистрлік PWM қолданады. Маған 3 монохромды жарық диоды бар 48 арна қажет, бұл 48 арна үшін 9 чіп керек. Чіптер SPI интерфейсі арқылы тізбектей жалғанып, Arduino Nano арқылы басқарылады. Деректерді беру және тұтану параллель жүзеге асырылады.

Цикл уақытын есептеу. Монохромды жарық диодтар үшін 1 цикл уақытын есептеу қажет. Монохромды жарық диодтар үшін 130 микросекундта 48-ден 384 кадрға дейін өңделеді. 3 сектордың ұзақтығы бір бағанға $384 \times 41,67 \times 3 = 125$ миллисекундты құрайды. Цикл уақыты $125: 384 = 326$ микросекундты құрайды.

Қара және ақ жарық диодтар. Бір жапырақта 336 жарық диод бар. 130 микросекундта $336 \times 12 = 4032$ бит беруіміз керек. Бұл микросекундқа 32 бит, яғни секундына 32 мегабит. Бұл өте жоғары жылдамдық болғандықтан, кадр жиілігін азайту керек. 4032 битті беру үшін 30 кадр жиілігіне төмендету қажет.

Анализ TLC5940. TLC5940-16 арналы тұрақты ток диодты драйвер. Әрбір арна 4096 сатылы PWM сұр жарықтық реттегішімен және 64 сатылы тұрақты ток реттегішімен жабдықталған. Бұл жарықтықтың нүктелік түзетуі бар екенін білдіреді. Енді оның блок схемасын қарастырған жөн:

Block Diagram

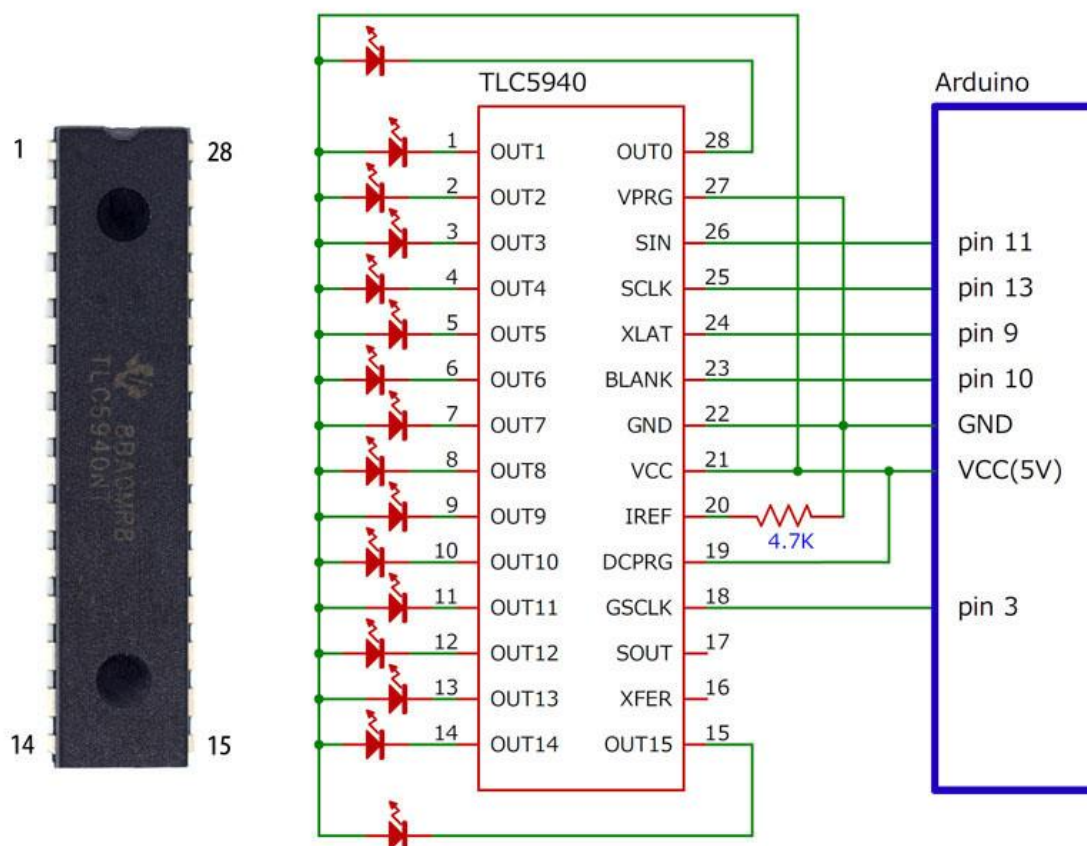


1.5 - сурет – TLC5940 ішкі құрылысы

TLC5940 Модулінің жұмыс принципі. TLC5940 модулі келесі принцип бойынша жұмыс істейді: VCC және DCPRG пиндері арқылы қуат беріледі, ал GND және VPRG пиндері жерге қосылады. SI, SCLK, BLANK, XLAT кірістеріне SPI сигналдары беріледі. CLK кірісіне арнаның PWM сигналы қосылады. Out пиндері арқылы жарық диодтарға шығу сигналдары беріледі. «TLC5940 Datasheet» деп ғаламтордан іздесеңіз, осы микросхема туралы толық ақпаратты табуға болады.

TLC5940NT чипі - бұл 16 арналы PWM (пульсті-ендік модуляция) 12 биттік жарықтықты (0-4096 деңгейінде) және 6 биттік (0-63 деңгейінде) токты реттейтін драйвер. Бұл чип жарық диодтарды, сервоприводтарды және PWM сигналын қажет ететін басқа құрылғыларды басқаруға арналған. Чип деректерді дәйекті түрде қабылдап, 120 мА/арнаға дейінгі токпен 16 арнаға параллель PWM сигналын береді.

TLC5940 модулі - бұл жарық диодтарын, сервоприводтарды және PWM сигналын қажет ететін басқа құрылғыларды басқаруға арналған кеңейту тақтасы. Модуль SPI интерфейсі арқылы басқарылады және әр арнаға 120 мА дейін ток беретін 16 PWM шығысы бар.



1.6-сурет – TLC5940 модулінің Arduino платасына қосылуы схема түрі

TLC5940 Модулінің Arduino платасына қосылу схемасы. TLC5940 - 16-каналды PWM (pulse-width modulation) контроллері, ол әр каналға тәуелсіз жарықтықты реттеуге мүмкіндік береді. Arduino платасымен қолданғанда, оның бірнеше кіріс және шығыс байланыстары бар. Міне, TLC5940-ді Arduino платасына қалай қосуға болатынын сипаттайтын схема:

TLC5940 Модулінің пиндері:

- VCC: Қуат көзі (5V);
- GND: Жер (Ground);
- SIN: Сериялық деректердің кірісі;
- SOUT: Сериялық деректердің шығысы;
- SCLK: Сериялық сағат сигналының кірісі;
- BLANK: PWM шығысын ажырататын сигнал;
- XLAT: Жүктеу сигналы (latch signal);
- GSCLK: PWM сағат сигналының кірісі;
- DCPRG: Dot correction программалау кірісі;
- VPRG: Жүктеу режимі кірісі.

Қосымша ақпарат:

- Шығыс пиндері (OUT0 - OUT15): Бұл пиндерге LED немесе басқа жүктемелерді қосуға болады. Әр пин жеке-жеке жарықтықты реттей алады.

- Қуат резисторлары: TLC5940 чипінің IREF пиніне (IREF) қосылған резистор LED токтарын реттеу үшін қолданылады.

Мысалы LED қосылу:

- OUT0 - Бірінші LED аноды;
- OUT1 - Екінші LED аноды;
- GND - Барлық LED катодтары.

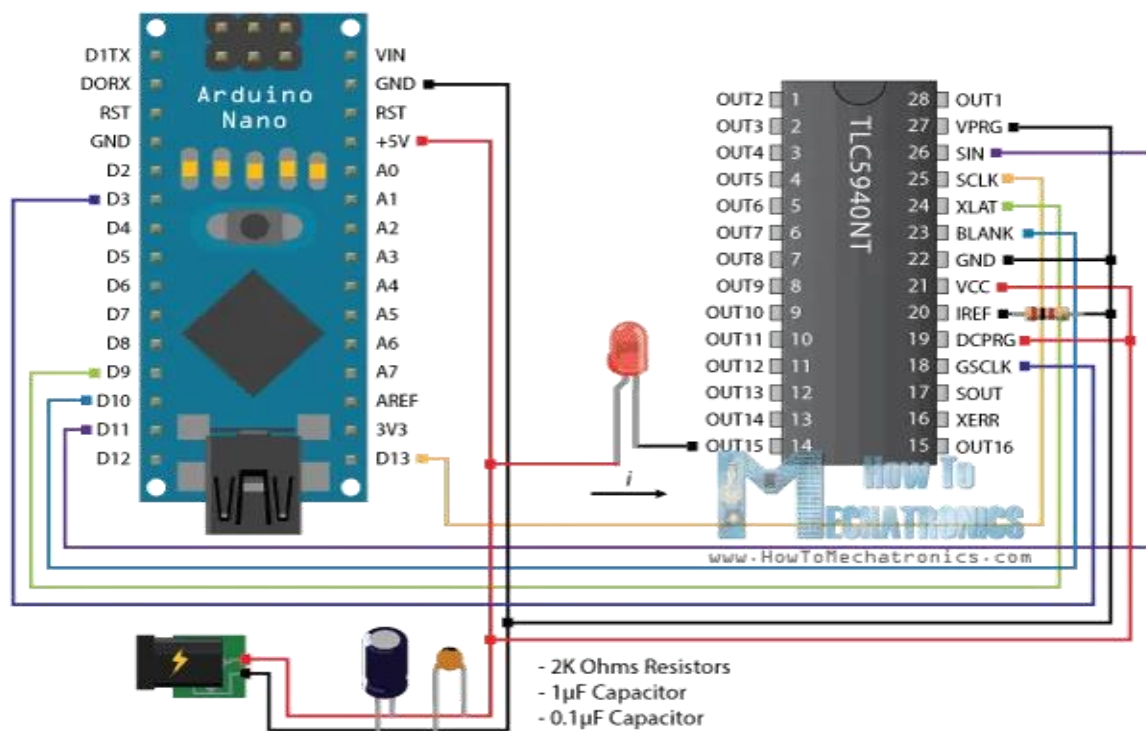
Arduino-дағы арнайы кітапхананы (мысалы, «TLC5940») пайдалана отырып, программалау арқылы әр LED жарықтығын реттеуге болады.

Бұл қосылымдар мен схемалар TLC5940 модулін тиімді пайдаланып, Arduino платасымен интеграциялауға мүмкіндік береді. DCPRG-арна тоғын түзету деректерімен жұмыс режимін орнатады (0-EEPROM, 1-DC Register)

TLC5940 жарықдиодты драйверінің жұмыс принципі мақсатына байланысты өзгереді. Ол, құрылымдағы кіші күйдегі жарықдиодтарды жеткізуді дайындайды. Бұл драйвердің негізгі құрылымы тұрақты ішкі жарықдиодтар мен бір немесе одан көп TLC5940 сипаттамалы жарықдиодтардың жалпы анод схемасын қолдануына мүмкіндік береді.

Бірінші этапта, деректер Input Shift Register ауысында түрлі PWM мәндерін немесе арна тоғын түзетуді (Dot Correction) жазу мақсатында еніп, олар XLAT сигналы мен vprg күйі арқылы PWM (GS Register) немесе арна тоғын түзету (DC Register) регистрлеріне жазылады.

Содан кейін, PWM контроллері жұмыс істейді. Ол, шамдармен жалпы анод схемасы бойынша жарықдиодтардың қатынасын бөліп, осы жарықдиодтарға барлық қажетті PWM сигналдарын жібереді. Сондықтан, TLC5940 жарықдиодты драйвері әр түрлі түстердің жарықдиодтарын қолдануға мүмкіндік береді және күрделі дайындай алады.



1.7 - сурет – TLC5940 модулінің Arduino платасына қосылуы



1.8 - сурет – Монохромды CMD плата

TLC5940 чипін басқару үшін бізге 4 Arduino тақтасының түйреуіштері қажет. TLC5940 басқару үшін біз Alex Leone әзірлеген арнайы кітапхананы қолданатындықтан (оны жүктеу сілтемесі төменде келтірілген), біз TLC5940 чипін берілген кітапханада көрсетілген Arduino тақтасының түйреуіштеріне дәл қосуымыз керек.

Мен тізбектегі светодиодтарды қуаттандыру үшін сыртқы көзді қолдандым (болашақта кейбір светодиодтар), бірақ егер ток шегінен аспаса, Arduino тақтасының VCC істікшесін де қолдана аламын.

Сондай-ақ, TLC5940 чипі шын мәнінде тұрақты ток қабылдағышы екенін есте ұстаған жөн, осылайша ток шығыс контактілері бағытында өтеді. Бұл жарық диодтарын қосқан кезде олардың катодтарын чиптің шығыс түйреуіштеріне қосып, анодты 5В тұрақты токқа қосу керек дегенді білдіреді.

Сонымен қатар, микросхеманың шығыс істікшесі арқылы өтетін токты шектеу үшін тізбекке 2 алынбалы конденсатор мен резистор қажет болады. Жоғарыдағы графикте біз бұл кедергінің кедергісі $2\text{к } \Omega$ болуы керек екенін анықтадық.



```
#include "Tlc5940.h"

void setup() {
  Tlc.init(0); //
}

void loop() {
  Tlc.set(0,4095);
  Tlc.update();
  delay(1000);
  for (int i = 0; i < 16; i++) {
    Tlc.set(i, 4095);
  }
  Tlc.update();
  delay(1000);
  Tlc.clear();
  Tlc.update();
  delay(1000);

  for (int i = 0; i < 16; i++) {
    Tlc.set(i, 4095);
    Tlc.update();
    delay(200);
    Tlc.clear();
    Tlc.update();
    delay(200);
  }
}
```

1.9 - сурет – TLC5940 арналған бағдарламалау

2 ПЛИС негізінде басқару бөлшегінің құрамы және программалау жолдары

ПЛИС (программаланатын логикалық интегралдық сызба) — бұл көп функционалды құрылғы, оның құрылымы және жұмыс істеу принциптері кәсіби қолданушының қажеттіліктеріне бейімделетін болады. ПЛИС құрылғылары әдетте басқару бөлшектері, сигналды өңдеу, телекоммуникация, автомобильдік құралдар және т.б. салаларда қолданылады.

ПЛИС негізіндегі басқару бөлшегінің құрамы

1. ПЛИС чипі: Ол негізгі процессор ретінде қызмет етеді және көптеген программа жүктелетін логикалық элементтер мен конфигурациялық жадынан тұрады.

2. Жад: ПЛИС чиптері жиі қосымша жадты қажет етеді, мысалы, SRAM, DRAM немесе Flash жады, программа коды мен мәліметтерді сақтау үшін.

3. Кіріс/шығыс порттары: Бұл арқылы ПЛИС сыртқы дүниемен байланыс жасайды. Көптеген ПЛИС модульдерінде GPIO (жалпы мақсаттағы кіріс/шығыс) порттары, сериялық порттар, USB, Ethernet және т.б. болады.

4. Таймерлер/санағыштар: Жүйенің нақты уақыт режимінде жұмыс істеуі үшін қажет.

5. Жиілік генераторы: Жүйенің жұмыс жиілігін анықтайды.

6. Өзге де құралдар: Кейбір ПЛИС модульдері АЦП (аналогты-цифрлық преобразователь), ДЦП (цифрлық-аналогты преобразователь) секілді функционалды құралдарды қамтиды.



2.1 - сурет – Altera фирмасының Stratix IV микропроцессоры



2.2-сурет – Xilinx ұсынған Спартандық FPGA

Alter-ді Xilinx-пен салыстырған кезде оның күшті және әлсіз жақтарын әртүрлі аспектілерден қарастырған жөн. Мұнда негізгі айырмашылықтарға шолу жасалады:

Өнімділік: Altera компаниясының Intel Agilex FPGA өнімділігі Xilinx Versal-ға қарағанда 30-50% - ға жоғары екені белгілі, ол 1 есе жоғары және 2 ватт, бұл оны инновациялар бойынша көшбасшы етеді.

Энергия тиімділігі: Xilinx FPGA-лары, әдетте, Altera FPGA-ға қарағанда энергияны үнемдейді.

Дизайн және әзірлеу: Xilinx сериялы FPGA тақталары негізінен қосымшаларды әзірлеуге арналған, Ал Altera сериялы FPGA тақталары өнеркәсіптік дизайн және ендірілген қолданбалар үшін қажет. Xilinx және Altera-ның нарықтағы жалпы үлесі шамамен 90% құрайды, бірақ Xilinx-тің шамалы артықшылығы бар.

Құралдар және қолдау: Xilinx көптеген құралдар мен қолдауға ие және әзірлеушілер арасында танымал.

Ғылыми зерттеулердің 90% - дан астамы Xilinx FPGA -да жүргізіледі, қоғаммен байланыс саласындағы ғылыми зерттеулердің 100% - ға жуығын Xilinx-да жұмыс істейтін зерттеушілер жүргізеді.

Құжаттар: Xilinx қоғаммен байланысқа қатысты техникалық құжаттарды, соның ішінде биттік ағындарды қалай жасау керектігін ұсынады, Бірақ Altera (қазіргі Intel) мұндай құжаттарды жарияламайды.

Графикалық интерфейс: Altera бағдарламалық жасақтамасының графикалық интерфейсі. Әдетте, Xilinx ISE-мен салыстырғанда пайдалану оңайырақ деп саналады.

Өнеркәсіптік қолданбалар: Altera компаниясының Stratix10sx650 сериясы xilinx құрылғыларына ұқсас перифериялық қосылым мен ресурстарды ұсынатын өнеркәсіптік қолданбалар үшін танымал таңдау болып табылады.

Нәтижесінде, Altera компаниясының Intel Agilex FPGA құрылғылары өнімділік тұрғысынан жақсырақ болғанымен, Xilinx FPGA құрылғылары энергияны үнемдейді және құралдар мен қолдаудың кең ауқымына ие. Xilinx сонымен қатар ғылыми-зерттеу және тәжірибелік-конструкторлық жұмыстарда көбірек танымал және Altera Stratix10sx650 сериясы өнеркәсіптік қолданбалар үшін өте қолайлы. Сіздің қажеттіліктеріңізге және жобаның техникалық сипаттамаларына байланысты Altera және Xilinx-ті таңдау алдыңғы аппараттық құралдармен үйлесімділік, өнімділік талаптары, құны және әзірлеу құралдарының қолжетімділігі сияқты бірқатар факторларға байланысты болуы мүмкін. Екі компания да сапалы өнім ұсынады және салалық стандарттар мен технологиялардың кең спектрін қолдайды.

2.1 ПЛИС программалау жолдары

ПЛИС программалау үшін бірнеше тәсілдер бар:

1. HDL (Hardware Description Language): Жабдық сипаттамасы тілдерін қолдану, мысалы, VHDL немесе Verilog. Бұл тілдер жоғары деңгейлі программалау тілдері сияқты, бірақ олар жабдықтың физикалық деңгейінде логикалық құрылымдарды сипаттайды.

2.Графикалық интерфейс: Жабдықтың логикалық схемаларын құру үшін графикалық интерфейсдерді қолдануға болады. Бұл тәсіл жаңадан бастаушылар үшін түсінікті болып келеді.

3. Қосымша бағдарламалық құралдар: Сонымен қатар, ПЛИС өндірушілері өздерінің чиптерін программалау үшін арнайы IDE және SDK ұсынады, мысалы, Xilinx-тің Vivado Design Suite немесе Altera-ның Quartus Prime.

ПЛИС-тің программалануы сондай-ақ жабдықтың физикалық сипаттамаларын ескереді және тиімді және оптимизацияланған жұмыс істеуін қамтамасыз етеді. Әрбір өндірушінің өзінің конфигурациялық протоколдары және құралдары болады, сондықтан нақты ПЛИС платформасын таңдағанда осы аспектілерді ескерген жөн.

Vivado Design Suite - бұл Xilinx компаниясы FPGA (бағдарламаланатын логикалық интегралды микросхемалар) негізіндегі электрондық жүйелерді әзірлеу үшін әзірлеген кешенді бағдарламалық жасақтама. Бұл құралдар

жинағы Ultrascale, 7 Series, Zynq-7000 және т.б. отбасыларды қоса алғанда, Xilinx ПЛИС архитектураларымен пайдалануға арналған. Vivado күрделі жүйелерді жобалау, талдау, модельдеу және іске асыру үшін бірқатар мүмкіндіктерді ұсынады.

Vivado design Suite негізгі компоненттері:

1. Vivado IDE: жобаны құру, басқару, синтездеу, орналастыру және маршруттау және уақытты талдауды қоса алғанда, жобалау процесінің басынан аяғына дейін барлық аспектілері үшін пайдаланушы интерфейсін қамтамасыз ететін интеграцияланған даму ортасы.

2. Vivado HLx Editions: Vivado-ның әртүрлі нұсқалары бар, соның ішінде WebPACK, design Edition және System Edition. Олардың әрқайсысы функционалдылықтың әртүрлі деңгейлерін ұсынады:

- WebPACK студенттер мен хоббистерге арналған негізгі құралдар жиынтығын ұсынады.

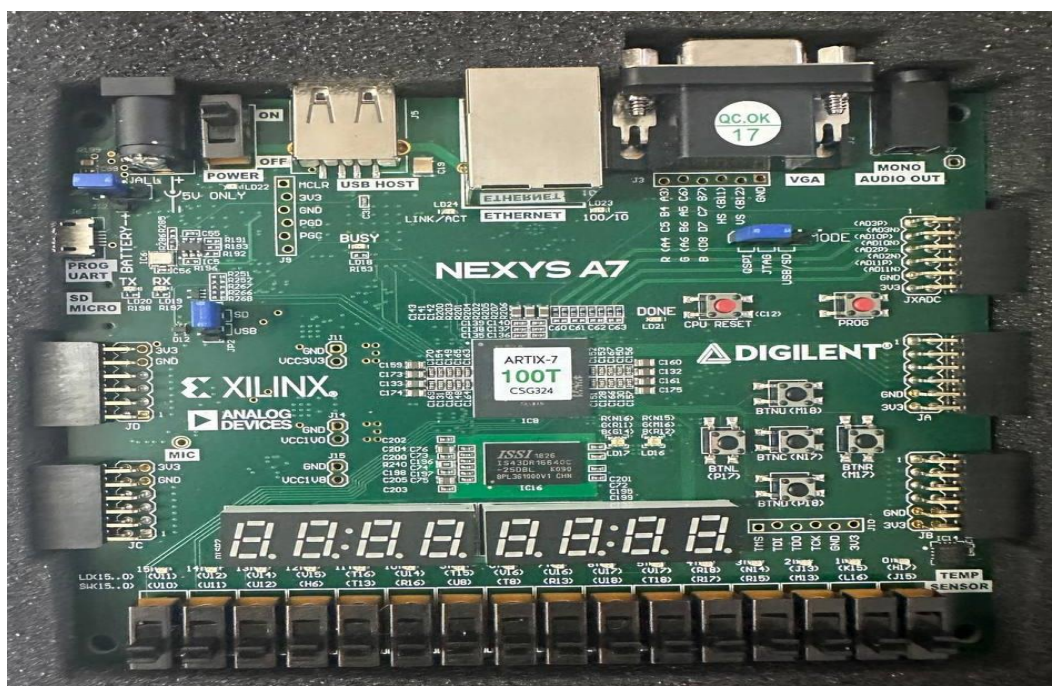
- Design Edition синтез бен талдауды қоса алғанда, кәсіби дамудың жетілдірілген мүмкіндіктерін ұсынады.

- System Edition барлық design edition мүмкіндіктерін және жүйелік дизайн мен тексеру деңгейіне арналған қосымша құралдарды қамтиды.

3. Vivado синтезаторы: VHDL немесе Verilog сипаттамаларын FPGA-да жүзеге асыруға болатын логикалық схемаларға түрлендіреді.

4. Уақыт анализаторы: өнімділік талаптарын қанағаттандыру үшін тізбектің уақыт сипаттамаларын талдауға және оңтайландыруға көмектеседі.

5. Тренажер: аппараттық платформада физикалық іске асырылғанға дейін оның жұмысын тексеру үшін функционалды және уақытша жобаны модельдеуге мүмкіндік береді.



2.3 - сурет – Xilinx фирмасының Nexys Artix-7 платасы

Негізгі функциялар мен мүмкіндіктер:

- IP Integrator: пайдаланушыларға өз жобаларында IP блоктарын оңай біріктіруге және конфигурациялауға мүмкіндік беретін графикалық дизайн құралы.

- Жоғары деңгейлі синтез (HLS): C/C++/System C кодын логикалық схемаларға түрлендіреді, бұл әзірлеушілерге аппараттық шешімдерді жобалау үшін жоғары деңгейлі бағдарламалау тілдерін пайдалануға мүмкіндік береді.

- Диагностика және жөндеу: Vivado аппараттық платформада модельдеу сатысында да, іске асырылғаннан кейін де аппараттық схемаларды жөндеуге мүмкіндік береді.

Дипломдық жұмыста қолдану:

Дипломдық жұмыс контекстінде, Vivado Design Suite үшін пайдаланылуы мүмкін:

- Нөлден бастап ПЛИС-бағдарланған жобаны жобалау және іске асыру.
- Әзірлеу процесін жеделдету үшін дайын IP пайдалану.
- Сіздің жобаңыздың функционалдығын қамтамасыз ету үшін модельдеу және күйін келтіру.

- ПЛИС өнімділігін талдау және ресурстарды оңтайландыру.

Xilinx 's Vivado Design Suite бұл жай ғана FPGA жобалау құралы емес, ол әзірлеушілерге синтезден бастап орналастыру мен маршруттауға, уақыт пен өнімділікті талдауға дейінгі барлық қажетті құралдармен қамтамасыз ететін қуатты орта. Vivado инженерлерге күрделі жүйелерді чипте (SoC) және ПЛИС-те тиімді жүзеге асыруға мүмкіндік беретін блоктарға немесе модульдерге негізделген әдістемені қолданады.

Тиімділік пен өнімділікті арттыруға арналған Vivado ерекшеліктері:

Жақсартылған архитектура: Vivado соңғы Xilinx plis архитектураларымен пайдалану үшін оңтайландырылған, бұл кристалды ресурстарды пайдалану кезінде жоғары өнімділік пен тиімділікке қол жеткізуге мүмкіндік береді.

Үлкен деректерді өңдеуді қолдау: Vivado интеграцияланған аналитикалық құралдарымен үлкен көлемдегі деректерді талдауға және басқаруға болады, бұл Машиналық оқыту және үлкен деректерді өңдеу сияқты заманауи қосымшалар үшін өте маңызды.

Systemverilog тілімен Vivado қолдану мысалы:

Vivado-да әзірлеуге және модельдеуге болатын systemverilog жобасының қарапайым мысалын қарастырайық. Бұл мысалда біз қарапайым 4 биттік қосқышты жүзеге асыратын модуль жасаймыз:

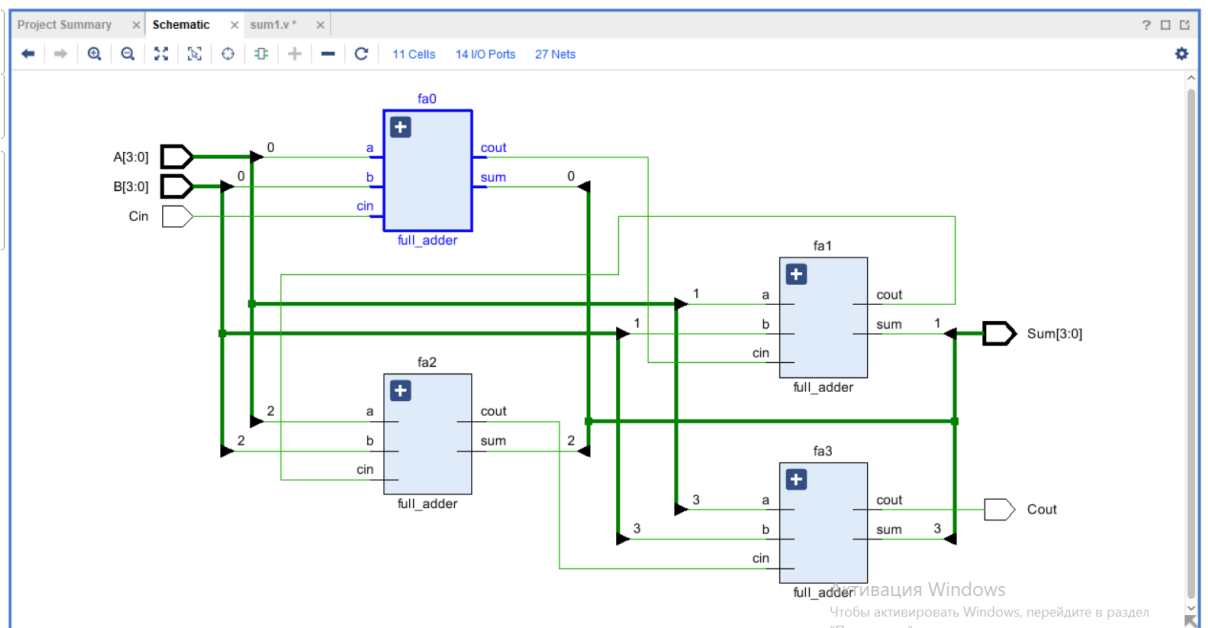
```

Project Summary | Schematic | sum1.v*
C:\Users\KALZHAN\sumator\sumator\srcs\ources_1\new\sum1.v

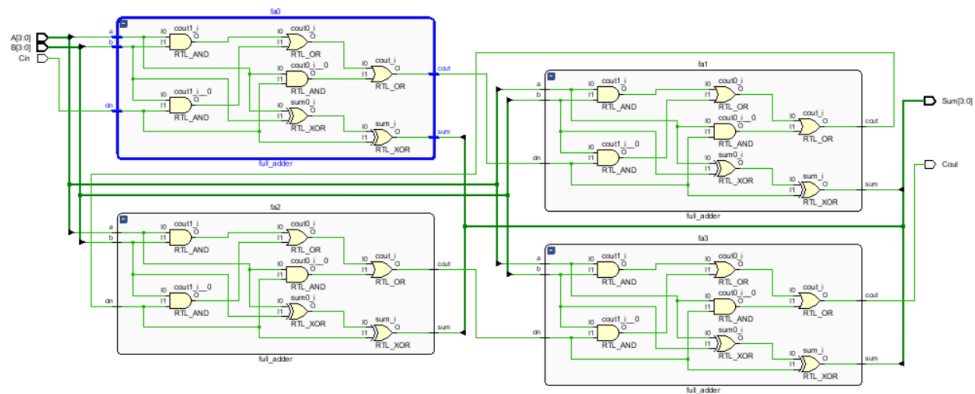
1 module full_adder_4bit(
2   input [3:0] A,
3   input [3:0] B,
4   input Cin,
5   output [3:0] Sum,
6   output Cout
7 );
8 wire [2:0] carry;
9 full_adder fa0(
10  .a(A[0]),
11  .b(B[0]),
12  .cin(Cin),
13  .sum(Sum[0]),
14  .cout(carry[0])
15 );
16 full_adder fa1(
17  .a(A[1]),
18  .b(B[1]),
19  .cin(carry[0]),
20  .sum(Sum[1]),
21  .cout(carry[1])
22 );
23 full_adder fa2(
24  .a(A[2]),
25  .b(B[2]),
26  .cin(carry[1]),
27  .sum(Sum[2]),
28  .cout(carry[2])
29 );
30 full_adder fa3(
31  .a(A[3]),
32  .b(B[3]),
33  .cin(carry[2]),
34  .sum(Sum[3]),
35  .cout(Cout)
36 );
37 endmodule
38
39 module full_adder(
40   input a, b, cin,
41   output sum, cout
42 );
43
44 assign sum = a ^ b ^ cin; // XOR для сумирования
45 assign cout = (a & b) | (b & cin) | (a & cin); // OR для переноса
46 endmodule

```

2.4 - сурет - 4-биттік Сумматор



2.5 - сурет - Схема түріндегі 4-биттік сумматор



2.6 - сурет – Сумматордың ішкі құрылысы

Xilinx компаниясы американдық компания, алдымен 1984 жылы Росс Фримен мен Бернанд Вонасоски тарапынан Сан-Хосе қаласында тұрған. Олар компанияның басты мақсатын FPGA (Field-Programmable Gate Array) технологиясын дамыту үшін алатын.

Xilinx өзінің тарихы бойынша инновацияларды енгізу бойынша көшбасшы компаниялардың бірі болып табылады. Компания FPGA технологиясында көптеген алғашқылардың бірі болып саналады және олар өз өнімдерінің көмегімен телекоммуникациялар, автомобиль өнеркәсібі, қорғаныс және аэроғарыш салаларында кеңінен қолданыс тапты.

Сондай-ақ, Xilinx FPGA технологиясына арналған екіге жеткізу және таңдау мәселелерін решетуде кеңес беру компаниясы ретінде да білінеді.

3 Басқару бөлшегінің негізгі интерфейсы SPI

Verilog-да SPI басқаруды жүзеге асыру үшін SPI интерфейсі арқылы деректерді беру мен қабылдауды басқаратын модульдің негізгі құрылымынан бастауға болады. Бұл мысалда мен шебердің қарапайым SPI іске асырылуын көрсетемін. Шебердің SPI модулі коммуникацияны бастайды, сағаттық сигналдарды шығарады және MOSI және MISO желілері арқылы деректерді беру мен қабылдауды басқарады.



3.1 - сурет – SPI интерфейсы

Құрамалық сұлба

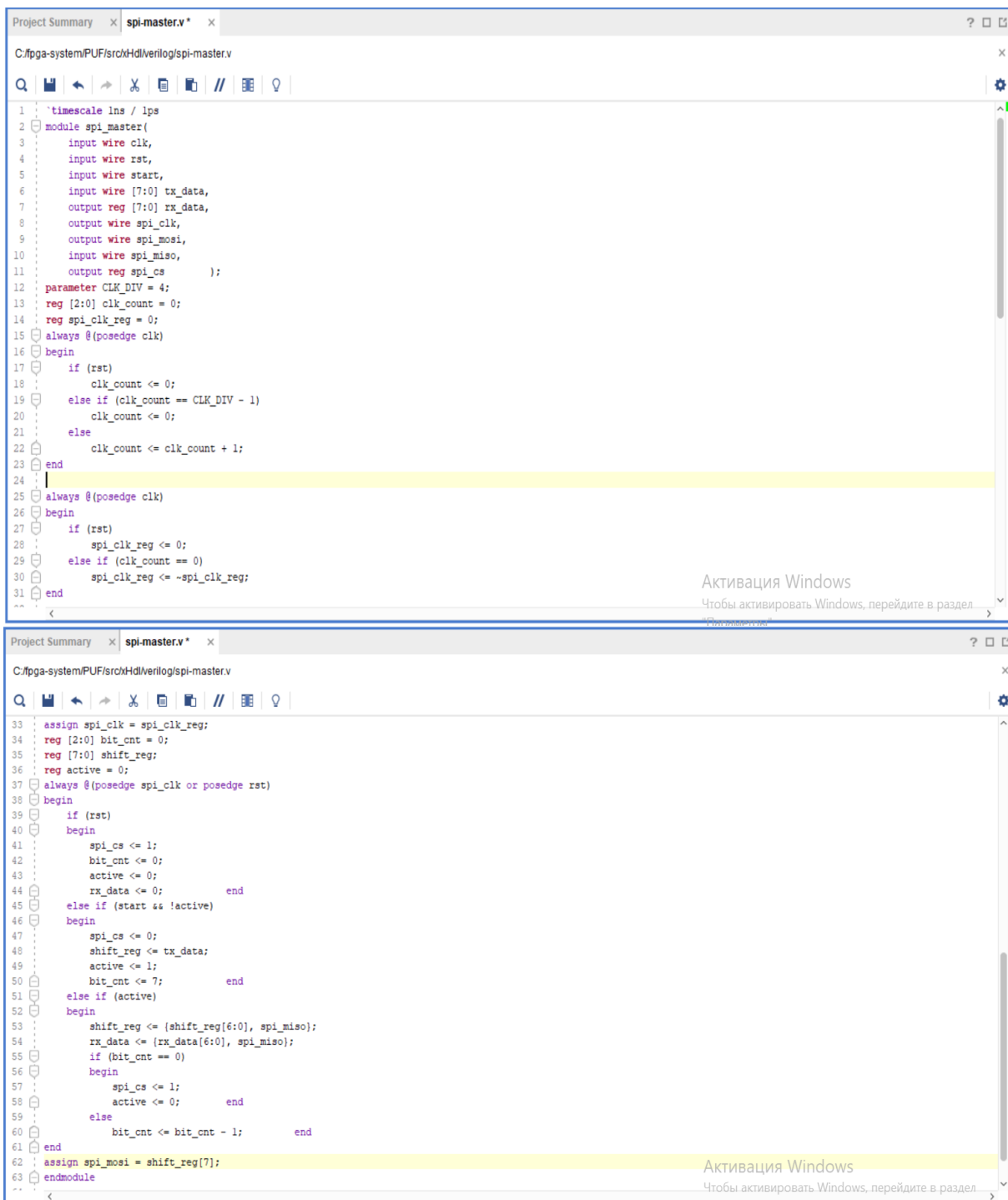
1. Моторды басқару бөлшегі.
2. Негізгі орталық процессор
3. 3D форматындағы воксельдік деректерді сақтайтын жады

Жалпы құрамалық сұлбада Орталық процессордан екі тарау жүреді. Біріншісі мотор басқару. Екіншісі үш немесе төр топқа параллелді деректерді жоғары жылдамдықпен SPI интерфейсімен беру бөлшегі. Үш немесе төрт топқа келген деректерді жеті ардуино nano процессорларына бөліп беру. Себебі жеті қабат цилиндрлік дисплейден тұрады. Әр линейкада MP4 форматы бойынша 240 сәулелі диод орналасады. Оны бір ардуино nano басқара алады ма? Есептеу керек. Диплом жақсы болу үшін негізі үш топтағы жеті наноның жұмысын БЛИС орындауы тиіс. Онда БЛИС тан 21 жоғары жылдамдықты SPI интерфейсі шығуы қажет. SPI интерфейсі жайында жаз. Оның тағы екі түрі бар. SPI quadro SPI duo деген. Жылдамдық екі есе және төрт есе өседі.

Алгоритмдер жайлы. Воксель дегеніміз әр воксель үш байттан тұрады. RGB түстері бойынша. Әр вокселдің координаты XYZ бойынша анықталады. Вокселдердің арасы пиксель сияқты мысалы бір мм тұрады, барлық үш бағыт

бойынша. Ол МРТ қондырғыларында, 3D принтерлерде қолданылады. Ол куб матрица. Содан шеңбер бойынша бір мм қадаммен ең жақын пикселдерді теріп алып бір тізбекті ретінде 21 бағыт бойынша параллелді шығарыпберу керек.

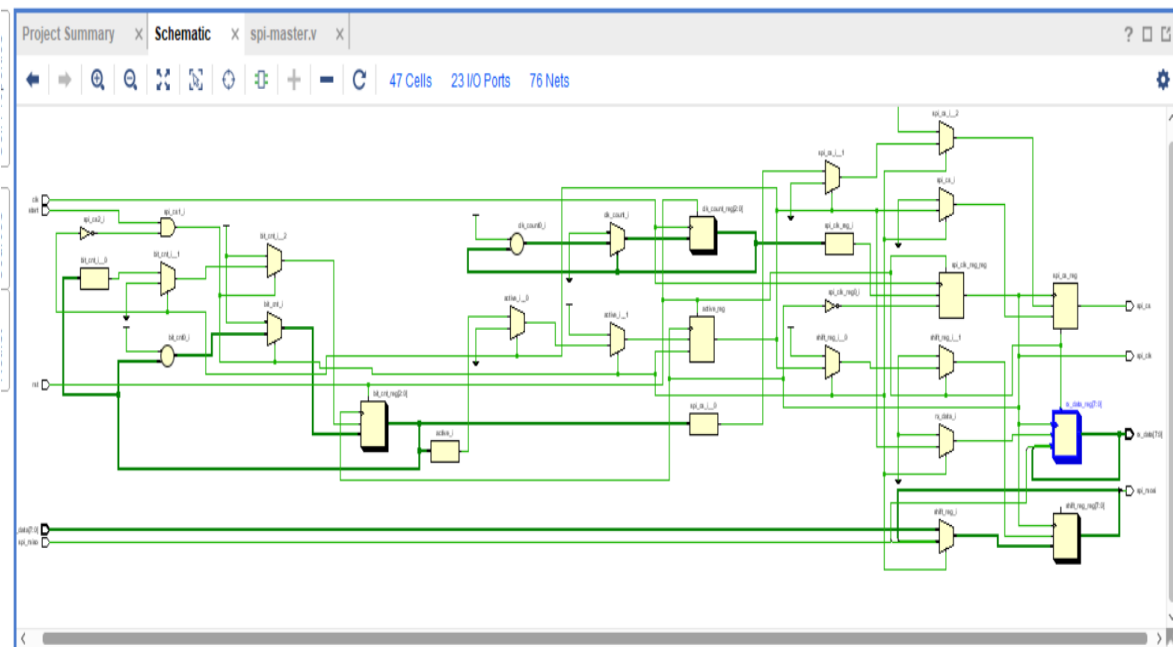
Бұл алгоритмдерді сурет ретінде салып шығып түсініктепе беру керек. Куб матрицасының өлшемі 256x256x256 деп бекітіп содан цилиндрлік кесіндінің 240-ын жеті қабатын шығарып алу керек. Бұл операция 256 микросекундта орындалуы тиісті.



```
1 `timescale 1ns / 1ps
2 module spi_master(
3     input wire clk,
4     input wire rst,
5     input wire start,
6     input wire [7:0] tx_data,
7     output reg [7:0] rx_data,
8     output wire spi_clk,
9     output wire spi_mosi,
10    input wire spi_miso,
11    output reg spi_cs
12);
13 parameter CLK_DIV = 4;
14 reg [2:0] clk_count = 0;
15 reg spi_clk_reg = 0;
16 always @(posedge clk)
17 begin
18     if (rst)
19         clk_count <= 0;
20     else if (clk_count == CLK_DIV - 1)
21         clk_count <= 0;
22     else
23         clk_count <= clk_count + 1;
24 end
25 always @(posedge clk)
26 begin
27     if (rst)
28         spi_clk_reg <= 0;
29     else if (clk_count == 0)
30         spi_clk_reg <= ~spi_clk_reg;
31 end
32
33 assign spi_clk = spi_clk_reg;
34 reg [2:0] bit_cnt = 0;
35 reg [7:0] shift_reg;
36 reg active = 0;
37 always @(posedge spi_clk or posedge rst)
38 begin
39     if (rst)
40     begin
41         spi_cs <= 1;
42         bit_cnt <= 0;
43         active <= 0;
44         rx_data <= 0;     end
45     else if (start && !active)
46     begin
47         spi_cs <= 0;
48         shift_reg <= tx_data;
49         active <= 1;
50         bit_cnt <= 7;     end
51     else if (active)
52     begin
53         shift_reg <= {shift_reg[6:0], spi_miso};
54         rx_data <= {rx_data[6:0], spi_miso};
55         if (bit_cnt == 0)
56         begin
57             spi_cs <= 1;
58             active <= 0;     end
59         else
60             bit_cnt <= bit_cnt - 1;     end
61     end
62 assign spi_mosi = shift_reg[7];
63 endmodule
```

3.2 - сурет – SPI интерфейсі үшін Verilog тіліндегі код

Verilog-да SPI басқаруды жүзеге асыру үшін SPI интерфейсі арқылы деректерді беру мен қабылдауды басқаратын модульдің негізгі құрылымынан бастауға болады. Бұл мысалда мен қарапайым SPI іске асырылуын көрсетемін. SPI модулі коммуникацияны бастайды, сағаттық сигналдарды шығарады және MOSI және MISO желілері арқылы деректерді беру мен қабылдауды басқарады.



3.3 - сурет – RTL кескіні

Ең бастысы айтылды. Ал енді негізгі практикаға көшетін болсақ. Бізде барлық SPI арнада 630 МГц жылдамдықта жұмыс істеу керек. Біз оны 21 SPI арнасына бөлетін болсақ әр-қайсысына 30 МГц бөлінеді. Ал бұл тапсырмамен тек БЛИС-қана атқара алатын. Жоғарыда біз тек ең қарапайым 1 арналы SPI интерфейсіне арналған Systemverilog тілінде жазылған кодты байқадық. Бізге ең бастысы ол SPI арна бірін-біріне параллель орналасу керек.

3.1 Басқару бөлшегінің құрамы

Басқару бөлшегі – бұл электрондық құрылғылардың негізгі компоненті, ол жүйені басқару және оның жұмысын үйлестіру үшін пайдаланылады. Басқару бөлшегінің құрамы келесі негізгі компоненттерден тұрады:

1. Процессор:

- CPU (Орталық процессорлық құрылғы): Басқару командаларын орындайтын негізгі есептеу құрылғысы.
- FPGA (Программаланатын логикалық массивтер): Логикалық функцияларды жылдам орындау үшін қолданылатын қайта программаланатын құрылғы.

- DSP (Сандық сигнал процессоры): Сандық сигналдарды өңдеу үшін арнайы жасалған процессор.

2. Жад (Memory):

- ROM (Тұрақты жады): Бағдарламалар мен тұрақты деректерді сақтау үшін.

- RAM (Жедел жады): Уақытша деректер мен аралық нәтижелерді сақтау үшін.

- Flash жады: Бағдарламаларды және деректерді қайта жазуға және сақтауға арналған жады.

3. Кіріс/Шығыс порттары (I/O Ports):

- GPIO (Жалпы мақсаттағы кіріс/шығыс): Сыртқы құрылғылармен байланыс жасау үшін.

- Сериялық порттар (UART, SPI, I2C): Микроконтроллер мен перифериялық құрылғылар арасында деректер алмасу үшін.

- USB, Ethernet: Компьютерлер мен желілерге қосылу үшін.

4. Таймерлер мен Санағыштар (Timers and Counters):

- Жүйелік уақытты басқару және нақты уақыт режиміндегі операцияларды орындау үшін.

5. Аналогты және цифрлық преобразователдер (ADC/DAC):

- ADC (Аналогты-цифрлық преобразователь): Аналогтық сигналдарды цифрлық форматқа түрлендіру үшін.

- DAC (Цифрлық-аналогты преобразователь): Цифрлық сигналдарды аналогтық форматқа түрлендіру үшін.

6. Коммуникациялық модульдер (Communication Modules):

- Wi-Fi, Bluetooth, Zigbee: Сымсыз байланыс үшін;

- Ethernet, CAN (Controller Area Network) Сымды байланыс үшін.

7. Жиілік генераторлары (Clock Generators):

- Жүйелік сағат сигналдарын генерациялау үшін.

8. Қуат басқару модульдері (Power Management Modules):

- Жүйенің тұрақты жұмысын қамтамасыз ету үшін кернеу реттеушілері, қуат бақылау элементтері.

Мысал: Vivado Design Suite пайдалану

Vivado Design Suite қолдану арқылы басқару бөлшегін жобалаудың негізгі қадамдары:

1. Жаңа жоба құру:

- Vivado-ны ашып, жаңа жоба жасаңыз. Құрылғыны және оның параметрлерін таңдаңыз.

2. Модульдер құру:

- Verilog немесе VHDL кодтарын жазыңыз. Мысалы, жоғарыда берілген толық сумматор модулін пайдаланыңыз;

- Басқару логикасын және интерфейстерді анықтаңыз.

3. IP Integrator пайдалану:

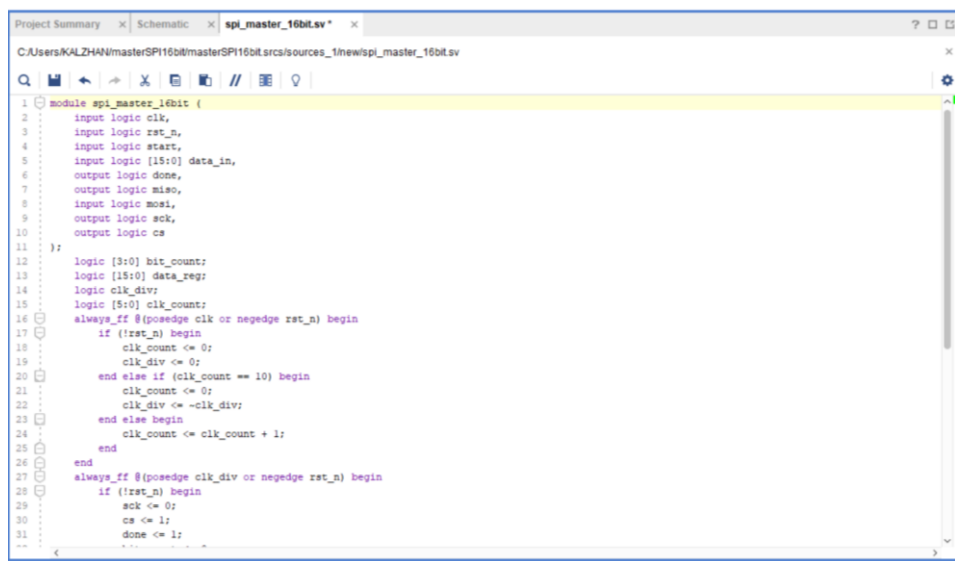
- Графикалық интерфейс арқылы түрлі IP (Intellectual Property) блоктарды қосып, интеграциялаңыз.

4. Синтез және жүктеу:

- Жобаны синтездеп, қажетті FPGA немесе SoC құрылғысына жүктеңіз;
- Симуляция және тестілеу арқылы жобаның дұрыстығын тексеріңіз.

Бұл процесс электрондық құрылғыларды және жүйелерді жылдам және тиімді жобалауға мүмкіндік береді.

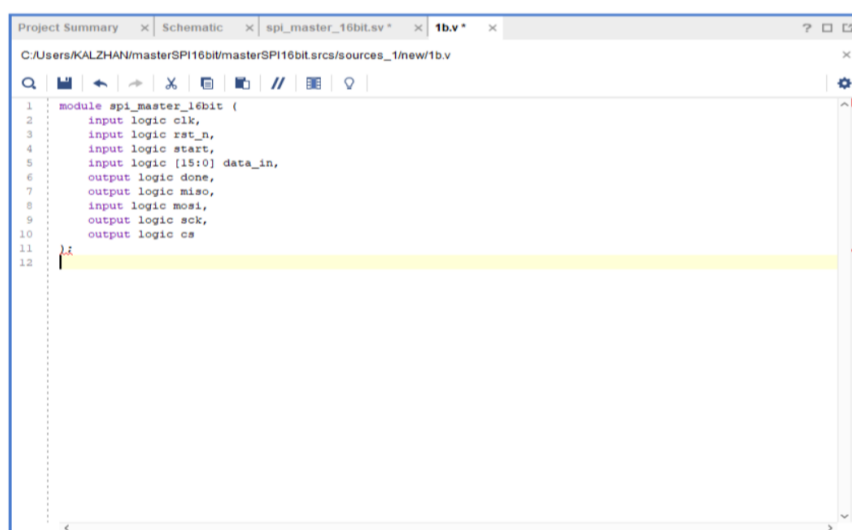
4 БЛИС негізгі сұлбалары және бағдарламалау



```
1 module spi_master_16bit (  
2     input logic clk,  
3     input logic rst_n,  
4     input logic start,  
5     input logic [15:0] data_in,  
6     output logic done,  
7     output logic miso,  
8     input logic mosi,  
9     output logic sck,  
10    output logic cs  
11 );  
12    logic [3:0] bit_count;  
13    logic [15:0] data_reg;  
14    logic clk_div;  
15    logic [5:0] clk_count;  
16    always_ff @(posedge clk or negedge rst_n) begin  
17        if (!rst_n) begin  
18            clk_count <= 0;  
19            clk_div <= 0;  
20        end else if (clk_count == 10) begin  
21            clk_count <= 0;  
22            clk_div <= -clk_div;  
23        end else begin  
24            clk_count <= clk_count + 1;  
25        end  
26    end  
27    always_ff @(posedge clk_div or negedge rst_n) begin  
28        if (!rst_n) begin  
29            sck <= 0;  
30            cs <= 1;  
31            done <= 1;  
32        end  
33    end  
34 endmodule
```

4.1 - сурет – Бұл көрсетілген код 16 биттік SPI мастер SystemVerilog тілінде.

Бұл кодты білмейтін, тақырыпта хабары жоқ адамға түсіндіретін болсақ. Келесі суретті қарасақ болады:



```
1 module spi_master_16bit (  
2     input logic clk,  
3     input logic rst_n,  
4     input logic start,  
5     input logic [15:0] data_in,  
6     output logic done,  
7     output logic miso,  
8     input logic mosi,  
9     output logic sck,  
10    output logic cs  
11 );  
12
```

4.2 - сурет – Модуль бөлімі

- clk: негізгі тактілік сигнал.
- rst_n: қалпына келтіру сигналы (активті төмен).
- start: жіберуді бастайтын сигнал.
- data_in: SPI арқылы жіберілетін кіріс деректер.
- done: жіберудің аяқталғанын көрсететін сигнал.

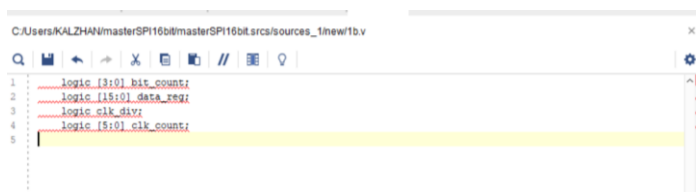
miso: Master In Slave Out - деректерді слейвтен мастерге жіберетін желі.

mosi: Master Out Slave In - деректерді мастерден слейвке жіберетін желі.

sck: Serial Clock - деректерді жіберуге арналған тактілік сигнал.

cs: Chip Select - құрылғыны таңдау сигналы.

Бұл ұсынылған тек модульдік сипаттамасы.



```
C:\Users\KALZHAN\masterSPI16bit\masterSPI16bit.srcs\sources_1\new1bv
1 logic [3:0] bit_count;
2 logic [15:0] data_reg;
3 logic clk_div;
4 logic [8:0] clk_count;
```

4.3 - сурет – Ішкі сигналдар

bit_count: қанша бит жіберілгенін бақылауға арналған санағыш.

data_reg: жіберу кезінде деректерді сақтау үшін арналған регистр.

clk_div: SPI жылдамдығын басқару үшін тактілік сигналды бөлу.

clk_count: тактілік сигналдың жиілігін бөлуге арналған санағыш.

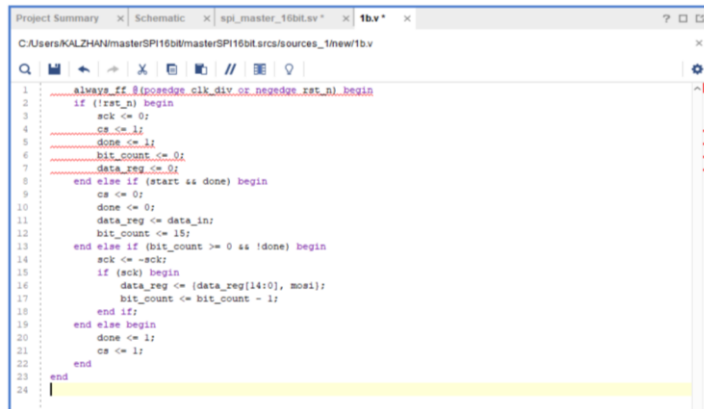


```
Project Summary | Schematic | spi_master_16bit.v* | 1bv*
C:\Users\KALZHAN\masterSPI16bit\masterSPI16bit.srcs\sources_1\new1bv
1 always_ff # (posedge clk or negedge rst_n) begin
2   if (!rst_n) begin
3     clk_count <= 0;
4     clk_div <= 0;
5   end else if (clk_count == 10) begin
6     clk_count <= 0;
7     clk_div <= -clk_div;
8   end else begin
9     clk_count <= clk_count + 1;
10  end
11 end
12
```

4.4 - сурет – Тактілік сигналды бөлу:

Бұл блок негізгі тактілік сигналды төменгі жиілікке бөледі. clk_count 10-ға жеткенде, clk_div өзгереді және clk_count нөлге орнатылады.

4.1 Негізгі SPI логикасы



```
1 always_ff @(posedge clk_div or negedge rst_n) begin
2   if (!rst_n) begin
3     sck <= 0;
4     cs <= 1;
5     done <= 1;
6     bit_count <= 0;
7     data_reg <= 0;
8   end else if (start == done) begin
9     cs <= 0;
10    done <= 0;
11    data_reg <= data_in;
12    bit_count <= 15;
13  end else if (bit_count >= 0 && !done) begin
14    sck <= ~sck;
15    if (sck) begin
16      data_reg <= {data_reg[14:0], mosi};
17      bit_count <= bit_count - 1;
18    end if;
19  end else begin
20    done <= 1;
21    cs <= 1;
22  end
23 end
24
```

4.5 - сурет – Негізгі SPI логикасы

Бұл блок жіберу процесін басқарады: Қалпына келтіру кезінде барлық сигналдар бастапқы күйге оралады. Егер start белсенді болса және жіберу аяқталса (done), жаңа жіберу басталады: cs төмендетіледі, және деректер data_reg-ке жүктеледі. Жіберу процесінде sck әр тактта ауысып, деректер data_reg-ке жылжиды. Барлық биттер жіберілгенде, done 1-ге орнатылады және cs жоғары көтеріледі.

Бұл модуль 16-биттік SPI мастерін көрсетеді, ол:

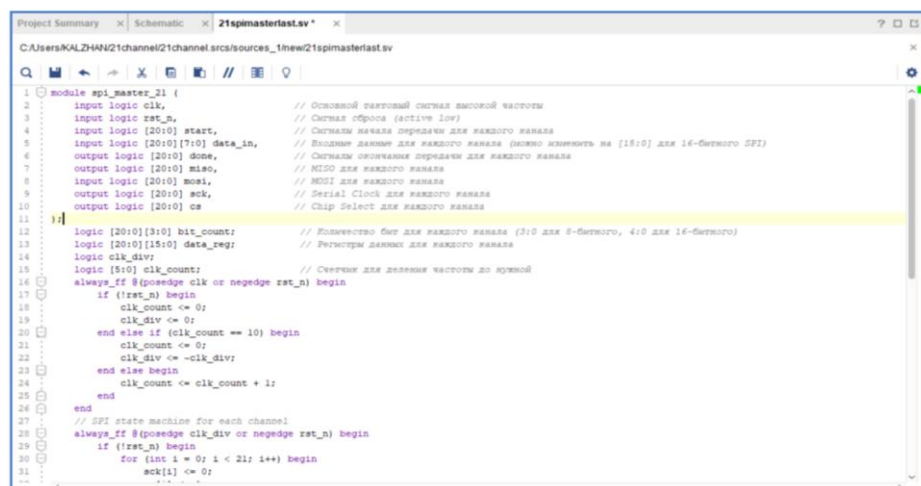
Негізгі тактілік сигналды SPI үшін төменгі жиілікке бөледі.

Start сигналы белсенді болған кезде жіберуді бастайды.

data_in-ке жүктелген деректерді miso арқылы биттік түрде жібереді, және data_reg-ке жаңа деректерді mosi арқылы қабылдайды.

Жіберу аяқталғаннан кейін done 1-ге орнатылады.

4.2 Басқару бөлшегін программалау



```
1 module spi_master_2l (
2   input logic clk, // Основной тактовый сигнал высокой частоты
3   input logic rst_n, // Сигнал сброса (active Low)
4   input logic [20:0] start, // Сигналы начала передачи для каждого канала
5   input logic [20:0][7:0] data_in, // Входные данные для каждого канала (можно изменить на [15:0] для 16-битного SPI)
6   output logic [20:0] done, // Сигналы окончания передачи для каждого канала
7   output logic [20:0] miso, // MISO для каждого канала
8   input logic [20:0] mosi, // MOSI для каждого канала
9   output logic [20:0] sck, // Serial Clock для каждого канала
10  output logic [20:0] cs // Chip Select для каждого канала
11 );
12 logic [20:0][3:0] bit_count; // Количество бит для каждого канала (3:0 для 8-битного, 4:0 для 16-битного)
13 logic [20:0][15:0] data_reg; // Регистры данных для каждого канала
14 logic clk_div;
15 logic [5:0] clk_count; // Счетчик для деления частоты до нужной
16 always_ff @(posedge clk or negedge rst_n) begin
17   if (!rst_n) begin
18     clk_count <= 0;
19     clk_div <= 0;
20   end else if (clk_count == 10) begin
21     clk_count <= 0;
22     clk_div <= ~clk_div;
23   end else begin
24     clk_count <= clk_count + 1;
25   end
26 end
27 // SPI state machine for each channel
28 always_ff @(posedge clk_div or negedge rst_n) begin
29   if (!rst_n) begin
30     for (int i = 0; i < 21; i++) begin
31       sck[i] <= 0;
32     end
33   end
34 end
35
```

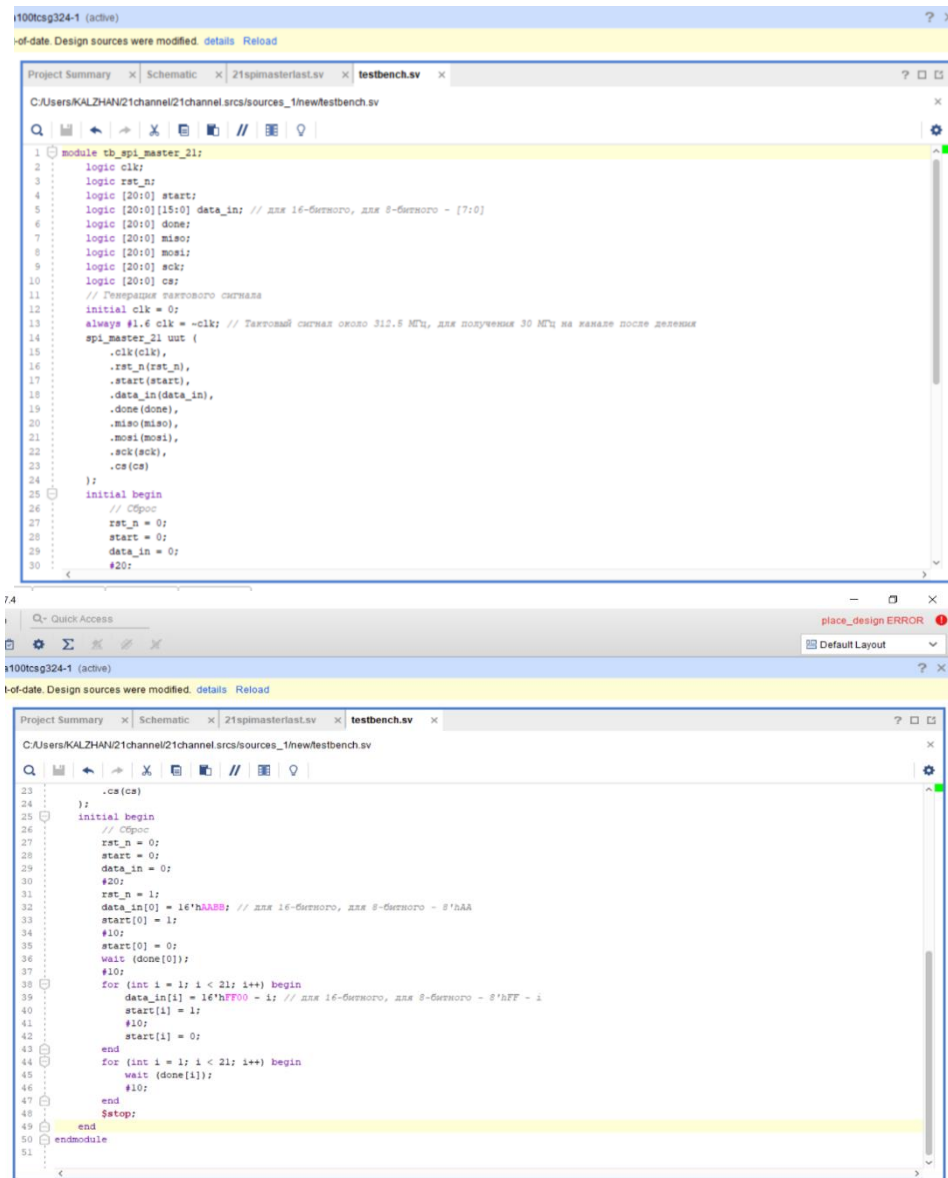


```
Project Summary x Schematic x 21spimasterlast.v* x
C:\Users\KALZHAN\2\channel2\channel2\srcs\sources_1\new21spimasterlast.v

31 sck[i] <= 0;
32 cs[i] <= 1;
33 done[i] <= 1;
34 bit_count[i] <= 0;
35 data_reg[i] <= 0;
36 end
37 end else begin
38   for (int i = 0; i < 21; i++) begin
39     if (start[i] == done[i]) begin
40       cs[i] <= 0;
41       done[i] <= 0;
42       data_reg[i] <= data_in[i];
43       bit_count[i] <= 15; // для 16-битного, для 8-битного - 7
44       end else if (bit_count[i] >= 0 == !done[i]) begin
45         sck[i] <= ~sck[i];
46         if (sck[i]) begin
47           data_reg[i] <= {data_reg[i][14:0], mosi[i]}; // для 16-битного, для 8-битного - 6:0
48           bit_count[i] <= bit_count[i] - 1;
49         end
50       end else begin
51         done[i] <= 1;
52         cs[i] <= 1;
53       end
54     end
55   end
56 end
57 always_ff @(negedge sck) begin
58   for (int i = 0; i < 21; i++) begin
59     miso[i] <= data_reg[i][15]; // для 16-битного, для 8-битного - 7
60   end
61 end
```

4.6 - сурет – SystemVerilog тілінде жазылған бағдарлама

Бұл кодты дайын деп есептеседе болады. Толыққанды жұмыс жасайтын 21 арналы SPI интерфейсының коды. Әрине бұл кодты тексеру үшін Vivado программасында өзінің ішіне енгізілген «Ішкі тексеру» функциясы бар. Нақтырақ айтқанда қолымызда тексеру құрылғысы, яғни баспа платасы болмаса. Осы функцияны қолдана отырып, программа ішінде өзіміз қосып тексерсекболады. Ол үшін негізгі кодтан бөлек testbench кодын жазу міндет болады.



4.7 - сурет – Testbench тестілеуі

Vivado бағдарламасының тағы бір бөлігі ол «testbench» жазылған кодты тексеру бөлігі. Жазылған кодты электронды түрде тексеру.

ҚОРЫТЫНДЫ

Бұл дипломдық жұмыста FPGA негізінде 21 SPI интерфейсінің жобалау және жүзеге асыру қарастырылды. Жұмыстың негізгі мақсаты - әрбір SPI желісінің жылдамдығын 30 МГц қамтамасыз етіп, жалпы жүйенің 630 МГц жылдамдығына қол жеткізу болды. Жобаның мақсаты - жоғары жылдамдықты деректерді беруді талап ететін 3D голограмма моторын басқару жүйесін құру.

Жұмыстың барысында келесі міндеттер орындалды:

1. Теориялық негіздеме: SPI интерфейсінің негіздері, оның жұмыс принциптері және FPGA технологиясы зерттелді. Бұл бөлімде SPI интерфейсінің артықшылықтары мен шектеулері талқыланып, оның жоғары жылдамдықты деректерді беру үшін қолайлы екендігі анықталды.

2. Жобалау: 21 SPI интерфейсінің құрылымы мен жұмыс принципі анықталды. Әрбір SPI желісінің 16-биттік деректерді жоғары жылдамдықта беру қабілеті қарастырылды. Бұл бөлімде жүйенің негізгі компоненттері мен олардың өзара әрекеттесу принциптері сипатталды.

3. Бағдарламалау: VHDL және SystemVerilog тілдері қолданылып, 21 SPI интерфейсінің бағдарламалық кодтары жазылды. Кодтар әрбір SPI желісінің деректерді жіберу және қабылдау процестерін қамтамасыз ететін модульдерді қамтыды.

4. Тестілеу және тексеру: Жүйенің жұмысын тексеру үшін тестілік сценарийлер әзірленді. Тестілеу нәтижелері бойынша жүйенің 630 МГц жылдамдықта жұмыс істейтіндігі расталды.

5. Қолдану: Жобаланған жүйе 3D голограмма моторын басқару үшін қолданылды. Жүйенің жоғары жылдамдықты деректерді беру қабілеті голограмма сапасын арттыруға мүмкіндік берді.

Бұл дипломдық жұмыс барысында орындалған міндеттер жүйенің тиімділігін көрсетті және оның нақты қолдану саласында жоғары өнімділікке қол жеткізетінін дәлелдеді. FPGA негізіндегі 21 SPI интерфейсі деректерді жоғары жылдамдықта беру мүмкіндігіне ие болып, күрделі басқару жүйелерін жүзеге асыруға мүмкіндік береді. Жұмыстың нәтижелері тек 3D голограммалар саласында ғана емес, сонымен қатар басқа да жоғары жылдамдықты деректерді беру талап ететін салаларда қолданылуы мүмкін.

Осылайша, дипломдық жұмыс мақсаттары толық орындалды және алынған нәтижелер қойылған міндеттерге сәйкес келеді. Жұмыс нәтижелері алдағы уақытта одан әрі жетілдіріліп, жаңа жобаларда қолдануға мүмкіндік береді.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Обзор шины SPI и разработка драйвера ведомого SPI устройства для embedded (<https://habr.com/ru/articles/123145/>) 2011 ж.
- 2 Запускаем сервис бесплатной аренды отладочных плат с FPGA и не только с FPGA (https://fpga-systems.ru/publ/raznoe/site_instructions/zapускаем_servis_besplatnoj_arendy_otladochnykh_plat_s_fpga_i_ne_tolko_s_fpga/50-1-0-163)
- 3 Макросы для sprint layout 4-5 (<https://radio-hobby.org/modules/tdm/downloads/singlefile.php?cid=6&lid=23>) 2020 ж.
- 4 Многоканальная система измерения положения на базе FPGA-PLC (<https://www.sciencedirect.com/science/article/abs/pii/S0019057821000148>) 2021 ж.
- 5 Многоканальная система сбора данных на базе FPGA для детекторов перегретой эмульсии (<https://www.sciencedirect.com/science/article/abs/pii/S0168900221004423>) 2021 ж.
- 6 Многоканальная система измерения положения на базе FPGA-PLC (<https://www.sciencedirect.com/science/article/abs/pii/S0019057821000148>) 2021 ж.
- 7 Язык описания аппаратуры Verilog HDL (<https://marsohod.org/verilog>)
- 8 Xilinx (<https://ru.wikipedia.org/wiki/Xilinx>)
- 9 Увеличение количества ШИМ контактов Arduino с помощью TLC5940 (<https://microkontroller.ru/arduino-projects/uvelichenie-kolichestva-shim-kontaktov-arduino-s-pomoshhyu-tlc5940/>) 2022 ж.
- 10 Разработка интерфейса SPI на ПЛИС (https://fpga-systems.ru/publ/raznoe/interfaces/spi_chast_1_obshhie_svedeniya/26-1-0-90) 2019 ж.

РЕЦЕНЗИЯ

Дипломдық жұмыс

Тажен Қалмұхамед Әбубәкірұлы

6B07112 – Electronic and Electrical Engineering

Тақырыбына: «БЛИС негізінде көп арналы басқару блогын әзірлеу»

Орындалды:

- а) графикалық бөлім 28 парақ;
б) түсініктеме 31 бет.

ЖҰМЫСҚА ЕСКЕРТУ

Берілген бітіру жұмысында БЛИС негізінде көп арналы басқару блогын әзірлеу туралы ақпарат жиналған. Негізгі өлшемдер жүргізіп, параметрлері есептелген. БЛИС қолдана отырып, басқару блогы көрсетіліп, есептеулер жасалған. Жоба сұлба бойынша құрастырылған.

Басқару блогын қолдануды жақсарту мәселелері қарастырылады. Жұмыста басқару блогы жайында мағлұматтар қарастырылған және оларды қолданудың бірнеше әдісі айтылған.

Басқару блоктарына талдау жасалып, осы өлшемдерде олардың тиімділігі анықталды. Сонымен қатар оларды одан әрі пайдалану және жетілдіру бойынша практикалық ұсыныстар беру. Дипломдық жұмыста басқару блоктары есептеулерін, құрылымы сызбасында студент өз тарапынан қандай жақсартулар енгізуі мүмкіндігін көрсете алмаған. Кейбір орфографиялық қателер кездеседі.

Графикалық және мәтіндік материалдар МСТҚ талабына сәйкес жазылған. Бұл дипломдық жоба жоғарғы оқу орындарының талаптарына сай жеткілікті жоғарғы дәрежеде жазылған, алынған нәтижелер – басқару блогын тиімді пайдаланудағы бағытқа жауап береді.

ЖҰМЫСТЫҢ БАҒАСЫ

Жалпы, дипломдық жұмысқа "өте жақсы" (90%) деген баға, ал студент Тажен Қалмұхамед Әбубәкірұлы 6B07112 – Electronic and Electrical Engineering білім беру бағдарламасының «техника және технологиялар бакалавры» дәрежесіне лайықты деп санаймын.

Рецензент:

ҚазҰАУ қауымдастырылған профессоры,



Қ.И. СӨТБАЕВ атындағы ҚАЗАҚ ҰЛТТЫҚ ТЕХНИКАЛЫҚ ЗЕРТТЕУ УНИВЕРСИТЕТІ
КОММЕРЦИЯЛЫҚ ЕМЕС АКЦИОНЕРЛІК ҚОҒАМЫ

ҒЫЛЫМИ ЖЕТЕКШІНІҢ ПІКІРІ

Дипломдық жұмыс

Тажен Қалмұхамед Әбубәкірұлы

6B07112 – Electronic and Electrical Engineering оқу бағдарламасы

Тақырыбы: «БЛИС негізінде көп арналы басқару блогын әзірлеу»

Студенттің БЛИС негізінде көп арналы басқару блогын әзірлеу жұмысы жоғары деңгейде орындалған. Жоба барысында студент заманауи технологияларды терең меңгергенін және инженерлік мәселелерді шешуде шығармашылық қабілеттерін көрсетті. Оның техникалық шешімдері нақты әрі тиімді, ал қолданылған әдістемелер инновациялық және практикалық маңыздылығын дәлелдеді. Бұл жұмыс тек теориялық білімі ғана емес, сонымен қатар практикалық дағдыларының да жоғары екенін көрсетеді. Студенттің осы жобасын болашақта одан әрі дамыту мүмкіндігі бар және оның кәсіби өсуіне үлкен ықпал ететініне сенімдімін.

Жалпы, дипломдық жұмысқа «өте жақсы» (90 %) деген баға қойылып, ал студент Тажен Қалмұхамед Әбубәкірұлы 6B07112 – Electronic and Electrical Engineering оқу бағдарламасы бойынша техника және технологиялар «бакалавры» академиялық дәрежесіне ұсынылады.

Ғылыми жетекші

Ғылыми жетекші
техн. ғыл. кандидаты,
аға оқытушы
Абдуллаев М. А.



«30» 05 2024 ж.

Протокол

о проверке на наличие неавторизованных заимствований (плагиата)

Автор: Тажен Қалмұхамед Әбубәкірұлы

Соавтор (если имеется):

Тип работы: Дипломная работа

Название работы: БЛИС негізінде көп арналы басқару блогын әзірлеу

Научный руководитель: Сұңғат Марқсұлы

Коэффициент Подобия 1: 6

Коэффициент Подобия 2: 1.8

Микропробелы: 5

Знаки из других алфавитов: 0

Интервалы: 0

Белые Знаки: 0

После проверки Отчета Подобия было сделано следующее заключение:

Заимствования, выявленные в работе, является законным и не является плагиатом. Уровень подобия не превышает допустимого предела. Таким образом работа независима и принимается.

Заимствование не является плагиатом, но превышено пороговое значение уровня подобия. Таким образом работа возвращается на доработку.

Выявлены заимствования и плагиат или преднамеренные текстовые искажения (манипуляции), как предполагаемые попытки укрытия плагиата, которые делают работу противоречащей требованиям приложения 5 приказа 595 МОН РК, закону об авторских и смежных правах РК, а также кодексу этики и процедурам. Таким образом работа не принимается.

Обоснование:

Дата 31.05.2024

Заведующий кафедрой



**Университеттің жүйе администраторы мен Академиялық мәселелер департаменті
директорының ұқсастық есебіне талдау хаттамасы**

Жүйе администраторы мен Академиялық мәселелер департаментінің директоры көрсетілген еңбекке қатысты дайындалған Плагиаттың алдын алу және анықтау жүйесінің толық ұқсастық есебімен танысқанын мәлімдейді:

Автор: Тажен Қалмұхамед Әбубәкірұлы

Тақырыбы: БЛИС негізінде көп арналы басқару блогын әзірлеу

Жетекшісі: Сұнғат Марксұлы

1-ұқсастық коэффициенті (30): 6

2-ұқсастық коэффициенті (5): 1.8

Дәйексөз (35): 0.9

Әріптерді ауыстыру: 0

Аралықтар: 0

Шағын кеңістіктер: 5

Ақ белгілер: 0

Ұқсастық есебін талдай отырып, Жүйе администраторы мен Академиялық мәселелер департаментінің директоры келесі шешімдерді мәлімдейді :

Ғылыми еңбекте табылған ұқсастықтар плагиат болып есептелмейді. Осыған байланысты жұмыс өз бетінше жазылған болып санала отырып, қорғауға жіберіледі.

Осы жұмыстағы ұқсастықтар плагиат болып есептелмейді, бірақ олардың шамадан тыс көптігі еңбектің құндылығына және автордың ғылыми жұмысты өзі жазғанына қатысты күмән тудырады. Осыған байланысты ұқсастықтарды шектеу мақсатында жұмыс қайта өңдеуге жіберілсін.

Еңбекте анықталған ұқсастықтар жосықсыз және плагиаттың белгілері болып саналады немесе мәтіндері қасақана бұрмаланып плагиат белгілері жасырылған. Осыған байланысты жұмыс қорғауға жіберілмейді.

Негіздеме:

Күні 31.08.2024

Кафедра меңгерушісі



Протокол

о проверке на наличие неавторизованных заимствований (плагиата)

Автор: Тажен Қалмұхамед Әбубәкірұлы

Соавтор (если имеется):

Тип работы: Дипломная работа

Название работы: БЛИС негізінде көп арналы басқару блогын әзірлеу

Научный руководитель: Сұңғат Марқсұлы

Коэффициент Подобия 1: 6

Коэффициент Подобия 2: 1.8

Микропробелы: 5

Знаки из других алфавитов: 0

Интервалы: 0

Белые Знаки: 0

После проверки Отчета Подобия было сделано следующее заключение:

Заимствования, выявленные в работе, является законным и не является плагиатом. Уровень подобия не превышает допустимого предела. Таким образом работа независима и принимается.

Заимствование не является плагиатом, но превышено пороговое значение уровня подобия. Таким образом работа возвращается на доработку.

Выявлены заимствования и плагиат или преднамеренные текстовые искажения (манипуляции), как предполагаемые попытки укрытия плагиата, которые делают работу противоречащей требованиям приложения 5 приказа 595 МОН РК, закону об авторских и смежных правах РК, а также кодексу этики и процедурам. Таким образом работа не принимается.

Обоснование:

Дата 31.05.2024.

 Марқсұлы С
проверяющий эксперт