**Міністерство освіти і науки україни**

**КПІ ІМ. ІГОРЯ СІКОРСЬКОГО**

**ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ**

**кафедра конструЮВАННЯ**

**ЕЛЕКТРОнно-ОБЧИСЛЮВАЛЬНОЇ АПАРАТУРИ**

**Розрахункова Графічна РОБОТА**

з кредитного модуля «Обчислювальні та мікропроцесорні засоби в радіоелектронній апаратурі - 1»

на тему «Передача власних даних протоколом SPI»

|  |  |
| --- | --- |
| Керівник: Антонюк Олександр Ігорович | Виконав: |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | студент гр. ДК-82 |
| Захищено з оцінкою \_\_\_\_\_\_\_\_\_\_\_\_\_ | Дмитрук Олександр |
| Дата «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_20220 р. | Залікова книжка № ДК- |

Київ – 2020

**Зміст**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

1

*ДК82.460100.001ПЗ*

Розробив

Дмитрук О.О.

Перевірив

Реценз.

Н. Контр.

Затвердив

Передача власних даних протоколом SPI

Літ.

Аркушів

12

КПІ ім. Ігоря Сікорського

Технічне завдання……….………………………………………….……….….2

Вступ………….……………………….……….………………………………..3

1. Деякі відомості про інтерфейс SPI..……….………………………..……..….4
2. Протокол роботи пристрою……………………………………...………….....6
3. Розробка SLAVE пристрою…………..…….……………………………….…8
4. Розробка MASTER пристрою………………………………………………….9
5. Тестування роботи пристроїв……………..………………………………....11

Висновки…………………………………………………………………….....12

**Технічне завдання**

Зм.

Арк

№ докум.

Підпис

Дата

Арк

2

*ДК82.460100.001ПЗ*

*ПЗ*

1. **Підстава для розробки**

Основою для розробки є завдання на розрахункову роботу, яке видане

«17» листопада 2020р.

1. **Найменування і область призначення**

Передача власних даних протоколом SPI. Застосовується для передачі ASCII слів збережених в SLAVE пристрої MASTER пристрою.

1. **Мета і призначення розробки**

Розробити MASTER та SLAVE пристрій. SLAVE пристрій буде отримувати адресу початку слова від MASTER пристрою. SLAVE пристрій буде передавати слово для MASTER пристрою.

1. **Джерело розробки**

Курс лекцій з кредитного модуля “Обчислювальні та мікропроцесорні засоби в радіоелектронній апаратурі - 1” Моделювання цифрових пристроїв, базові конструкції мови опису апаратури Verilog та способи опису різноманітних цифрових пристроїв.

1. **Технічне завдання** 
   1. **Функціональні можливості**

Slave пристрій повинен отримувати адресу початку слова від MASTER пристрою, і видати на лінію пакет ASCII слів.

* 1. **Технічні характеристики**

– Швидкість передачі протоколу SPI 9600 бод.

– Розмір одного пакету даних 8 біт.

**Вступ**

Зм.

Арк

№ докум.

Підпис

Дата

Арк

3

*ДК82.460100.001ПЗ*

В кожній цифровій апаратурі, мікроконтролери, різні цифрові мікросхеми, мікропроцесори, тощо повинні обмінюватись даними, передавати данні в пам’ять або зчитувати данні з неї. Для цього створюють різні протоколи передачі даних такі як SPI (serial peripheral interface) – послідовний периферійний інтерфейс, працюючий в режимі повного дуплексу, є дуже швидким , синхронним та простим в реалізації. Використовується в цифровій схемотехніці якщо необхідна швидкий обмін даними між пристроями. I2C (Inter – integrated circuit) послідовна асиметрична шина для зв’язку між інтегральними схемами всередині електричних пристроїв. Даний інтерфейс не таки швидкий як інтерфейс SPI, але має лише 2 лінії зв’язку (SDA і SCL) а не 4, як в SPI. I2C інтерфейс використовується для підключення низькошвидкістних периферійних компонентів з процесорами та мікроконтролерами (наприклад на материнських платах, в мобільних телефонах, передачі даних для виводу їх на дисплей LCD екранів, тощо). В даній розрахунковій роботі буде використаний саме SPI інтерфейс, так як він має значну перевагу в швидкості обміну даними порівняно з I2C, особливо це потрібно в передачі ASCII слів, де для передачі 25 ASCII слів потрібно передати аж 25 пакетів по 8 байт. У випадку якщо потрібно передати значну кількість слів, це може спричинити до значних затримок, використовуючи низькошвидкістні інтерфейси передані даних.

**Деякі відомості про інтерфейс SPI**

SPI (Serial peripheral interface) – як згадувалось раніше, послідовний периферійний інтерфейс, для синхронної швидкої передачі даних між декількома пристроями, з дешевою та простою реалізацією. Даний інтерфейс зазвичай використовують для підключення одного головного пристрою (master device) до одного або декількох периферійних пристроїв (slave device), але й можлива реалізації SPI інтерфейсу з декількома master пристроями, але це складно реалізувати. В інтерфейсі SPI є 4 лінії зв’язку. А саме:   
1. MISO (master input, slave output) – лінія зв’язку, по які передаються дані з Slave пристрою до Master пристрою.

Зм.

Арк

№ докум.

Підпис

Дата

Арк

4

*ДК82.460100.001ПЗ*

2. MOSI (master output, slave input) лінія зв’язку, по які передаються дані з Master пристрою до Slave пристрою.

3. SS (slave select) – лінія зв’язку, по які Master виставляє активний логічний рівень для конкретного Slave пристрою, кажучи йому що він хоче працювати саме з ним (активує Slave пристрій).

4 SCK – лінія на якій формується тактовий сигнал синхронізації передачі, який створює Master пристрій.

Спрощена схема передачі даних по інтерфейсу SPI наведена на рисунку 1.1.

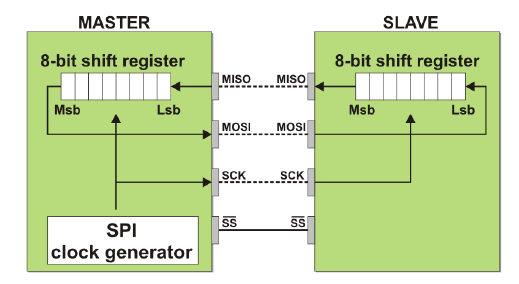


рис.1.1. Спрощена схема передачі даних по інтерфейсу SPI

Як бачимо Master і Slave утворили кільцевий регістр. Master має вбудований генератор тактового сигналу SCK. Як тільки прийде активний фронт тактового сигналу SCK на вхід SCK Slave пристрою і буде активний логічний рівень на вході SS ( На рисунку 1.1. SS з активним низьким рівнем) , старший біт здвигового регістру що находиться в Master пристрої запишеться в молодший біт здвигового регістру Slave пристрою. Якщо пристрої налаштовані на обмін пакетами по 8 біт, то після формування 8 імпульсів тактового сигналу SCK пакет з 8 біт передасться з Master пристрою до Slave пристрою, і 8 біт з Slave в Master. Таким чином працює повно дуплексний інтерфейс SPI. Можна реалізувати окремо регістри для Miso та Mosi лінії як в Master так і в Slave пристрої, за бажанням, якщо так буде зручніше в реалізації пристроїв. Інтерфейс SPI працює на визначених швидкостях передачі даних, узгодженими інженерами. В даній розрахунковій роботі швидкість обміну даними вирішено обрати 9600 бод .

Зм.

Арк

№ докум.

Підпис

Дата

Арк

5

*ДК82.460100.001ПЗ*

**Протокол роботи пристрою**

Розглянемо принцип роботи пристрою що розробляється. Даний пристрій що розробляється являється по своїй природі пам’ятю в якій будуть збережені ASCII слова. В кожній комірці ROM пам’яті, що находиться в Slave пристрої буде зберігатись 1 ASCII слово збережене в 16-бітному форматі. Для збереження такого слова потрібно 8 біт (для кожного символу зарезервований свій двійковий код. Діапазон даних кодів лежить в межах 0 – 255 десяткових чисел) Тобто для того щоб передати всі можливі ASCII слова нам необхідно передавати слова 8 бітними пакетами. Було вирішено для кожного повного слова (набору ASCII символів) резервувати 25 байт пам’яті – тобто максимальна довжина повного слова становить 25 символів – не більше. Так як вирішено обрати передачу пакету рівню 8 бітам, Master зможе звернутись лише до (128/25) 5 повних слів в ROM пам’яті. За необхідності більшої кількості слів, збільшуємо бітність пакету. Основним \*серцем\* як Master так і Slave пристрою будуть кінцеві автомати.

Зм.

Арк

№ докум.

Підпис

Дата

Арк

6

*ДК82.460100.001ПЗ*

У випадку Slave пристрою кінцевий автомат буде влаштований наступним чином:

1. Приходить пакет від Master пристрою з адресою пам’яті. Автомат обробляє даний пакет(за умови що ~SS = 0) протягом 1 такту SCK і записує з ROM пам’яті в здвиговий регістр лінії MISO 1 ASCII слово (8 біт).
2. Автомат виставляє дозвіл зсуву регістру зсуву, і по приходу активного фронту сигналу SCK починається зсув регістру, тобто починається транзакція даних до Master пристрою.
3. Як тільки прийде 8 імпульсів, за умови якщо Master не скинув сигнал ~SS, Slave пристрій запише наступне ASCII слово, (всі символи записані в ROM пам’яті послідовно) в здвиговий регістр лінії MISO, і буде чекати тактового сигналу SCK.
4. Транзакція ASCII слів буде виконуватись доки Master не дизактивує SLAVE пристрій. Як тільки відбудеться дизактивація, Slave пристрій скинеться до початкового стану і виставить вихід MISO в високоімпедансний стан.

Зм.

Арк

№ докум.

Підпис

Дата

Арк

7

*ДК82.460100.001ПЗ*

У випадку Master пристрою кінцевий автомат буде влаштований наступним чином:

1. Автомат дозволить запис адреси пам’яті в здвиговий регістр лінії MOSI та активує Slave пристрій.
2. Автомат дозволить формувати сигнал SCK.
3. Через 8 імпульсів SCK прийде адреса на SLAVE пристрій, автомат ще чекає 8 імпульсів SCK щоб отримати пакет даних.
4. Як тільки приходить повний пакет (1 ASCII слово), автомат перевіряє його, і у випадку якщо він рівний коду закінчення слова, дизактивує Slave пристрій. А якщо не рівний даному коду, продовжує формувати сигнал SCK , і через кожні 8 імпульсів отримувати новий пакет даних.
5. Було вирішено реалізувати автомат таким чином, що у випадку, коли прийшов пакет з кодом закінчення слова, автомат почне формувати неактив. лог. рівень на лінії ~SS деякий час, і скидатись в початковий стан. Тобто почнеться цикл спочатку, запишеться нова адреса пам’яті. Таким чином буде продемонстровано можливість звернення до різних адрес пам’яті.

**Розробка slave пристрою**

Як було згадано раніше , в основі slave пристрою лежить ROM пам’ять , в якій зберігаються ASCII символи , записані в 16-бітному відображенні та кінцевий автомат який буде керувати всім Slave пристроєм.

Зм.

Арк

№ докум.

Підпис

Дата

Арк

8

*ДК82.460100.001ПЗ*

Всередені Slave-у має бути присутній регістр зсуву розрядністю 8 біт для лінії Mosi, щоб отримувати адресу, по якій лежать данні які хоче отримать Master. Зсув даного регістру буде відбув. по приходу актив. фронту сигн. синх. SCK і актив. лог. рівню на лініх ~SS.

В пристрої присутні такі схеми як Double flopping-и щоб синхронізувати входи на лініях MOSI.

Також присутні детектор фронту сигналу SCK та CS щоб детектувати приход переднього фронту. Детектор буде формувати імпульс тривалістю 1 такт. сигнал CLK.

Як тільки прийде повний пакет адреси слова від Master пристрою (в пристрої є лічильник count1 котрий рахує 8 імпульсів сигналу SCK) , автомат дозволить запис з здвигового регістру лінії MOSI в паралельний регістр par\_reg1. Це потрібно для того щоб зберегти адресу, і в подальшому інкрементувати її для передачі наступного слова. Вихід даного регістру заводиться на адресний вхід ROM пам’яті.

Перейшовши в наступний стан автомат дозволить запис ASCII слова в регістр зсуву shift\_reg\_miso.

Далі автомат перейшовши в настуний стан дозволяє зсув даного регістру. По кожному актив. фронті SCK (для детектування у нас є детектор фронту sck\_detect) передаватиметься біт на лінію MISO Master пристрою.

Після передачі 8 імпульсів (лічильник рахує 8 імпульсів), відбувається інкрементування вмісту par\_reg1,, і по наступному актив. фронті CLK автомат ввертається в стан запису слова в здвиговий регістр shift\_reg\_miso.

У випадку якщо Slave передав пакет з нулями, (даний код вирішили взяти як код закінчення слова) , Master виставить ~SS = 1, що дезактивує Slave пристрій, автомат скинеться в початковий стан, вміст всіх регістрів скинуться.

Зм.

Арк

№ докум.

Підпис

Дата

Арк

9

*ДК82.460100.001ПЗ*

**Розробка master пристрою**

В основі master пристрою також лежить кінцевий автомат який буде керувати master пристроєм (в даному випадку реалізація master пристрою потрібно для відображення роботи slave пристрою. В ролі Master -пристрою може виступати мікроконтролер і т.д.)

В master пристрої реалізований генератор сигналу SCK. Він побудований на базі лічильнику, який буде ділити частоту сигналу CLK (у нашому випадку CLK = 50мГц) на число рівне 5208 щоб сформувати сигнал SCK з частотою 9600 Гц. В лічильника є вхід дозволу рахунку, у випадку якщо необхідно припинити формувати сигнал SCK і вхід скидання, щоб скинути його.

Також в Master реалізована регістрова пам’ять “data\_reg”, щоб зберігати в ній передані ASCII слова від Slave пристрою.

Автомат Master-у працює наступним чином. В початковому стані він дозволяє запис в здвиговий регістр shift\_reg\_mosi адреси слова (адреси збережені в ROM пам’яті “addr\_reg”) і формує сигнал ~SS.

Далі він переходить в наступний стан, дозволяючий формувати сигнал SCK( дозволяє рахунок лічильнику sck\_cnt),дозволивши також здвиг регістру зсуву shift\_reg\_mosi, і чекає поки передасться 8 біт в Slave (лічильник cnt\_sck\_impulse рахує імпульси SCK).

Коли передались 8 біт адреси , автомат скидає лічильник cnt\_sck\_impulse , і продовжує формувати SCK , щоб отримувати пакет слів які збережені по переданій адресі.

Почекавши приходу пакету даних – 1 ASCII символу , автомат починає перевіряти прийдені дані з нулем, і у випадку якщо вміст не рівний нулю, дозволяє запис в пам'ять data\_reg вмісту здвигового регістру shift\_reg\_miso. Далі автомат переходить в стан ‘C’ , де він скидає лічильник , котрий рахує імпульси SCK і цикл починаєтся спочатку - рахунок 8 імпульсів , перевірка , запис.

Зм.

Арк

№ докум.

Підпис

Дата

Арк

10

*ДК82.460100.001ПЗ*

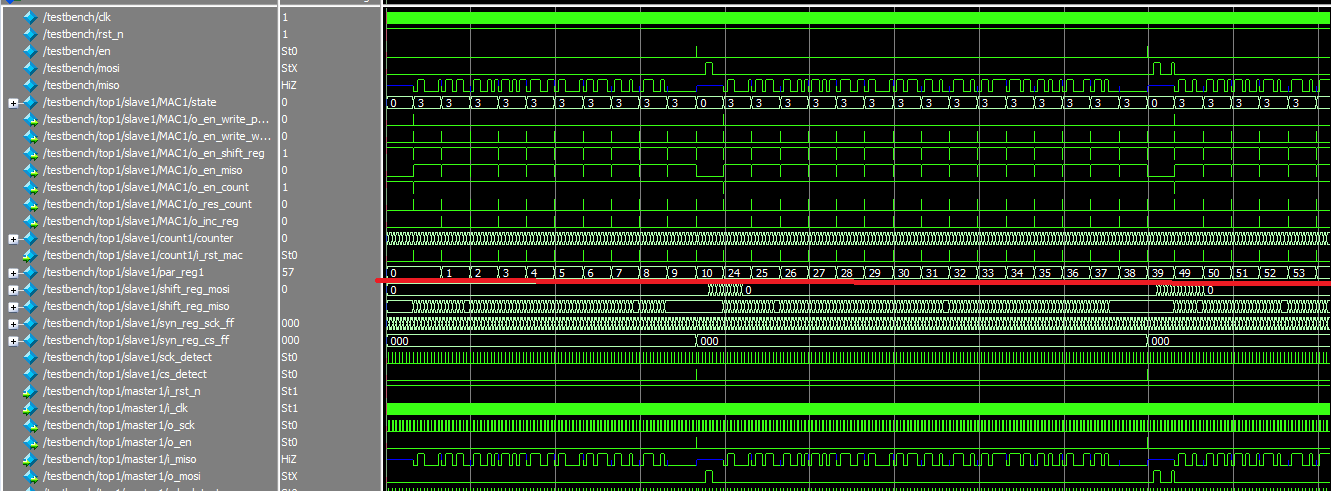
У випадку якщо прийшов нуль, автомат формує неактив. лог. рів. на лінію ~SS (лог. 1. ) витримуючи його декілька тактів сигналу CLK , щоб Slave пристрій точно скинувсь.

Далі автомат скидається до початкового стану + інкементувавши лічильник адреси, щоб записати згодом нову адресу в регістр зсуву shift\_reg\_mosi.

**Тестування роботи пристроїв**

Для тестування пристроїв необхідно створити топ файл, в якому будуть створені екземпляр модулю Master та Slave пристроїв.

Також потрібно створити тестбенч файл для даного топ файлу, в якому будуть генеруватись сигнали синхронізації для Master і Slave пристроїв і початкове скидання даних пристроїв. На рисунку 1.2. зображено доказ того що відбувається звернення до різних комірок ROM памяті. Спочатку Master звернувсь до нульової комірки. Транзакція припинилась коли Slave передав вміст 10 комірки. В 10 комірці зберігавсь код закінчення слова. Master обробивши його сформував ~SS = 1, скинувши Slave пристій. Після звернувсь до 24 комірки.



**Висновки**

Зм.

Арк

№ докум.

Підпис

Дата

Арк

11

*ДК82.460100.001ПЗ*

рис.1.2. вміст регістру par\_reg1

На рисунку 1.3. Зображений вміст регістрової пам’яті data\_reg який находиться в master пристрої. Ці дані передались з Slave до Master пристрою.

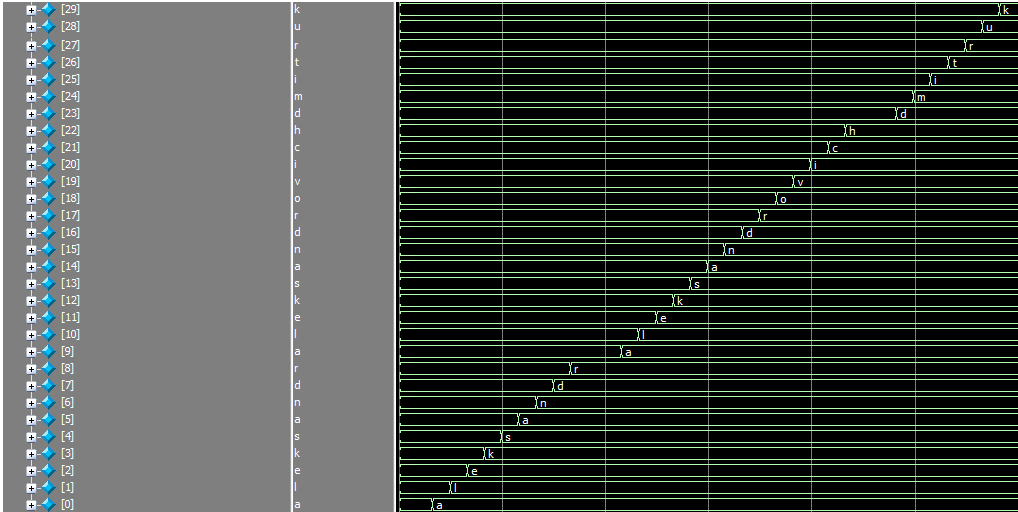


рис.1.3. Передані дані з Slave до Master пристрою

**Висновок**

Під час виконання розрахунково-графічної роботи було реалізовано Slave пристрій з ROM пам’ятю , в якому находяться ASCII слова збереженні в 16-бітному відображені. Було реалізовано інтерфейс передачі даних по протоколу SPI. Також була описана логіка Master пристрою, який формував адреси слів Slave пристрою. Slave в свою чергу передавав слова Master пристрою. Під час виконання даної роботи було розглянуто і вивчено багато матеріалу, робота різних протоколів обміну даних, покращено навички опису цифрової логіки на мові Verilog.

Зм.

Арк

№ докум.

Підпис

Дата

Арк

12

*ДК82.460100.001ПЗ*