השוואה בין סוגי תקשורת טוריות

א. כללי - תקשורת טורית ומקבילית

שתי שיטות השידור הנפוצות, להתחברות בין 2 נקודות, הן תקשורת מקבילית ותקשורת טורית. בתקשורת מקבילית משדרים במקביל על 8 חוטים נתון בגודל ביית – BYTE - כשבכל חוט יש את הערך של ביט מ D0 על החוט הראשון ועד D7 על החוט השמיני. בתקשורת טורית יש רק 2 חוטים ומשדרים את הביית ביט אחרי ביט. מכאן ברור שיתרון התקשורת המקבילית על הטורית הוא במהירות העברת הנתונים ואילו החיסרון בתקשורת המקבילית הוא מחיר (יותר חוטים).

תקשורת טורית היא תהליך של שליחת מידע של ביט אחד אחרי השני . ביט אחד בכל פרק זמן נתון. הביטים מועברים ברצף אחד אחרי השני עד שמסתיים הביית ולאחריו מתחיל שידור הביטים של הביית הבא וכך הלאה עד לסיום ההודעה הרצויה.

התקשורת הטורית משמשת במקרים שצריך תקשורת לטווחים ארוכים או ברשתות מחשבים שבהם עלות הכבלים וקשיי הסנכרון של תקשורת מקבילית אינם מעשיים וגורמים העדפה של תקשורת טורית. היום מאד נפוץ להעביר נתונים בצורה טורית אפילו במרחקים קצרים.

ישנם מספר סוגי תקשורת טורית נפוצים:

1. תקשורת טורית "רגילה" הנקראת גם UART – Universal Asynchronous Receiver Transmitter - משדר מקלט אסינכרוני אוניברסלי.
2. תקשורת SPI (Serial Peripheral Interface – ממשק טורי היקפי).
3. I2C - (Inter-Integrated Circuit - מעגל בין משולב)הנקראת גם i^2c וגם IIC .
4. I2S - Inter Ic Sound – שהוא תקשורת טורית לשמע דיגיטאלי.
5. One wire - תקשורת שתוכננה על ידי חברת דאלאס Dallas , הנותנת מהירות נמוכה (16.3 Kbps) . התקשורת היא על קו אחד . היא דומה ל I2C אבל בקצב נמוך יותר ובטווח גדול יותר. משמשת בדרך כלל לתקשורת עם רכיבים לא יקרים כמו טרמומטר דיגיטאלי ומכשירי מזג אוויר. כאשר בתקשורת יש מספר רכיבים המתחברים אל רכיב מסטר היא נקראת MicroLAN .

תחילה נסביר את התקשורת הטורית הרגילה ולאחר מכן נסקור בקצרה את 2 סוגי התקשורת I2C ו SPI ואז נשווה בין סוגי התקשורת הטורית . על תקשורת I2S ניתן לקרא בקישור :
http://arikporat.com/projects/I2S%20comunication.pdf

ב. תקשורת טורית "רגילה"

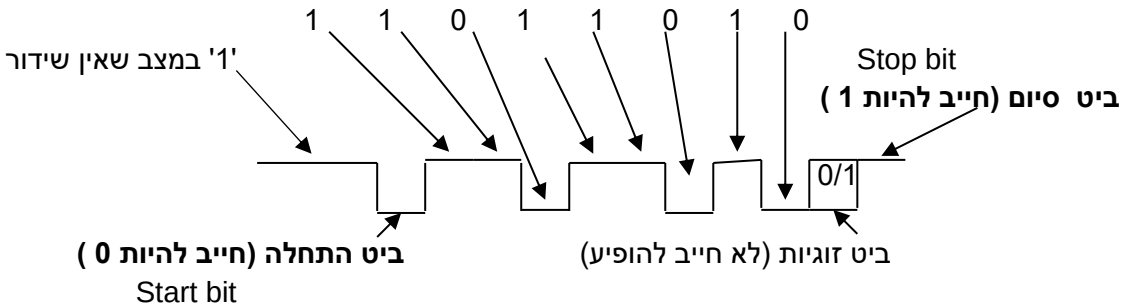
1. תקשורת טורית "רגילה" היא תקשורת שבה מתחברים רכיבים עם הדק TXD של צד אחד המתחבר להדק RXD של השני והדק TXD של השני מתחבר להדק RXD של הראשון

(חיבור כזה נקרא חיבור מוצלב). בתוך המיקרו בקר יש מעגל חומרה הנקרא UART – Universal A Synchronous Receiver Transmitter – משדר מקלט א-סינכרוני אוניברסלי המבצע את השידור והקליטה הטוריים.

2. בצורה כזו ה UART מבצע את פעולת השידור והקליטה של הנתונים והמיקרו פנוי לטפל בדברים נוספים ולא צריך לנהל את השידור או הקליטה הטורית.
3. התקשורת הטורית הרגילה נקראת גם RS232 אם כי שם זה לא מדויק. ה RS232 הוא תקשורת טורית "רגילה" אבל עם רמות מתח שונות (0 לוגי יכול להיות מתח של בין 7 וולט ל 15 וולט ו 1 לוגי יכול להיות מינוס 7וולט עד מינוס 15 וולט. כאשר רמת ה '0' גבוהה מרמת ה '1' - לוגיקה שלילית). מתחים אלו מאפשרים שידור לטווח גדול יותר מאשר 0 ו 5 וולט .
4. התקשורת הטורית עם UART היא תקשורת א-סינכרונית. כלומר אין שעון מערכת System Clock או שעון ראשי המסנכרן את התקשורת.
5. בתקשורת טורית כזו ניתן גם לשדר וגם לקלוט בו זמנית. זה נקרא **Full Duplex** .

כיצד שולחים נתון בקו תקשורת טורית ?

1. מערכת תקשורת טורית רגילה – UART - יודעת לקבל ביית מהמיקרו ולשדר אותו בהדק TxD של המיקרו, כמקובל בתקשורת טורית : קודם את **ביט ההתחלה – Start Bit** , אח"כ את סיביות הנתון ולבסוף את ביט הזוגיות - **PARITY** (אם הוחלט שעובדים עם זוגיות) ואת **ביט הסיום Stop Bit** . בסיום שידור התו ה UART מודיע למיקרו (גם בעזרת פסיקה וגם על ידי דגל) על סיום שידור התו. השידור מתבצע בקצב שהמתכנת קבע ל UART .
2. בקליטה טורית קורה תהליך הפוך. המיקרו מקבל בהדק RxD של המיקרו את האות הטורי. ה UART מזהה את ביט ההתחלה, את הביטים של הנתון , את הביט של הזוגיות (אם הוחלט לעבוד עם זוגיות) ואת ביט הסיום ומודיע למיקרו על ידי פסיקה וגם על ידי דגל על סיום קליטה טורית.
3. נניח שרוצים לשדר את התו 5BH כלומר 01011011 בינארי. התו משודר **מהביט הנמוך אל הגבוה (מה LSB אל ה MSB).**



איור מספר 1 : שידור תו טורי

4. כמות הביטים הנשלחת בשנייה נקראת **קצב התקשורת או קצב הביטים**. היחידות הן **סיביות לשנייה או סל"ש. באנגלית Bits Per Second – bps** . קצבי שידור מקובלים בתקשורת טורית : 4800, 9600 ו 19200 ביטים בשנייה. קיים גם מושג שנקרא **באוד – BAUD** , על שם מהנדס צרפתי שהמציא את קוד הטלטייפ בן 5 הביטים. המושג מתייחס למספר שינויי האות או הסמל המחוללים בשנייה אחת. סמל הוא אחד משינויי המתח. אצלנו יש ל 0 מתח של כ 0 וולט ול 1 יש מתח של כ 5 וולט ולכן קצב באוד וקצב הביטים שווה.

5. ביט הזוגיות הוא ביט שמציין את כמות ה '1' (האחדים) שיש בנתון. בתחילת ימי התקשורת הטורית היו משתמשים בביט הזוגיות כדי שהצד הקולט יידע האם הוא קלט את המידע נכון. קיימים 2 סוגי זוגיות : **א. זוגיות זוגית – Even Parity** **ב. זוגיות אי זוגית Odd Parity**. בזוגיות זוגית כמות ה '1' של הנתון פלוס ביט הזוגיות צריך להיות זוגי. בזוגיות אי זוגית כמות ה '1' בנתון פלוס ביט הזוגיות הוא אי זוגי. לדוגמא : נניח שמשדרים את התו 5bH ועובדים בזוגיות אי-זוגית. בתו יש 5 פעמים 1 . היות וכמות ה 1 איננה זוגית אז בביט הזוגיות נשלח '0' . הצד הקולט סופר את כמות ה 1 שקלט. היות והוא קלט 5 פעמים 1 הוא יודע שביט הזוגיות שהוא צריך לקלוט הוא '0'. אם אחד הביטים היה מתהפך בדרך אז הצד הקולט היה קולט מספר זוגי של 1 ואז היה רואה שביט הזוגיות הוא 0 והיה מבחין שקלט נתון שגוי. במקרה כזה הוא יכול לבקש מהצד המשדר לשלוח את הנתון פעם נוספת. הוספת ביט הזוגיות גורם לביט טורי תשיעי בנוסף ל 8 הביטים של המידע, דבר שמאט את קצב התקשורת הטורית, הנמוך ממילא. היום יש שיטות חדשות לאיתור שגיאות ואפילו לתיקון שגיאות.

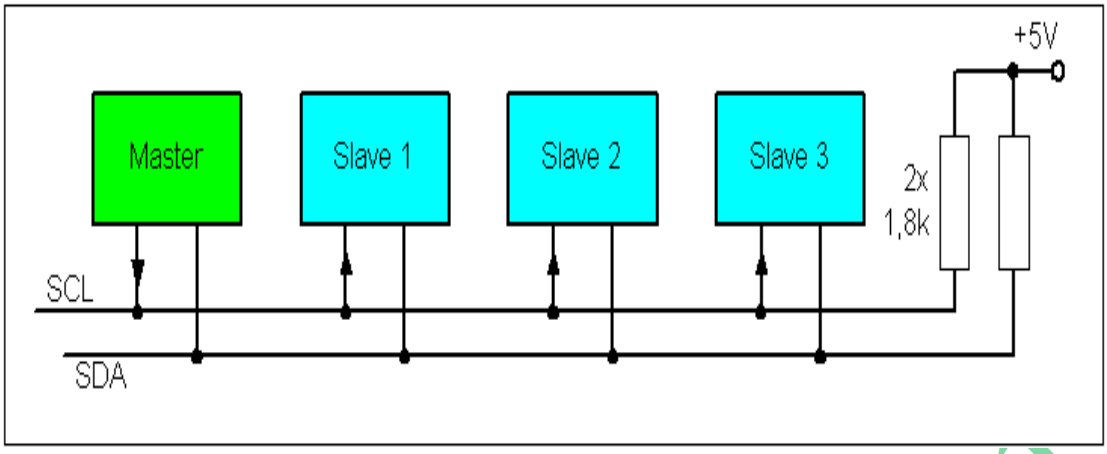
www.arikporat.com

נתאר בשני איורים את הארכיטקטורה של כל אחד מהפרוטוקולים:

ג. I2C ו SPI על קצה המזלג

ג.1 תקשורת I2C

התקשורת נוצרה על ידי חברת פיליפס ב 1982 (היום נקראת החברה NXP). הסבר מפורט על התקשורת ניתן למצוא בקישור : http://arikporat.com/projects.htm (לגלול עד למטה) . תקשורת זו היא **Half Duplex** כי לא ניתן לשדר ולקלוט בו זמנית.

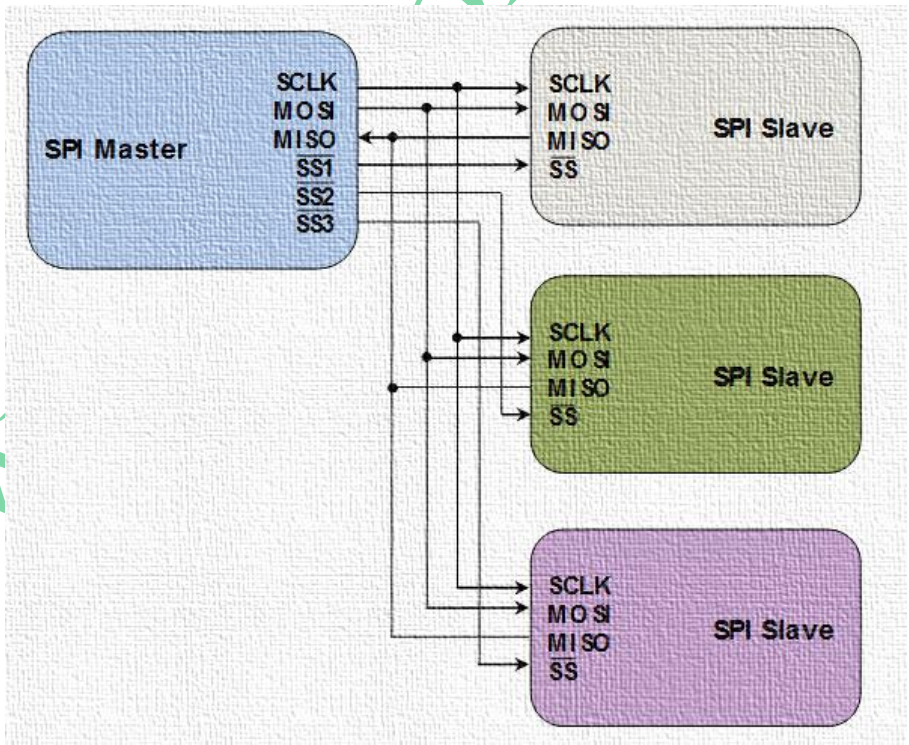


איור 2 : תקשורת I2C

ג.2 תקשורת SPI

התקשורת נוצרה על ידי חברת מוטורולה ב 1979 עם יציאת המיקרו פרוסור motorola 68000. הסבר מפורט על התקשורת ניתן למצוא בקישור :

http://arikporat.com/projects.htm (לגלול עד למטה)



איור 3 : תקשורת טורית SPI .

מתוך 2 האיורים רואים שבתקשורת I2C מחוברים שני נגדים ולעומת זאת כמות החוטים גבוהה יותר ב SPI . בהמשך נבצע השוואות נוספות.

ג.3 היסטוריה : תקשורת טורית - I2C מול SPI

התקשורת I2C שבאיור 2 פותחה בשנת 1982. המטרה המקורית שלה הייתה לספק דרך קלה לחיבור בין מעבד לרכיבים קלט/פלט פריפריאליים במערכות טלוויזיה. הפיתוח בוצע כי הרכיב הפריפריאלי חובר כזיכרון ממופה – memory mapped (כאילו הרכיב הוא כתובת בזיכרון) וזה אומר שהרכיב היה מתחבר במקביל על פס הכתובות , הנתונים והבקרה של הזיכרון. כתוצאה מכך היה נדרש חיווט רב במעגל המודפס ובנוסף גם מערכת פענוח שתזהה שפונים לרכיב קלט/פלט ולא לזיכרון וגם שתזהה לאיזה רכיב קלט/פלט . כדי לחסוך בהוצאות פיתחה חברת פיליפס הנמצאת בעיר איינדהובן שבהולנד את פרוטוקול ה I2C שבו רק 2 חוטים המחברים בין המיקרו בקר והרכיב הפריפריאלי. המפרט המקורי הגדיר מהירות של 100 קילו ביטים בשנייה . בהמשך המהירות שופרה בשנת 1995 ל 400 קילו ביטים בשנייה ובשנת 1998 ל 3.4 מיליון ביטים בשנייה עבור רכיבים פריפריאליים מהירים יותר. החברה איננה מחייבת בכסף עבור חיבור לתקשורת I2C אבל גובה כסף עבור קביעת כתובת I2C לרכיב חדש.

הפרוטוקול SPI הוצג לראשונה בשנת 1979 עם המיקרו בקר של מוטורולה שהיה מאד פופולארי בזמנו - 68000 . הפרוטוקול הגדיר חיבור של 4 קווים בין המיקרו בקר (נקרא מסטר) והרכיב הפריפריאלי (הנקרא עבד). קשה למצא דפי מפרט רשמי נפרדים של הפרוטוקול וניתן למצא אותו בדפי הנתונים של המיקרו בקר 68000 בפרק האפליקציות.

ד. העמקה ב 2סוגי התקשורת I2C ו SPI

ד.1 תקשורת SPI

התקשורת SPI שבאיור 3 היא פשוטה והגיונית יחסית ל I2C. בתקשורת SPI יש 4 קווים .

א. **MOSI – Master Out Slave In** - קו הנתון מהמסטר (המיקרו בקר) אל העבד.

ב. **MISO – Master In Slave Out** – קו הנתון הטורי מהעבד אל המסטר.

ג. שני אותות אלו מסונכרנים בעזרת קו השעון הטורי **SCLK** - Seial CLocK .

ד. קו נוסף נקרא SS – Slave Select – בחירת עבד. בעזרת קו זה המיקרו בקר מודיע לאיזה עבד הוא פונה. למעשה אלו מספר קווים כמספר הרכיבים הפריפריאליים. לכל רכיב עבד יתחבר קו מהמסטר. באיור 3 הם נקראים SS3 SS2 SS1 . הם מסומנים עם גג מעליהם לציין שהקו פעיל בנמוך Active Low. לדוגמה המסטר מוריד את הקו SS1 ל 0 ואז רק רכיב העבד הראשון (בצבע אפור) יודע שהמסטר מתקשר איתו. שאר העבדים יודעים שהמיקרו בקר איננו מתקשר איתם.

תקשורת **SPI** היא תקשורת ב **Full Duplex** כי ניתן לשדר ולקלוט בו זמנית.

www.arikporat.com

במערכת ה SPI (גם במסטר וגם בעבד) יש 2 רגיסטרים. האחד הוא רגיסטר הזזה המקבל את הדגימות ומזיז את הנתון ועוד רגיסטר נתון שבסיום ההעברה הנתון שהתקבל נמצא בו.

בפולס שעון יש 2 מעברים (מגבוה לנמוך ולהפך) הכוללים גם עלייה וגם ירידה ויש לעשות 2 דברים :

א. במעבר מ 0 ל 1 (או מ 1 ל 0 – יוסבר בהמשך) גם המסטר וגם העבד מוציאים את ביט הנתון לקו הנתון (MOSI במסטר MISO בעבד).

ב. במעבר השני מתבצעת הזזה של הנתון ברגיסטר ההזזה שנמצא גם במסטר וגם בעבד.

שני פרמטרים/מאפיינים חשובים בתקשורת SPI : א. **CPOL (Clock POLarity -** קוטביות השעון) הקובעת מה מצב הקו (נמוך או גבוה) במצב סרק - IDLE - כאשר אין פולסי שעון (לפני התחלת תקשורת ובסיומה) . כאשר CPOL=0 בקו יש '0' כאשר CPOL=1 בקו יש '1' .

ב. **CPHA (Clock PHAse** – פאזת השעון) הקובעת באיזה מעבר יוצא ביט הנתון בקו ה MOSI ובקו ה MISO ובאיזה מעבר הוא מוזז ברגיסטרי ההזזה גם במסטר וגם בעבד .

בעזרת 2 הפרמטרים אלו ניתן לקבוע אחד מ 4 אופני עבודה (מאופן 0 ועד אופן 3) שנתאר בהרחבה בהמשך.

ד.2 תבנית (פורמט) העברה של נתון עבור CPHA=0

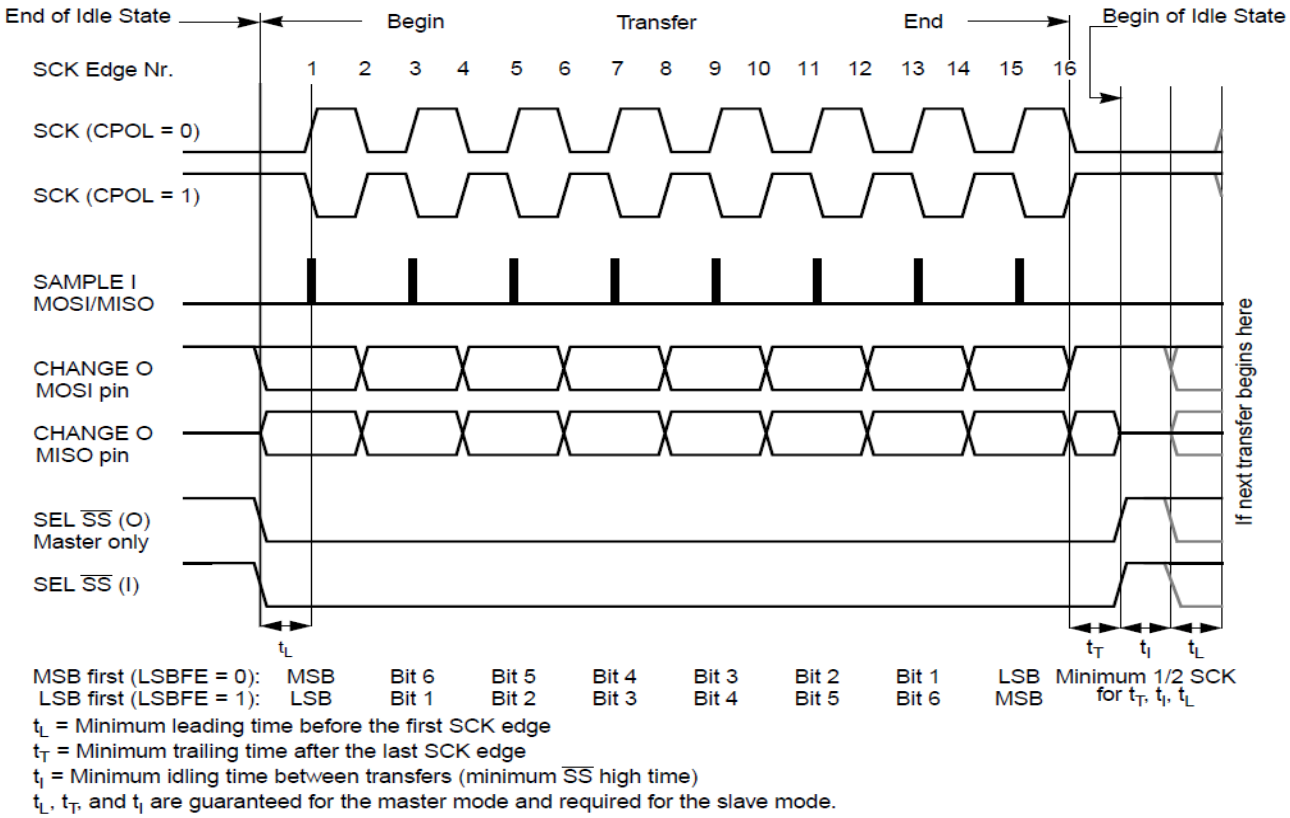
איור 4 שבהמשך מתאר את תבנית העברה עם צורות הגל של תקשורת SPI כאשר CPHA=0 . חלק התחתון רואים את התחלת התקשורת כאשר קו SS יורד ל 0 ואת סיומה כאשר הקו SS עולה ל 1 . הקו פעיל בנמוך ובחלק מהשרטוטים הוא נקרא NSS.

קו השעון SCK מתואר על ידי 2 צורות גלים . העליונה ביותר מתארת את פולסי השעון כאשר CPOL=0 ואז יש 0 בקו פולסי השעון והפולס הראשון הוא עלייה וכאשר אין תקשורת וצורת הגל שמתחתיה כאשר CPOL=1 ואז יש '1' בקו פולסי השעון והפולס הראשון הוא ירידה.

במקום הרשום באיור O הכוונה Output (יציאה) ובמקום שרשום I הכוונה Input (כניסה).

הביטים של הנתון הטורי בהדקי MOSI ו MISO נדגמים ומתבצעת הזזה ברגיסרי ההזזה באחת מ 2 אפשרויות. או בעליית השעון או בירידה לפי נתוני הרכיב שאליו מתחברים.

www.arikporat.com



איור 4 : צורות גל בתקשורת SPI כאשר CPHA=0.

המעבר הראשון של קו SCK (מ 0 ל 1 או מ 1 ל 0 תלוי ב CPOL) משמש להכנסת ביט הנתון הראשון של העבד אל המסטר (בקו MISO) ואת ביט הנתון הראשון של המסטר אל העבד (בקו ה MOSI) . באיור 4 במרכז רואים את הקו נקרא SAMPLE i והוא כניסה גם למסטר בקו ה MISO וגם לעבד בקו ה MOSI .

במעבר השני ,בחצי המחזור הבא , מופיע מעבר נוסף בקו SCK ואז הערך שנדגם מהעבד בקו MISO נכנס לתוך רגיסטר ההזזה שבמסטר. אחרי המעבר הזה משודר הביט הבא של המסטר על קו ה MOSI . התהליך נמשך עד 16 מעברים בקו SCK כאשר **נתון ננעל אל המסטר במעברים אי זוגיים ומועבר אל העבד במעברים זוגיים.**

אחרי מעבר מספר 16 הנתון שהיה ברגיסטר ה SPI של המסטר צריך להיות ברגיסטר הנתון של העבד והנתון שהיה ברגיסטר הנתון של העבד צריך להיות במסטר.

באיור מופיעים זמנים כמו :

t_L המראה מה הזמן המינימאלי מרגע הורדת קו SS ל 0 ועד להתחלת פולסי השעון.

t_T הוא הזמן המינימאלי בין פולס השעון האחרון והעלאת קו SS ל 1 .

T_I הוא זמן הסרק –IDLE - בין העלאת קו ה SS ל 1 ועד שידור נתון חדש על ידי הורדת SS ל 0 .

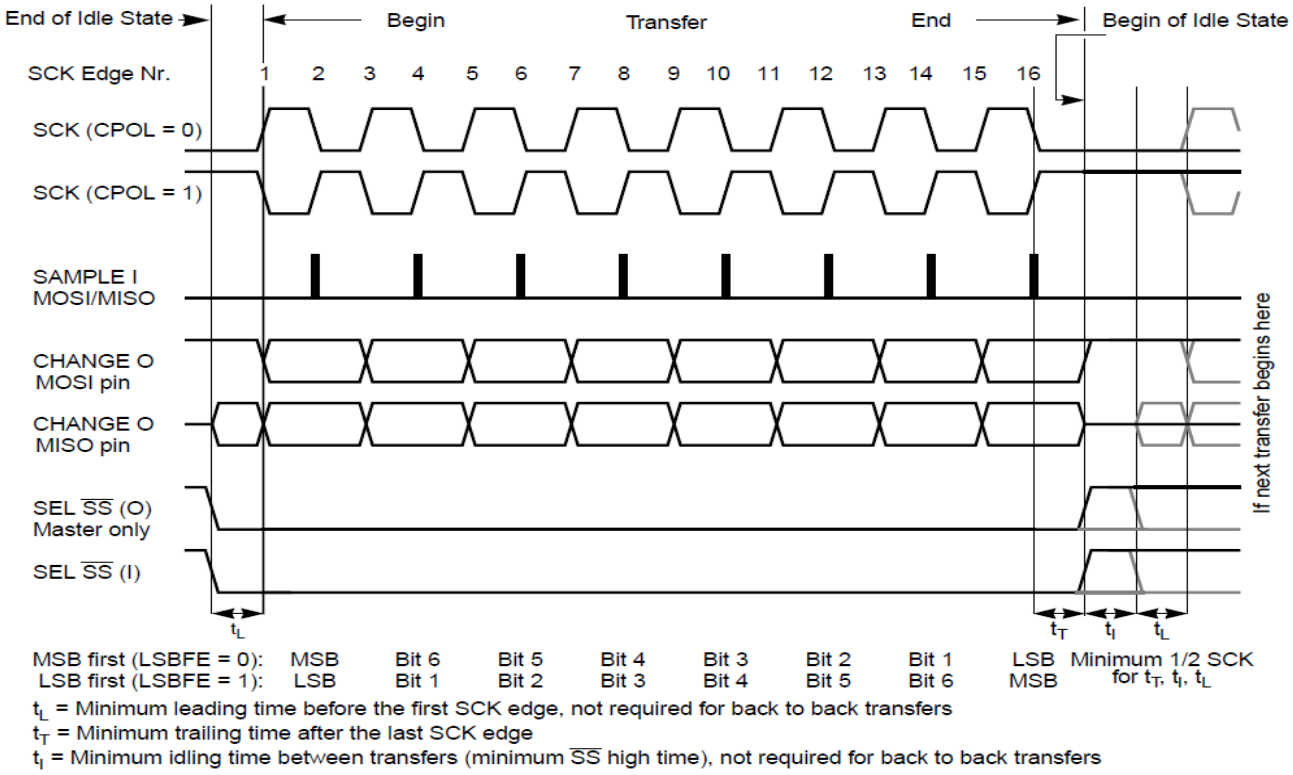
מתחת לצורות הגלים ניתן לראות שאפשר לשדר את הנתון מביט ה LSB אל ה MSB או להיפך.
קיים ביט LSBFE (LSB First Enable – אפשור LSB ראשון) שקובע מי נכנס ראשון . כאשר הוא 0 נכנס ביט ה MSB ראשון וכאשר הוא 1 נכנס ה LSB ראשון.

קבלת הנתון היא בשני שלבים. היא מוזזת לרגיסטר ההזה של ה SPI בזמן ההעברה ומועברת לרגיסטר הנתון של ה SPI כאשר הביט האחרון הוזז פנימה.

ד.3 תבנית (פורמט) העברה של נתון עבור CPHA=1

קיימים רכיבים שצריכים את המעבר של פולס השעון הראשון לפני שניתן יהיה לגשת לביט הנתון הראשון בקו היציאה של הנתון. המעבר השני מכניס את הנתון למערכת. פורמט זה מתקבל על ידי השמה של CPHA ל 1 .

איור 5 מתאר את התקשורת עבור CPHA=1 .



איור 5 : צורות גל בתקשורת SPI כאשר CPHA = 1 .

באיור רואים שהמעבר הראשון משמש כהשהיית סנכרון ואומר לעבד להוציא נתון בקו ה MISO . בחצי המחזור הבא, במעבר השני, יש נעילה של ביט הנתון גם במסטר וגם בעבד. כאשר מגיע המעבר השלישי מועבר הביט שנגעל במעבר השני לתוך ה LSB או ה MSB של רגיסטר ההזזה (כתלות בביט **LSBFE**) . אחרי מעבר זה יוצא ביט הנתון הבא של המסטר

בקו ה MOSI . התהליך חוזר על עצמו עבור כל 16 המעברים, כאשר הנתון ננעל במעברים הזוגיים וההזזה קוראת במעברים האי זוגיים.

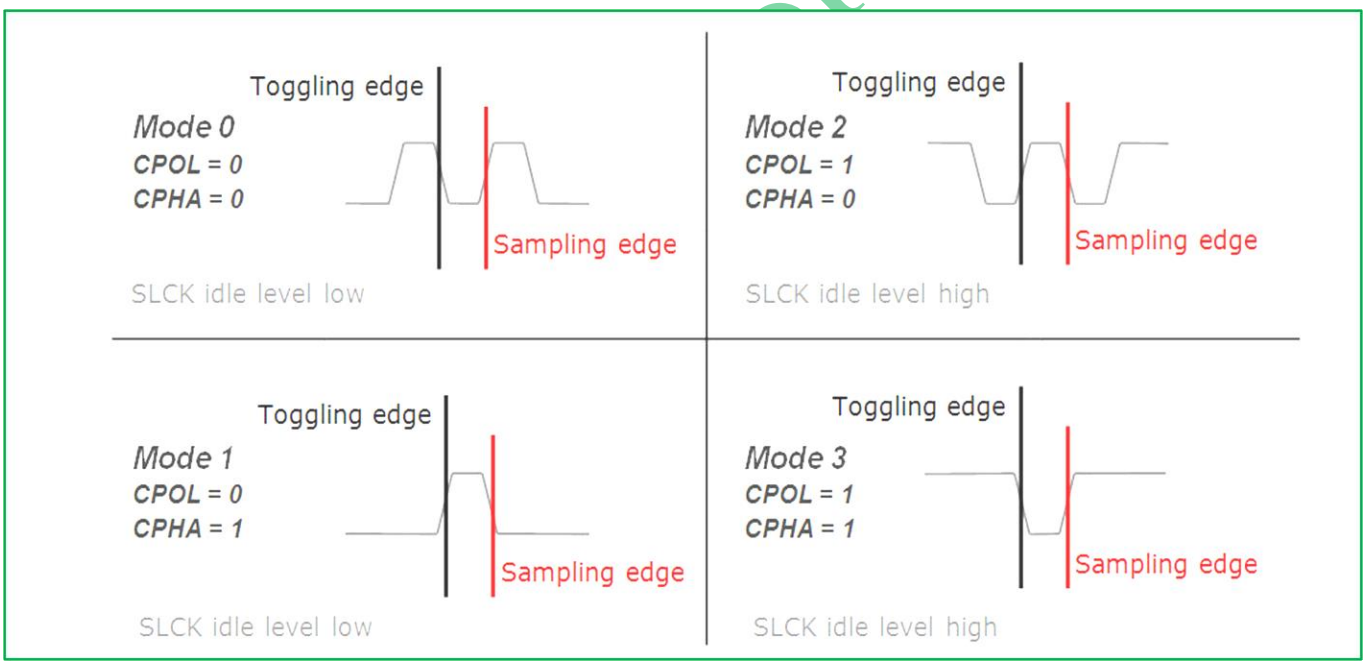
בדומה להעברה עבור CPHA=0 , גם כאן קבלת הנתון היא בשני שלבים. היא מוזזת לרגיסטר ההזה של ה SPI בזמן ההעברה ומועברת לרגיסטר הנתון של ה SPI כאשר הביט האחרון הוזז פנימה.

ד.4 - 4 אופני עבודה בתקשורת SPI

באיור 6 רואים את 4 אופני העבודה של התקשורת SPI . האופנים מוגדרים בעזרת הפרמטרים CPOL - Clock POLarity – קוטביות השעון ו CPHA - Clock PHAse – פאזת השעון. הם מגדירים 3 פרמטרים :

א. את המעבר (שפה EDGE) – של ביט הנתון אל קו הנתון (גם במסטר וגם בעבד) שמתואר על ידי הקו השחור בכל אחד מהאיורים.

ב. את המעבר של דגימת הנתון וההזזה בתוך רגיסטר ההזזה (גם במסטר וגם בעבד) המתוארת על ידי הקו האדום בכל אחד מהאיורים.



ג. את מצב השעון במצב סרק -IDLE - שבו אין תקשורת ולכן אין פולסי שעון.

איור 6 : 4 אופני העבודה בתקשורת SPI

אופן 0 : CPOL=CPHA=0

הוצאת ביט הנתון לקו הנתון (MOSI או MISO) מתבצע בירידת השעון. הדגימה וההזזה של הביט לתוך רגיסטר ההזזה ברכיב (גם במסטר וגם בעבד) מתבצעת בעליית השעון. כשאין תקשורת קו השעון ב 0 .

אופן 1 : CPOL=0 CPHA=1

ההעברת הנתון בעליית השעון והדגימה והזזה של הנתון ברגיסטר ההזזה - בירידת השעון. כשאין תקשורת הקו ב 0 .

אופן 2 : CPOL=1 CPHA=0

ההעברה בעליית שעון , דגימה והזז מתבצעת בירידת השעון. באין תקשורת הקו ב 1 .

אופן 3 : CPOL=1 COHA=1

ההעברה בירידת שעון , הדגימה וההזזה בעליית השעון. כשאין תקשורת הקו ב 1 .

צמד של אדון/עבד חייבים להשתמש באותם פרמטרים – תדר השעון , ה CPOL וה CPOH חייבים להיות זהים. אם יש מספר עבדים שהתצורה -קונפיגורציה - שלהם שונה (CPOL CPOH שונים), על המסטר להתאים את עצמו בכל פנייה לעבד לקונפיגורציה המתאימה.

תקשורת SPI איננה מגדירה קצב נתונים מרבי וגם לא עוסקת בכתובות כלשהן. אין ל SPI אפשרות לדעת האם המסטר או העבד קיבלו נתון כלומר, אין אישור קבלת נתונים מצד לצד. למעשה, למאסטר SPI אין שום ידע אם קיים עבד, אלא אם כן ' משהו ' נוסף נעשה מחוץ לפרוטוקול ה SPI (כמו בדיקה בתוכנה מה התקבל מהעבד ב NISO).

צמד של אדון/עבד חייבים להשתמש באותם פרמטרים – תדר השעון , ה CPOL וה CPOH חייבים להיות זהים. אם יש מספר עבדים שהתצורה -קונפיגורציה - שלהם שונה (CPOL CPOH שונים) , על המסטר להתאים עצמו בכל פנייה לעבד המתאים.

תקשורת SPI איננה מגדירה קצב נתונים מרבי וגם לא עוסקת בכתובות כלשהן. אין ל SPI אפשרות לדעת האם המסטר או העבד קיבלו נתון כלומר אין אישור קבלת נתונים מצד לצד. למעשה, למאסטר SPI אין שום ידע אם קיים עבד, אלא אם כן ' משהו ' נוסף נעשה מחוץ לפרוטוקול SPI

לתקשורת SPI לא אכפת ממאפייני הממשק הפיזי כמו מתח I/O ותקן המשמש בין המכשירים.

www.arikporat.com

ה.1 : תקשורת I²C

I²C - IIC - I2C הוא פרוטוקול תקשורת טורי סינכרוני המשתמש ב-2 קווים הנקראים :

1. **SDA** - **S**erial **DA**ta - נתונים טוריים (SDA)
2. **SCL** - **S**erial **CL**ock שעון טורי.

בתקשורת זו אין צורך בקו בחירת עבד - SS – Slave Select או מעגלי לוגיים לפענוח כתובת. כמעט כל מספר של עבדים וכל מספר של מאסטרים יכולים להיות מחוברים לשני קווי אות אלה ולתקשר האחד עם השני באמצעות פרוטוקול המגדיר:

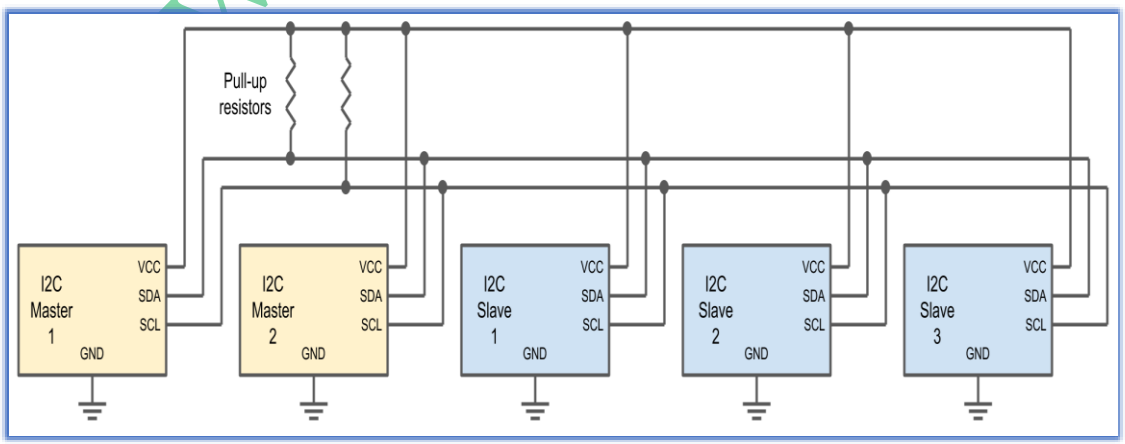
1. 7 ביטים עבור כתובות העבדים. לכל התקן המחובר לקווים יש כתובת ייחודית .
2. הנתונים מחולקים לבתים של 8 ביטים (סיביות).
3. ישנן כמה סיביות לשליטה כמו :
   - א. יצירת אות START התחלת התקשורת
   - ב. יצירת אות עצירה - STOP
   - ג. ביט כיוון הנתון (כתיבה או קריאה)
   - ד. יצירת אות אישור - acknowledge .

קצב העברת הנתונים צריך להיבחר בין :

1. מצב סטנדרטי : 100 kbps (קילו ביטים בשנייה)
2. מצב מהיר : 400 kbps
3. מהירות גבוהה : 3.4 Mbps (מיליון ביטים בשנייה).

ניתן למצא גם שמות תקפים כמו מהירות נמוכה של 10 kbps ומהירות גבוהה פלוס (fast+) של 1 Mbps .

באיור הבא מתוארים רכיבים המחוברים על היציאות של קווי I2C . באיור יש 2 מסטרים ו 3 עבדים.



איור 7 : רכיבים המחוברים על היציאות של קווי I2C

הקווים SDA ו SCL הם קווים בחיבור Open Collector (עבור טכנולוגיה מטרנזיסטורים) או ב Open Drain (עבור טכנולוגיית FET) והם מחוברים ל V_{CC} דרך 2 נגדי משיכה למעלה (Pull Up Resistors). במצב '0' ביציאה הטרנזיסטור (או טרנזיסטור ה FET ?) ברוויה ודרך נגד המשיכה המתאים ודרך הטרנזיסטור נסגר זרם לאדמה. מתח היציאה הוא כ 0.2 וולט בטרנזיסטור ואפילו פחות בטרנזיסטור FET . במצב של '1' הטרנזיסטור בקטעון ולא זורם דרכו זרם. נגד המשיכה המתאים מושך את הקו למתח 5 וולט (או מתח ספק רצוי שנחבר כ Vcc).

הייתרון הגדול ביציאות אלו הוא שאם מספר רכיבים שמים רמה לוגית שונה על אחד הקוים לא יינזק אף רכיב . ה '0' תמיד מנצח ובקו יהיה '0' ולא '1' .

פס התקשורת I2C מורכב מ-2 החוטים SDA ו-SCL וחיבור אדמה - GND. 2 הקווים הם דו-כיווניים. מפרט הפרוטוקול I2C קובע כי הרכיב היוזם העברת נתונים נחשב למאסטר האפיק. כתוצאה מכך, כל שאר הרכיבים נחשבים לעבדים . אנחנו בדרך כלל עובדים כשהמיקרו בקר הוא המסטר ושאר הרכיבים המחוברים אליו הם העבדים ומכאן שהוא יוצר את פולסי השעון בקו ה SCLK ורק קו ה SDA הוא דו כיווני.

ה.2 : ניהול התקשורת

ה.2.א מצב כתיבה

1. ראשית, המאסטר יוצר אות התחלה – START . הדבר משמש כאות ' תשומת לב ' לכל הרכיבים/העבדים המחוברים. כול הרכיבים נכנסים להקשבה להמשך הנתונים הנכנסים.
2. לאחר מכן, המאסטר שולח את הכתובת של ההתקן אליו הוא מעוניין לגשת, יחד עם אינדיקציה האם הגישה היא פעולת קריאה או כתיבה (כתיבה בדוגמה שלנו).
3. כל אחד מהעבדים משווה את הכתובת ששלח המסטר עם הכתובת שלו. אם הכתובת איננה מתאימה, הוא פשוט ממתין עד שקווי התקשורת "משתחררים" על ידי המסטר שיוצר אות סיום - STOP . העבד עם הכתובת התואמת, ייצור תגובה הנקראת אות אישור - acknowledge.
4. לאחר שהמאסטר מקבל את האישור, הוא יכול להתחיל לשדר או לקבל נתונים. במקרה שלנו, המסטר יעביר נתונים. כאשר כל הנתונים הועברו, המאסטר שולח את אות העצירה STOP. זהו סימן שקובע כי קווי התקשורת שוחררו וכי כל העבדים המחוברים עשויים לצפות ששידור אחר יתחיל בכל רגע.

www.arikporat.com

ה.2.ב מצב קריאה

1. כאשר מאסטר מעוניין לקבל נתונים מעבד, הוא יוצר אות START ולאחריו הוא שולח כתובת של 7 ביטים , אך שם בביט RD/$\overline{WR}$ אחד '1' לוגי שמראה שהוא רוצה לקרוא .
2. ברגע שהעבד המתאים מזהה את הכתובת שלו , הוא מתחיל לשלוח את הנתונים המבוקשים, ביית אחר ביית.
3. לאחר שידור של ביית נתונים, הוא מחכה למאסטר שיצור אות אישור acknowledge – שהוא קלט את הנתונים ורק אז העבד משדר את הביית הבא .

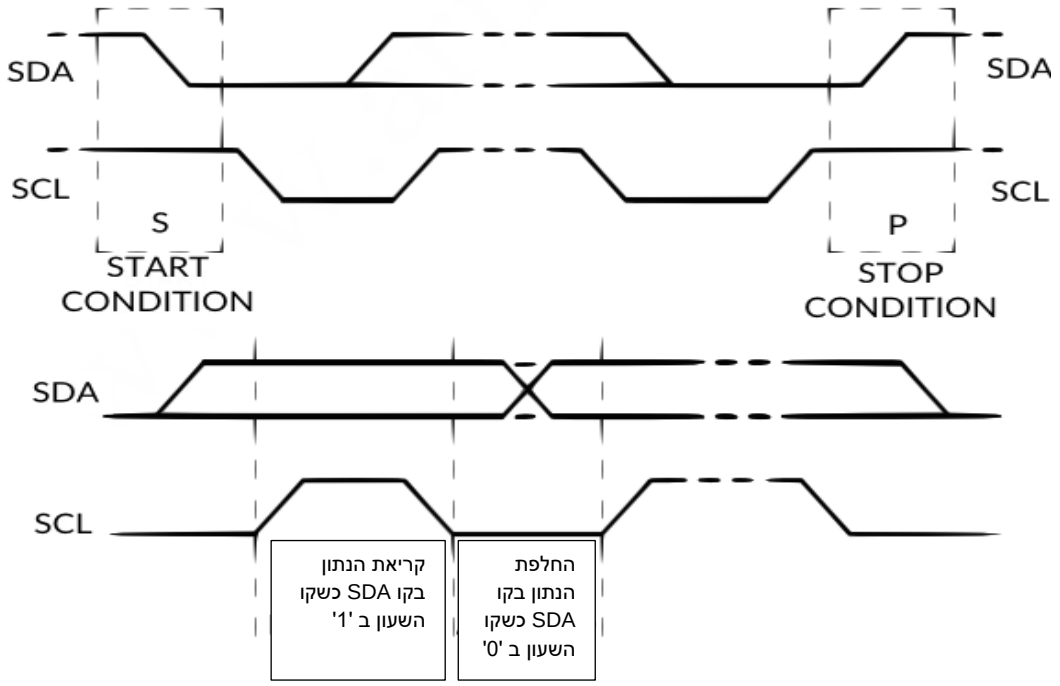
ה.2.ג מצבי פס ורמות מתח של התקשורת.

תחילת כל שידור בתקשורת I2C היא בעזרת אות START והסיום הוא בעזרת STOP או NOT ACKNOWLEDGE .

נתון יכול להשתנות בקו הנתון הטורי SDA רק כאשר אות השעון SCL ברמה נמוכה.

הנתון בקו SDA נחשב ליציב כאשר הקו SCL בגבוה.

באיור 8 ניתן לראות את אותות START ו STOP .



איור 8 : אותות START ו STOP (חלק עליון באיור) ומצב שינוי בקו הנתון SDA (בחלק התחתון).

מגדירים את מצבי הפס הבאים :

Bus Not Busy - פס לא עסוק

גם קו הנתון וגם קו השעון בגבוה.

START DATA TRANSFER - התחל העברת נתון

שינוי במצב קו הנתון מגבוה לנמוך כאשר השעון נמצא בגבוה מוגדר כמצב START .

STOP DATA TRANSFER - עצור העברת נתון

שינוי במצב קו הנתון מנמוך לגבוה כאשר השעון במצב גבוה מוגדר כמצב STOP .

מה קורה אם שני התקנים מנסים בו לשים מידע על שורות SDA ו/או SCL?
היישום הפיזי של הפס מאפשר , במידה ויש מספר מסטרים , לכתוב ולהאזין לקווים . בדרך זו, כל התקן מסוגל לזהות התנגשויות. במקרה של התנגשות בין שני מאסטרים (אחד מהם מנסה לכתוב אפס והשני אחד), המאסטר שמשיג את השליטה על הפס אפילו לא מודע שהיה מאבק: רק המאסטר כי לאבד ידע – שכן הוא מתכוון לכתוב את . אחד וקורא אפס היגיון כתוצאה מכך, המאסטר כי הבוררות בוררות על התואר הראשי של C יפסיק לנסות להיכנס לאוטובוס. ברוב המקרים, הוא רק יעכב את הגישה שלו וינסה את אותה גישה מאוחר יותר.

ה.2.ד : פנייה אל רכיבים עם כתובת של 10 ביט

לכל רכיב I2C יש כתובת בת 7 ביטים. באופן תאורטי ניתן לחבר 128 (2^7) רכיבי I2C שונים. היות ויש מספר רב של רכיבי I2C יכול להיות מצב שלשני רכיבים תהיה אותה כתובת. כדי להתגבר על המגבלה יש לרכיבים כמה כתובות והמהנדס יכול לבחור בעזרת תצורת הדקים חיצונית את הכתובת הרצויה. לדוגמה אם לשני רכיבים יש כתובת 40H אז ברגל חיצונית של הרכיבים הנקראת A0 ניתן לשים לרכיב אחד '0' ולרכיב השני '1' ואז הרכיב הראשון נמצא בכתובת 40H והשני בכתובת 41H . יש רכיבים עם מספר הדקים חיצוניים של כתובת כמו A0 A1 A2 וכו'. תקן I2C מאפשר גם מראש סכמת מיעון (כתובות) של 10 סיביות כדי להרחיב את הטווח של כתובת ההתקנים הזמינים.

איור 9 מתאר גישה לרכיב במיעון כתובות של 10 ביט. שני בתים משמשים ככתובת של הרכיב. ביית ראשון הוא קוד של 11110 ועוד 2 ביטים A9 A8 ועוד ביט המציין האם מדובר בכתיבה (0) או קריאה (1). הביית השני הוא החלק הנמוך של הכתובת. הקוד 11110 מציין שמדובר בכתובת של 10 ביט.

www.arikporat.com

START	1	1	1	1	0	A9	A8	Rd/ nWr	ACK	A7	A6	A5	A4	A3	A2	A1	A0	ACK	...
	First address word									Second address word									

10 bits address:

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

איור 9 : תקשורת עם מיעון כתובת של 10 ביטים

הביית הראשון הוא החלק הגבוה של הכתובת כאשר שולחים בביטים הגבוהים את 11110 ולאחריהם את הביטים A8 A9 שהם ביטים של כתובת. הביית השני הוא הביטים מ A0 עד A7 . בצורה כזו אפשרנו חיבור של 2^{10} = 1024 רכיבים שונים.

למעשה, ישנם קודי כתובות שמורים אחרים עבור סוגים מסוימים של גישה (ראה בטבלה 1).
לא נרחיב על הטבלה כי היא פחות שימושית . לפרטים נוספים, נא לעיין במפרט ה-I ² C.

#	Address	Purpose
1	0000000 0	General Call – addresses all devices supporting the general call mode
2	0000000 1	Start Byte
3	0000001 X	CBUS addresses
4	0000010 X	Reserved for different bus formats
5	0000011 X	Reserved for future purpose
6	00001XX X	High-speed Master code
7	11110XX X	10-bits slave addressing
8	11111XX X	Reserved for future purposes

טבלה 1 : קודי כתובות שמורים אחרים עבור סוגים מסוימים של גישה

ה.2.ה : הרחבת פולסי השעון

בתקשורת של I ² C ההתקן הראשי – המסטר - קובע את מהירות השעון. האות SCL הוא אות שעון המסנכרן את התקשורת.

עם זאת, ישנם מצבים בהם העבד (Slave) איטי ולא יכול לעבוד עם מהירות השעון שנקבעת על ידי המאסטר ולכן צריך להאט קצת את קצב השעון. ההאטה מתבצעת על ידי מנגנון המכונה מתיחת השעון - Clock Stretching - והוא מתאפשר על ידי המבנה של

www.arikporat.com

Open Drain או Open Collector של הפס. לעבד מותר להחזיק את קו השעון במצב נמוך '0' כדי להקטין את מהירות הפס. המאסטר , לאחר שהעלה את קו השעון ל '1' נדרש לקרוא את המתח של קו השעון ולחכות עד שהעבד "ישחרר" את הקו לגבוה. בצורה כזו העבד האט את מהירות התקשורת כדי שיוכל לקלוט את הנתונים בצורה אמינה.

ה.2.ו עבודה באופן מהירות גבוהה

השימוש בנגדי המשיכה למעלה מגבילה את המהירות המרבית של הפס כי הנגדים קובעים ביחד עם קיבול היציאה של הטרנזיסטור את קבוע הזמן טאו Ŧ = R*Cout . זו הסיבה מדוע פותחה המהירות Mbps 3.4 - מצב מהירות גבוהה הוצג. לפני שימוש במצב זה, המסטר שולח במהירות נמוכה את הקוד High Speed Master (מצב 6 בטבלה 1) המפעיל את התקשורת ב-3.4 Mbps ואז הוא יכול לעבור לעבוד במהירות זו. ברכיבי I2C משתמשים בחוצצי I/O מיוחדים כדי לקצר את זמני העלייה (במעבר מ '0' ל '1') ולהגדיל את מהירות הפס.

ו : מי המנצח ?

נבצע השוואה בין תקשורת I2C ו SPI מכמה אספקטים עקרוניים :

ו.1 – טופולוגיה/ניתוב/משאבים של הפס

I2C צריך 2 קווים בלבד, בעוד SPI באופן רשמי מגדיר לפחות 4 קווים ויותר (אם נוסיף עבדים נצטרך קו נוסף עבור כל עבד נוסף).

SPI דורש חומרה נוספת, לוגיקה ו/או פינים אם רוצים לעבוד עם כמה מסטרים .

הבעיה היחידה של I2C היא מקום מוגבל של 128 כתובות (7 סיביות) עם אפשרות להרחיב גם ל 10 סיביות. ועבודה ב Half Duplex לעומת SPI שהוא Full Duplex .

מי המנצח מבין 2 סוגי התקשורת ? אין מנצח ברור אם כי במקום שכמות ההדקים של המיקרו קטנה ברור שנבחר ב I2C במקום שצריך מהירות נבחר ב SPI.

ו.2 : תפוקה / מהירות

אם יש להעביר נתונים במהירות גבוהה - SPI הוא בבירור פרוטוקול הבחירה מעל I2C.

SPI הוא **דופלקס מלא** Full Duplex ואילו I2C **הוא Half Duplex**. SPI אינו מגדיר הגבלת מהירות כלשהי. היישומים לעתים קרובות גבוהים מ 10Mbps . I2C מוגבל ל-1Mbps במצב מהיר + ו - 3.4 Mbps במצב מהירות גבוהה .אבל מצב זה מחייב חוצצי

I/O מיוחדים. תקשורת I2C דורשת אישור ACKnoledge של הצד המקבל, דבר שגורם להאטה של מהירות העברת הנתונים.

3.ו אלגנטיות

לעתים קרובות אומרים כי I2C הוא הרבה יותר אלגנטי מאשר SPI וכי פרוטוקול SPI הוא 'פשוט מאוד' (ואולי 'טיפש'). הרבה מהנדסים נוטים לחשוב ששני הפרוטוקולים הם אלגנטיים באותה מידה ודומים באמינות .

I2C הוא אלגנטי משום שהוא מציע תכונות מתקדמות מאוד – כגון אין נזק בהתנגשויות בין מספר מסטרים (התנגשות הכוונה שמסטר אחד שם 0 ומסטר שני 1) והתגברות אוטומטית - הטיפול מובנה - על מצב זה (מחכים שהמסטר ששם 0 יעלה את הקו ל 1) . ניהול מיעון הכתובות – קל מאוד. אם כי במידת מה התקשורת חסרה ביצועים.

מצד שני ,תקשורת SPI קלה מאוד להבנה ויישום ומציעה הרבה גמישות עבור הרחבות ושינויים. פשטות היא המקום שבו האלגנטיות של SPI מנצחת. SPI צריך להיחשב כפלטפורמה טובה לבניית פרוטוקולים מותאמים אישית עבור תקשורת בין מעגלים משולבים (ג'וקים). כך שלפי הצורך של המהנדס/ת, שימוש ב-SPI עשוי לגרום לעבודה רבה יותר, אך נקבל ביצועים משופרים של העברת נתונים וחופש כמעט מוחלט.

גם SPI וגם -I2C מציעות תמיכה טובה לתקשורת עם רכיבים העובדים במהירות נמוכה, אבל SPI הוא מתאים יותר ליישומים בהם קיימת העברת נתונים , בעוד I2C טובה יותר בפעולה עם מספר מסטרים .

בשימוש נכון, שני הפרוטוקולים מציעים את אותה רמת החוסן – אמינות - והצליחו באותה מידה בקרב המשתמשים ויצרנים. רכיבים כמו EEPROM , ADC (ממיר אנלוגי לדיגיטלי), DAC (ממיר דיגיטלי לאנלוגי), RTC (שעון זמן אמת), מיקרו בקרים, חיישנים, LCD (תצוגת גביש נוזלי) מיקרו בקרים זמינים במידה רבה עם 2 הממשקים.

4.ו מסקנות

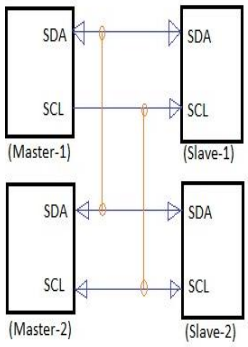
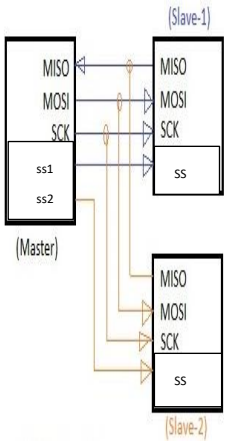
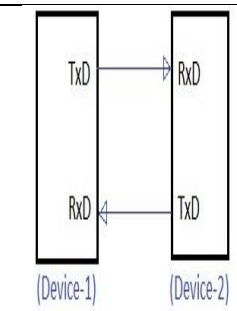
בעולם של פרוטוקולי תקשורת, I2C ו SPI נחשבים לעתים קרובות ' פרוטוקולים קטנים ' לעומת PCI-Express , SATA , USB , Ethernet ואחרים, כי התפוקה שלהם היא ממאות מגה ביטים לשנייה ועד gigabit לשנייה. למרות זאת, כדאי לזכור שפרוטוקולים Ethernet, USB, SATA מיועדים ' לתקשורת מחוץ לתיבה ' וחילופי נתונים בין מערכות שלמות. כאשר יש צורך ליישם תקשורת בין מעגלים משולבים כגון מיקרו-בקר וקבוצה של ציוד היקפי איטי יחסית, אין טעם להשתמש בפרוטוקולים מורכבים באופן מוגזם. שם, I2C ו SPI מתאימים

באופן מושלם והפכו למאוד פופולריים. סביר מאוד להניח כי כל מהנדסת/מהנדס מערכת ישתמשו בהם במהלך הקריירה שלהם.

המנצח : אין מנצח. "בגדול" ניתן לומר שכאשר מהירות היא חלק משמעותי בפרויקט נעבוד עם SPI . אם כמות ההדקים חשובה נעבוד עם I2C .

ז. השוואה בין סוגי התקשורת הטורית UART , SPI ו I2C

בטבלה מספר 2 הבאה נסכם את המשותף והמבדיל בין 3 סוגי התקשורת הטורית:

מס"ד	הנושא	I2C	SPI	UART
1	**השם**	**I**nter **I**ntegrated **C**ircuit בתוך מעגל משולב. התקשורת נקראת גם SMBus.	**S**erial **P**eripheral **I**nterface - ממשק היקפי טורי	**U**niversal **A**synchronous **R**eceiver **T**ransmitter – משדר מקלט א-סינכרוני אוניברסלי
2	**האם התקשורת סינכרונית או אסינכרונית**	סינכרונית	סינכרונית	אסינכרונית
3	**הממשק**			
4	**משמעות ההדקים**	SDA – **S**erial **D**ata נתון טורי SCL – **S**erial **CL**ock שעון טורי	Miso – **M**aster **I**n **S**lave **O**ut כניסת מסטר יציאת עבד	TxD – **T**ransmit **D**ata שידור נתון RxD -**R**eceive **D**ata

www.arikporat.com

			MOSI – **M**aster **O**ut **S**lave **I**n יציאת מסטר כניסת עבד SCK- Serial clock שעון טורי SS – **S**lave **S**elect בחירת עבד	קליטת נתון
5	**קצב העברת נתונים**	תמיכה בקצבי תקשורת של 100 קילו ביטים בשנייה -Kbps , 400 קילו ביטים בשנייה ו 3.4 מגה ביטים בשנייה. יש גם תמיכה ב 10 קילו ביטים בשנייה וב 1 מגה ביטים בשנייה.	אין ציון של מגבלת קצב העברת נתונים בממשק SPI. בדרך כלל תומכת בקצב של 10 מגה ביטים בשנייה – Mbps – ועד 20 מגה ביטים בשנייה.	בין 230 קילו ביטים בשנייה ועד 460 קילו ביטים בשנייה
6	**מרחק**	גדול יותר	הכי גדול	נמוך - 50 רגל (feet) שהם כ 15.24 מטר
7	**כמות המסטרים**	יכולים להיות כמה מסטרים	מסטר אחד	אין מצב עבודה של מסטר ועבד
8	**פולסי השעון**	כל מסטר יכול ליצור את השעון המרכזי.	יש שעון מרכזי אחד מהמסטר אל העבדים	אין שעון מרכזי. כל אחד מהרכיבים משתמש בשעון פנימי שלו.
9	**מורכבות החומרה**	2 קווים אבל יש להוסיף 2 נגדים חיצוניים. מורכבת מבחינת התוכנה.	לפחות 4 קווים (כשיש רק מסטר ועבד אחד) אבל פשוטה מבחינת תפיסה הגיונית.	פשוטה. 2 קווים בקליטה ושידור בו זמנית Full Duplex
10	**פרוטוקול**	יש שימוש באות התחלה - START ואות	לכל חברה או יצרן יש פרוטוקול SPI שלה	עבור נתון של 8 ביט משתמשים בדרך

www.arikporat.com

		עצירה STOP. יש שימוש באות אישור– Acknowledge אחרי כל שידור ביית שמציין שהנתון התקבל בצד הקולט.	ויש לקרא בדפי הנתונים של הרכיב.	כלל בביט התחלה ובביט סיום.
11	**שימוש במיעון (כתובות)**	יש אפשרות לכמה מסטרים וכמה עבדים כשכל מסטר יכול להתחבר לעד 128 עבדים באופן תאורטי (מעשי 112). (יש אפשרות להרחיב תאורטית עד 1024 עבדים).	ניתן לחבר כל מספר רצוי של עבדים. הכמות תלויה במספר הקווים הפנויים של המסטר שישמשו כבחירת העבד - SS	אין שימוש בכתובת כי החיבור הוא ישיר בין 2 רכיבים.
12	**אישורים**	המסטר מקבל אישור Acknowledge מהעבד שהנתון שהוא שלח התקבל , ולהפך המסטר שולח לעבד אישור שהנתון שהעבד שלח התקבל .	אין אישור קבלת נתון מצד לצד	אין אישור קבלת נתון מצד לצד
13	**כיוון נתונים**	התקשורת היא Half Duplex , כלומר לא ניתן לשדר ולקלוט בו זמני. קו השעון חד כיווני מהמסטר לעבד. קו הנתון דו כיווני.	התקשורת היא Full Duplex . כל קו הוא חד כיווני.	התקשורת היא לרוב Full Duplex (שידור וקליטה בו זמנית). כל קו הוא חד כיווני, כלומר מעביר נתון רק בכיוון אחד.
14	**יתרונות**		* פרוטוקול פשוט ולא דורש זמן עיבוד	התקשורת פשוטה והכי נפוצה

www.arikporat.com

		• ניתן להשתמש ביותר ממסטר אחד. • רק 2 קווי תקשורת. • מיעון הכתובות פשוט ולא דורש קווי SS (בחירת עבד) נפרדים כמו ב SPI וניתן להוסיף רכיבים בקלות. • בגלל השימוש ב open collector יש גמישות במתח היציאה (ניתן לחבר את נגדי המשיכה למעלה למתח ספק רצוי. • פחות רגיש לרעש מ SPI • אפשר להאט את קצב התקשורת עם רכיבים איטיים בעזרת Clock Stretch . המושג נקרא Flow Control שפירושה	רב או תוכנה מורכבת. *. קצב התקשורת גבוה והטווח גדול. * תצרוכת ההספק נמוכה מ I2C. * התקשורת היא Full Duplex כלומר ניתן לשדר ולקלוט בו זמנית. * תצרוכת הספק קטנה יחסית ל SPI .	ומשתמשת ב UART . כמעט לכל המיקרו בקרים יש כזו תקשורת. היא נקראת לפעמים RS232 אם כי ליתר דיוק תקן RS232 מגדיר רמות מתח אחרות מאלו שיש למיקרו בקר העובד עם 5 וולט.

www.arikporat.com

		התאמת מהירות בין רכיב מהיר ורכיב איטי. • זולה מ SPI בגלל כמות הקווים.		
15	**חסרונות**	• התקשורת היא Half Duplex , כלומר לא ניתן לשדר או לקלוט בו זמנית. • התוכנה מורכבת יותר מ SPI . • טווח תקשורת קטן מ SPI . • מהירות תקשורת נמוכה מ SPI . • יש אישור - ACKnowledge - מצד לצד שהנתון התקבל.	• לכל עבד יש לחבר קו SS משלו דבר המגדיל את כמות הפינים הנדרשת ויש לבצע התאמת תוכנה מתאימה. • רגיש יותר לרעש מ I2C . • אין Flow Control .	• תקשורת בין 2 רכיבים בלבד. • קצבי תקשורת קבועים (19200, 33600..). • יש לתאם קצב תקשורת בין 2 הצדדים מראש לפני הפעלת התקשורת ביניהם.

טבלה 2 : השוואה בין 3 סוגי התקשורת.

www.arikporat.com