

תוכן העניינים

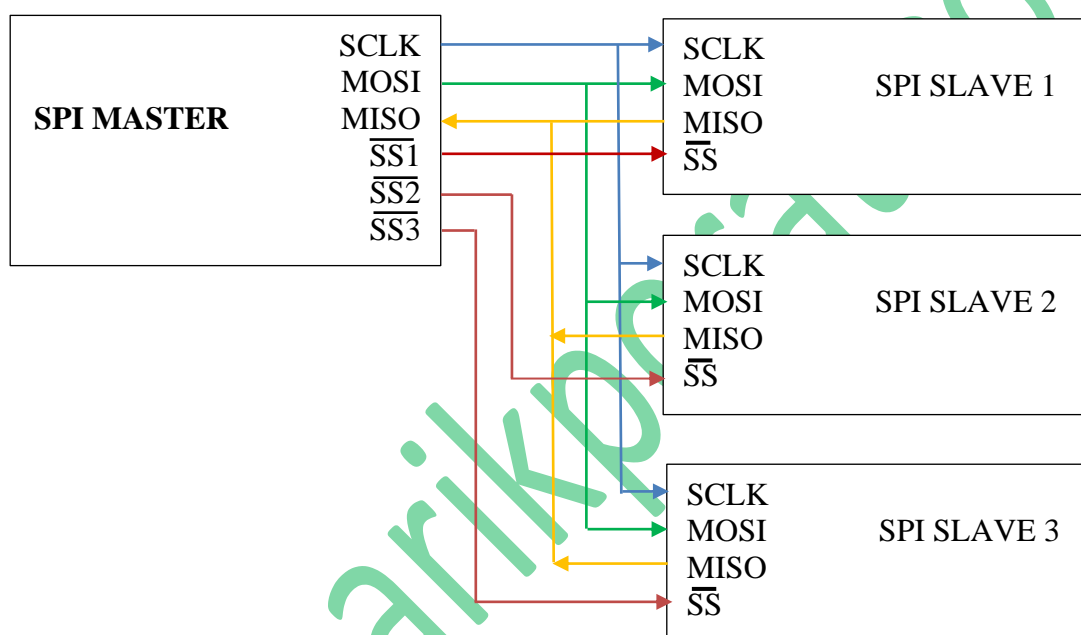
פרק 11 : תקשורת טורית SPI

3	11.1	מבוא לתקשורת SPI
4	11.2	תבנית (פורמט) העברה של נתון עבור CPHA=0
5	11.3	תבנית (פורמט) העברה של נתון עבור CPHA=0
7	11.4	4 אופני העבודה בתקשורת SPI
8	11.5	מערכת ה SPI ברכיב C8051F380
8	11.5.1	היחידה SPI0
8	11.5.2	אופני העבודה
8	11.5.2.1	אופן 3 חוטים
9	11.5.2.2	אופן עבד 4 חוטים או אופן מרובה מסטרים
9	11.5.2.3	אופן מסטר 4 חוטים
10	11.6	דיאגרמת מלבנים של מערכת ה SPI
11	11.7	רגיסטרים של ה SPI
11	11.7.1	רגיסטר SPI0CFG
13	11.7.2	רגיסטר SPI0CN
15	11.7.3	רגיסטר SPI0CKR
16	11.7.4	רגיסטר SPI0DAT
16	11.8	תוכנה

פרק 11 : תקשורת טורית SPI

11.1 מבוא לתקשורת SPI

התקשורת נוצרה על ידי חברת מוטורולה ב 1979 עם יציאת המיקרו פרוססור של חברת מוטורולה שנקרא 68000 .
השם SPI הוא – Serial Peripheral Interface - ממשיק טורי היקפי . זוהי תקשורת טורית סינכרונית כי יש בה שעון
שמסנכרן את הנתונים
באיור הבא מתואר חיבור טורי של תקשורת SPI .



איור 1 : תקשורת טורית SPI .

באיור רואים שבתקשורת 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 - Serial CLock** .

ד. קו נוסף נקרא **SS – Slave Select** - בחירת עבד. בעזרת קו זה המיקרו בקר מודיע לאיזה עבד הוא פונה. לכל עבד יש קו \overline{SS} משלו . במסטר יש מספר קווים לפי מספר העבדים. לכל רכיב עבד יתחבר קו מהמסטר. באיור הם נקראים $\overline{SS1}$ $\overline{SS2}$ $\overline{SS3}$. הם מסומנים עם גג מעליהם לציין שהקו פעיל בנמוך **Active Low**. לדוגמה המסטר מוריד את הקו $\overline{SS1}$ ל 0 ואז רק רכיב העבד הראשון (SPI SLAVE 1) יודע שהמסטר מתקשר איתו. שאר העבדים יודעים שהמיקרו בקר איננו מתקשר איתם.

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 – פאזה השעון) הקובעת באיזה מעבר (האם מ 0 ל 1 או מ 1 ל 0) 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.

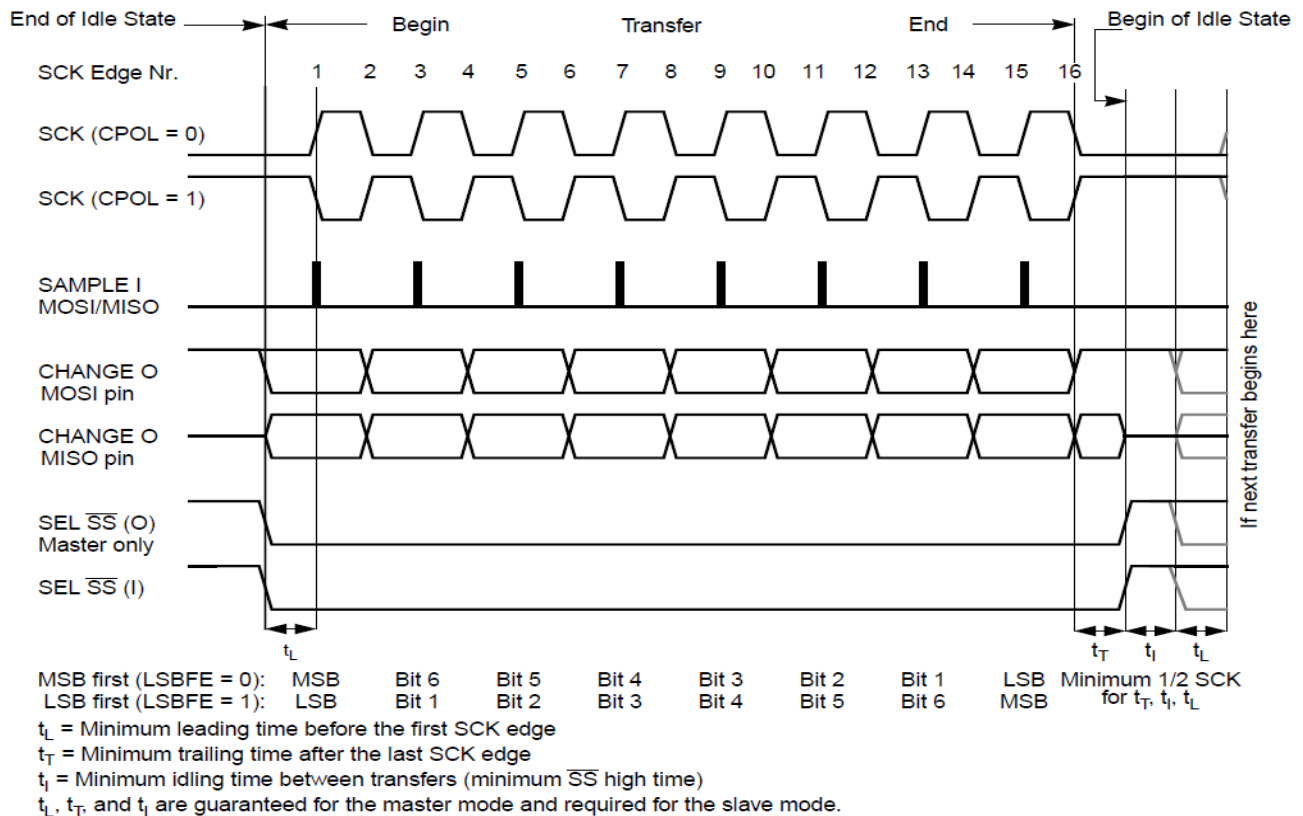
11.2 תבנית (פורמט) העברה של נתון עבור CPHA=0

האיור הבא מתאר את תבנית העברה עם צורות הגל של תקשורת SPI כאשר CPHA=0. בחלק התחתון רואים את התחלת התקשורת כאשר קו \overline{SS} יורד ל 0 ואת סיומה כאשר הקו \overline{SS} עולה ל 1. הקו פעיל בנמוך ובחלק מהשרטוטים הוא נקרא NSS כדי שלא נצטרך לשים קו מעל ה SS.

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

במקום הרשום באיור O הכוונה Output (יציאה) ובמקום הרשום 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 בירידת פולסי השעון ואת זה נקבע לפי נתוני הרכיב שאליו מתחברים.



איור 2 : 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 MOSI). באיור 3 במרכז רואים את הקו נקרא SAMPLE והוא כניסה (מסומן באיור ב I) גם למסטר בקו MISO וגם לעבד בקו 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 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 \overline{SS} ל 0 ועד להתחלת פולסי השעון.

t_T הוא הזמן המינימאלי בין פולס השעון האחרון והעלאת קו \overline{SS} ל 1.

t_I הוא זמן הסרק - IDLE - בין העלאת קו ה \overline{SS} ל 1 ועד שידור נתון חדש על ידי הורדת \overline{SS} ל 0.

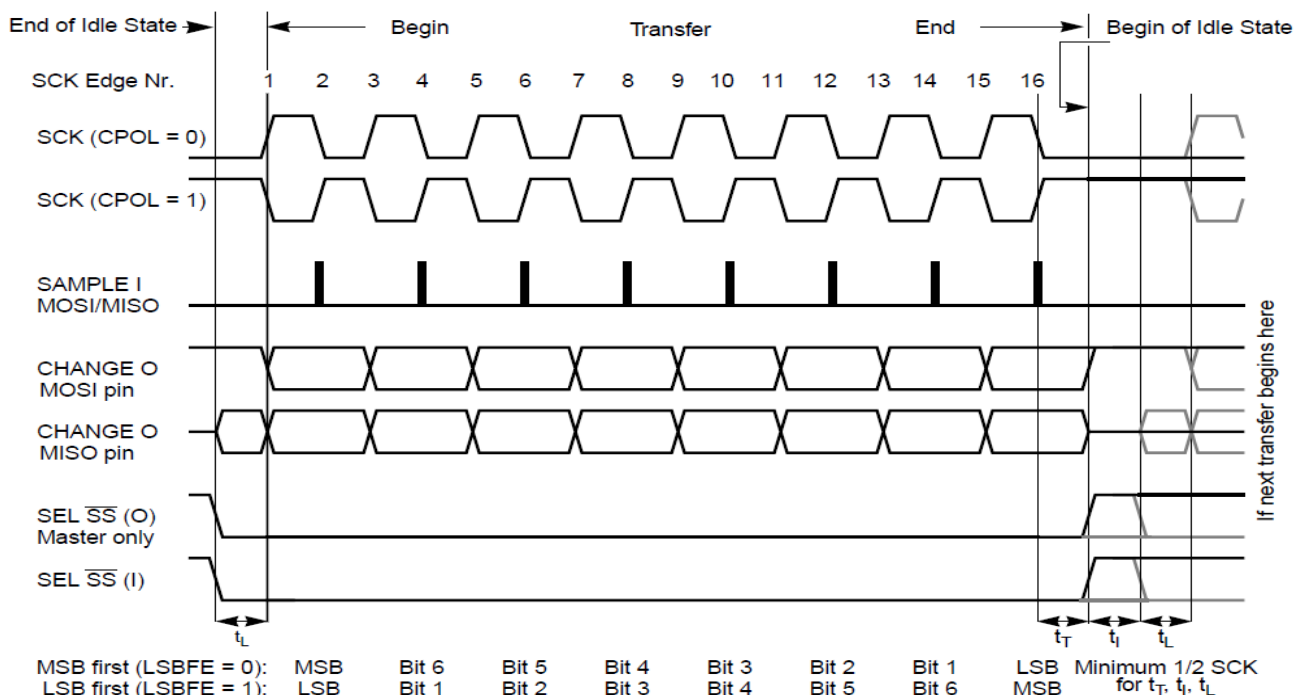
מתחת לצורות הגלים ניתן לראות שאפשר לשדר את הנתון מביט ה LSB אל ה MSB או להיפך. קיים ביט LSBFE (LSB First Enable – אפשר לראשון) שקובע מי נכנס ראשון. כאשר הוא 0 נכנס ביט ה MSB ראשון וכאשר הוא 1 נכנס ה LSB ראשון.

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.

11.3 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 4 מתאר את התקשורת עבור CPHA=1.



איור 3 :

צורות גל

בתקשורת

SPI

כאשר 1

CPHA

=

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 שננעל במעבר השני לתוך ה LSB או ה MSB של רגיסטר ההזה (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 וההזה קוראת במעברים האי זוגיים.

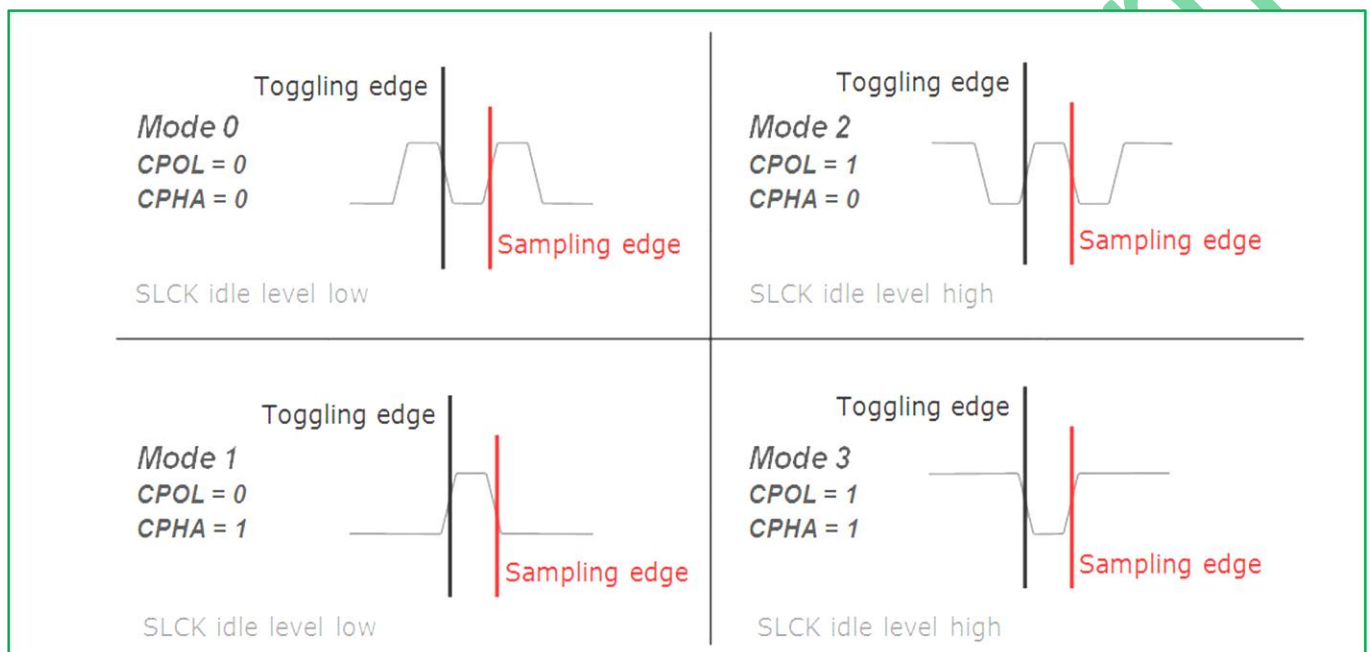
בדומה להעברה עבור CPHA=0, גם כאן קבלת הנתון היא בשני שלבים. היא מוזזת לרגיסטר ההזה של ה SPI בזמן ההעברה

ומועברת לרגיסטר הנתון של ה SPI כאשר הביט האחרון הוזז פנימה.

11.4 - 4 אופני העבודה בתקשורת SPI

באיור הבא רואים את 4 אופני העבודה של התקשורת SPI. האופנים מוגדרים בעזרת הפרמטרים Clock POLarity - CPOL וקוטביות השעון 1 - Clock PHase - CPHA – פאזה השעון. הם מגדירים 3 פרמטרים :

- את המעבר (שפה EDGE) – של ביט הנתון אל קו הנתון (גם במסטר וגם בעבד) מתואר על ידי הקו השחור בכל אחד מהאיורים.
- 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 4 : 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 וההזזה של הנתון ברגיסטר ההזזה - בירידת השעון. כשאין תקשורת הקו ב 0.

אופן 2 : CPOL=1 CPHA=0

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

אופן 3 : CPOL=1 CPHA=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 (CPOH CPOL 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 MISO).

11.5 מערכת ה SPI ברכיב C8051F380

11.5.1 היחידה SPI0 – Serial Peripheral Interface – מממשק טורי היקפי.

היחידה מספקת גישה לפס טורי סינכרוני, 16 ביווני - Full Duplex, גמיש. היא יכולה לפעול כרכיב מסטר או עבד באופני עבודה של 3 חוטים (3-wire) או 4 חוטים (4 wire) ותומכת במצב מרובה מסטרים ועבדים על פס spi יחיד. ניתן לקנפג (קונפיגורציה) את אות בחירת העבד - Slave Select (NSS) ככניסה ולעבוד במצב של עבד או לא לאפשר פעולת אופן עבודה כמסטר בסביבה מרובת מסטרים ולהימנע ממאבק כאשר יותר ממסטר אחד מנסה לבצע העברת נתון סימולטאנית. קו ה NSS יכול להיות יציאה באופן עבודה כמסטר או לא להיות מאופשר בפעולה של 3 חוטים (3 wire). בנוסף, ניתן להשתמש בהדק כללי של I/O ולבחור מספר רכיבי עבד במצב עבודה של מסטר. אנחנו נשתמש ביחידה SPI0 באופן עבודה של מסטר עם רכיב עבד אחד. כאשר נרצה לחבר עבד נוסף נשתמש באחת מרגלי המיקרו כקו SS נוסף.

11.5.2 – אופני העבודה

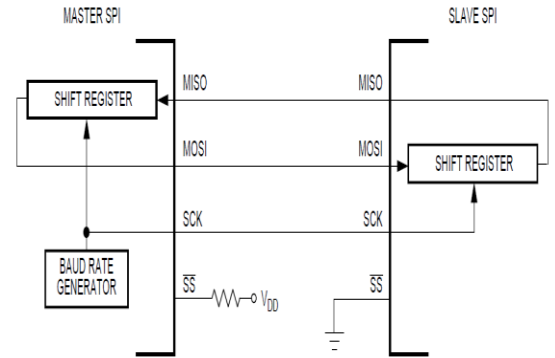
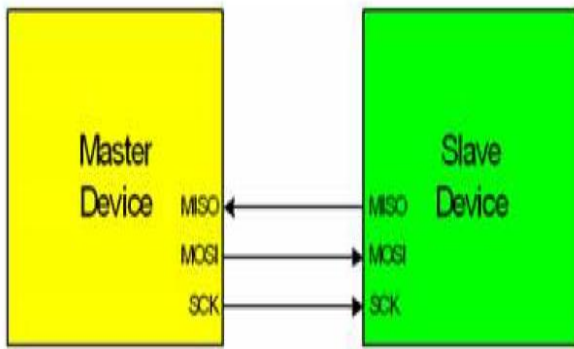
קיימים 3 אופני עבודה :

- * 3-wire Master or 3-Wire Single Slave Mode.
- * 4-Wire Slave or Multi-Master Mode.
- * 4-Wire Master Mode.

נסביר בקצרה את 3 האופנים.

11.5.2.1 3-wire Master or 3-Wire Single Slave Mode - אופן 3 חוטים עם מסטר או 3 חוטים עם עבד יחיד.

באופן עבודה זה היחידה יכולה לשמש כמסטר או עבד עם 3 חוטים. איור 6 השמאלי מתאר את החיבור של 3 החוטים והאיור הימני מראה את הדק ה SS במסטר ובעבד:

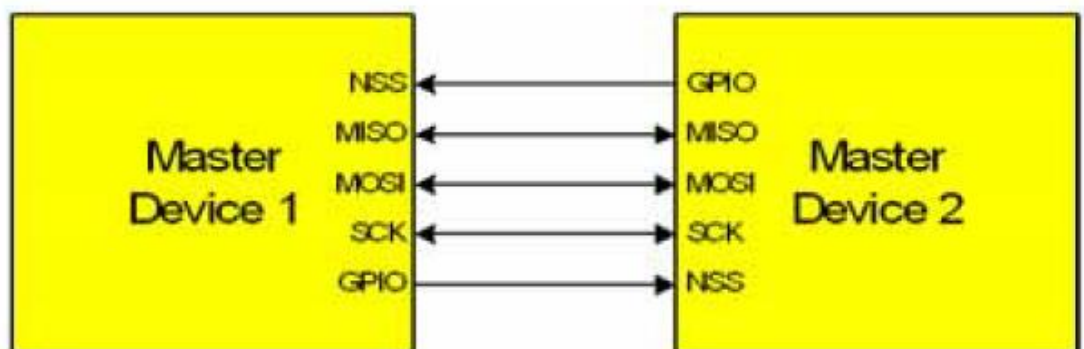


איור 5 : אופן עבודה מסטר או עבד עם 3 חוטים. משמאל שלושה החוטים ומימין מה עושים עם ה SS .

באיור רואים את רכיב המסטר בצד שמאל והעבד בצד ימין. קו ה SS איננו מחובר בין הרכיבים. במסטר הוא מחובר דרך נגד ל V_{DD} ובעבד הוא מחובר לאדמה. בצורה כזו העבד "יודע" ש"מדברים" אליו. גם כאן את פולסי השעון יוצר מחולל קצב באוד ברכיב המסטר ופולסי השעון גורמים להזזת הנתונים בתוך רגיסטרי ההזזה גם במסטר וגם בעבד (כפי שהסברנו בפרק הקודם). היחידה SPI0 יכולה להיות המסטר או העבד.

11.5.2.2 4-Wire Slave or Multi-Master Mode – אופן עבד 4 חוטים או אופן מרובה מסטרים.

אופן עבודה זה הוא עם 4 חוטים והדק NSS מאופשר ככניסה. בפעולה כעבד קו ה NSS בוחר את רכיב ה SPI0 . בפעולה כמסטר בסביבה מרובת מסטרים, המסטר הראשון שמוריד את קו ה NSS מ 1 ל 0 לא מאפשר את פעולת ה SPI0 של האחרים והוא זה המנהל את הפס, כך שמספר רכיבי מסטר יכולים להיות בשימוש בפס ה SPI . איור 7 מתאר דיאגרמה של אופן מרובה מסטרים. רואים שקווי הנתונים MOSI ו MISO הם דו כיווניים וכך גם קו השעון. לכל מסטר יש קו NSS משלו והרכיב הראשון שמוריד את קו ה NSS ל 0 הוא המסטר שיוצר את פולסי השעון ובהתאמה את כיוון הנתונים בקווי MOSI ו MISO .

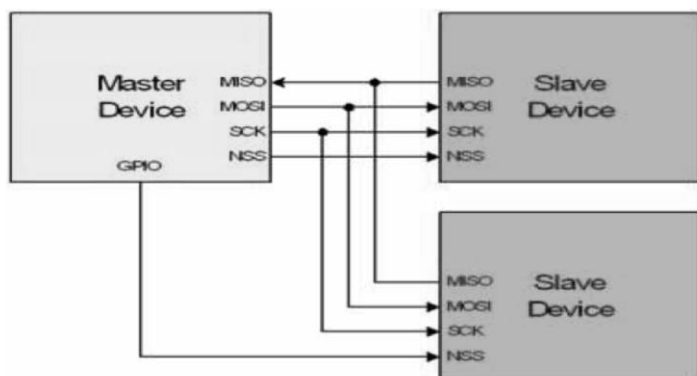


איור 6 דיאגרמה חיבור של אופן מרובה מסטרים.

11.5.2.3 4-Wire Master Mode – אופן מסטר 4 חוטים .

ה SPI0 פועל עם 4 חוטים וקו ה NSS הוא יציאה. בקונפיגורציה זו נשתמש רק כאשר SPI0 פועל כרכיב מסטר. האיור הבא

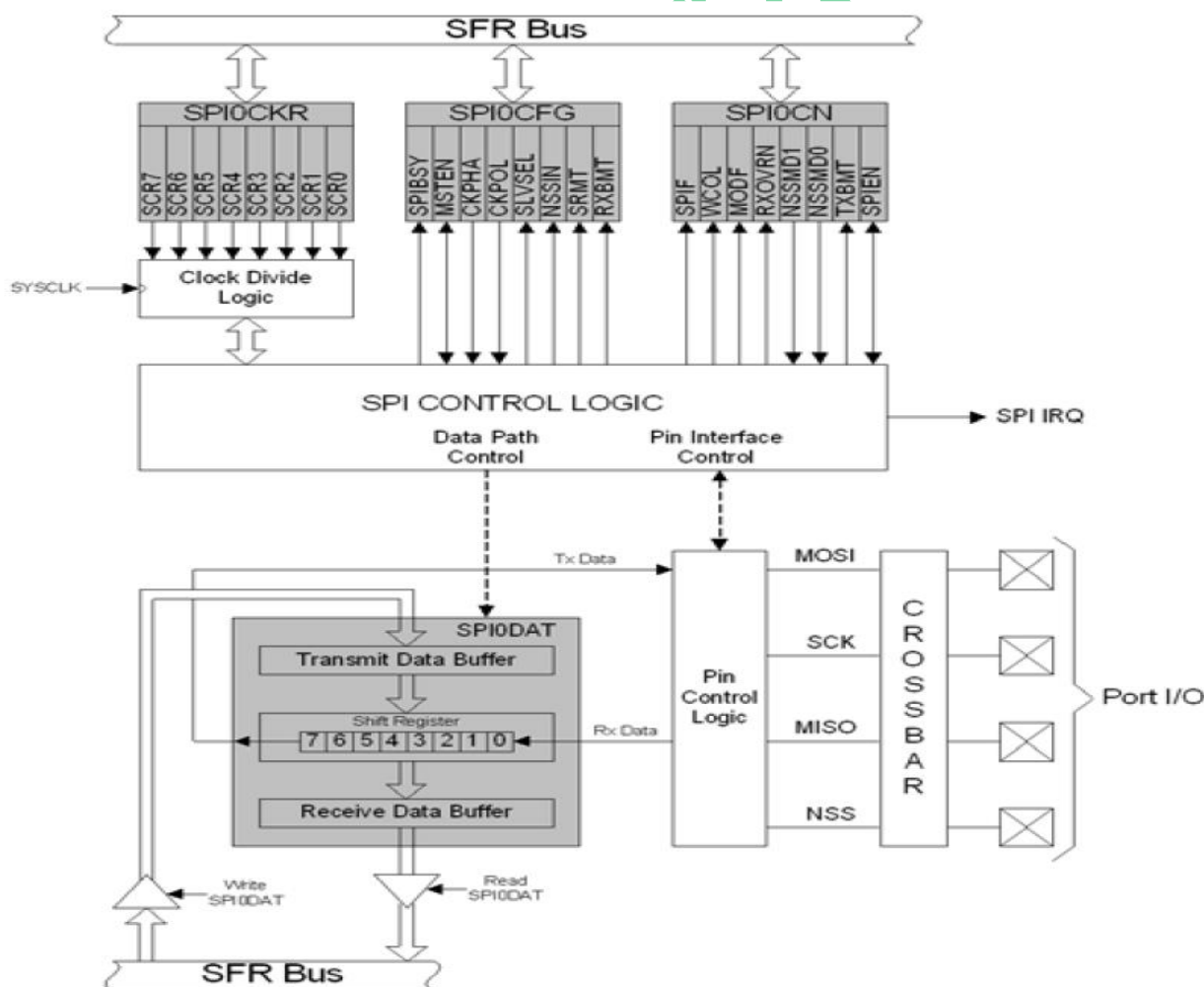
מתאר רכיב מסטר המתחבר אל כל אחד מ 2 העבדים בעזרת 4 חוטים. בעזרת קו נוסף GPIO - General Purpose Input Output - קלט פלט למטרה כללית -ניתן לפנות לרכיב עבד נוסף. המשתמש יצטרך להוריד את הקו הזה ל 0 בתוכנה.



איור 7 : חיבור באופן מסטר 4 חוטים.

11.6 דיאגרמת מלבנים של מערכת ה SPI

האיור הבא מתאר את המבנה המלבני של יחידת ה SPI .



איור 8 : סכמה מלבנית יחידת ה SPI .

בחלק העליון ובחלק התחתון של האיוור רואים את המלבן SFR Bus. הרגיסטרים השולטים על מעגלי האלקטרוניקה של ה SPI נמצאים באזור ה SFR (Special Functions Registers) – הרגיסטרים לתפקידים מיוחדים של המיקרו בקר.

בחלק העליון באיוור רואים 3 רגיסטרים (בצבע אפור) שאותם המשתמש מתכנת ועל ידי כך קובע את אופני העבודה , קצב התקשורת ושאר הפרמטרים של ה SPI . המשתמש שולח 3 מילות בקרה ל 3 הרגיסטרים ומבצע קונפיגורציה למערכת ה SPI . במרכז האיוור יש את מלבן **SPI CONTROL LOGIC** – לוגיקת הבקרה של ה SPI . כאן נמצאים כל מעגלי האלקטרוניקה השולטים על ה SPI על פי המילים ב 3 הרגיסטרים שהמשתמש שלח להם מילות בקרה.

בחלק התחתון בצד שמאל – בצבע אפור – רואים את המלבן **SPI0DAT** . במלבן זה 3 רגיסטרים. במרכז יש את רגיסטר ההזזה Shift Register - שממש גם לשידור הביטים וגם לקליטת הביטים. עבור כל ביט שמשודר גם נקלט ביט וכך זה מתבצע עבור 8 הביטים.

מעל רגיסטר ההזזה יש את רגיסטר השידור שנקרא Transmit Data Buffer - חוצץ נתון השידור. לכאן המשתמש שולח את הביט שרוצים לשדר בקו ה MOSI והמערכת משדרת ביט אחרי ביט בהתאמה לפולסי השעון . מתחתיו יש את רגיסטר Receive Data Buffer – חוצץ נתון הקליטה. לכאן ייכנס הביט הנקלט מקו ה MISO . אחרי 8 פולסים (16 מעברים) הביט שנשלח לשידור בחוצץ נתון השידור יצא לקו ה MOSI והביט שנקלט נטען אל רגיסטר חוצץ נתון הקליטה. הטעינה מחוצץ נתון השידור לרגיסטר ההזזה היא מקבילית וכמוה גם ההעברה מרגיסטר ההזזה לחוצץ נתון הקליטה בסיום קליטת הביט. מימין למלבן SPI0DAT יש שני מלבנים PIN CONTROL LOGIC - לוגיקת בקרת הדקים ומלבן ה CROSSBAR שבעזרתם קובעים את יציאות הדקים של ה SPI לפורט הרצוי.

11.7 רגיסטרים של ה SPI

ביחידת ה SPI0 4 רגיסטרים . 3 רגיסטרים השולטים על מערכת ה SPI0 ועוד רגיסטר נתון (למעשה יש 2 רגיסטרי נתון. באחד יש את הנתון לשידור ובשני יש את הנתון שנקלט). הם נמצאים ב SFR .

11.7.1 רגיסטר SPI0CFG (SPI0 ConFiGuration - תצורה ה SPI)

האיוור הבא מתאר את רגיסטר התצורה (קונפיגורציה).

בחלק העליון של האיוור רואים טבלה ובה 4 שורות. השורה העליונה מתארת את מספר הביט (מ 0 עד 7) . השורה השנייה מתארת את שם הביט. השורה השלישית האם קוראים מהביט Read או כותבים לביט Write או שניתן לקרוא מהביט וגם לכתוב לביט R/W (לא בו זמנית). בשורה הרביעית מראים מה מצב הביטים אחרי פעולת RESET .

כתובת הרגיסטר הוא A1H בכל הדפים של ה SFR .

הסבר הביטים:

SPIBSY – SPI BuSY - עסוקה SPI : מראה שהמערכת עסוקה בהעברת נתון. 1 בביט מראה שמתבצעת העברת נתון (גם באופן מסטר וגם באופן עבד). 0 מראה שהסתיים תהליך ההעברה. ניתן לעבוד איתו בשאילתה - POLLING . ביט קריאה בלבד.

MSTEN - MaSTer mode Enable - אפשרות אופן מסטר : אם נשים בביט 0 לא מאפשרים אופן מסטר ועובדים באופן עבד. אם שמים 1 עובדים כמסטר.

Bit	7	6	5	4	3	2	1	0
Name	SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT
Type	R	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	1	1	1

SFR Address = 0xA1; SFR Page = All Pages

Bit	Name	Function
7	SPIBSY	SPI Busy. This bit is set to logic 1 when a SPI transfer is in progress (master or slave mode).
6	MSTEN	Master Mode Enable. 0: Disable master mode. Operate in slave mode. 1: Enable master mode. Operate as a master.
5	CKPHA	SPI0 Clock Phase. 0: Data centered on first edge of SCK period.* 1: Data centered on second edge of SCK period.*
4	CKPOL	SPI0 Clock Polarity. 0: SCK line low in idle state. 1: SCK line high in idle state.
3	SLVSEL	Slave Selected Flag. This bit is set to logic 1 whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.
2	NSSIN	NSS Instantaneous Pin Input. This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.
1	SRMT	Shift Register Empty (valid in slave mode only). This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK. SRMT = 1 when in Master Mode.
0	RXBMT	Receive Buffer Empty (valid in slave mode only). This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0. RXBMT = 1 when in Master Mode.
Note: In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 25.1 for timing parameters.		

איור 9 : רגיסטר SPI0CFG.

CKPHA - ClocK PHase – פאזת השעון. אם שמים 0 הנתון נדגם ומוזז במעבר הראשון. אם שמים 1 הנתון נדגם ומועבר במעבר השני (הסבר מפורט בפרק ה').

CKPOL – ClocK POLarity – קוטביות השעון : מראה מה מצב הדק השעון במצב IDLE . אם נשים 0 הקו של פולסי השעון יהיה ב 0 במצב שאין תקשורת. אם נשים 1 הקו יהיה ב 1 .

SLVSEL - SLaVe SElect flag – דגל בחירת עבד : כאשר עובדים באופן עבד אז בביט יהיה 1 כאשר בהדק ה NSS יש 0 (מראה שהרכיב שלנו העובד באופן עבד נבחר ומסטר חיצוני פונה אל הרכיב שלנו) ובביט יהיה 0 כאשר בהדק ה NSS יש 1 (הרכיב שלנו לא נבחר).

NSSIN - NSS instantaneous pin INput – הדק כניסה מיידי : הביט מראה מה הערך שיש בהדק ה NSS בזמן שקראנו את הרגיסטר.

SRMT - Shift Register eMpTy – רגיסטר הזזה ריק : הביט תקף רק באופן עבד . הוא עולה ל 1 כאשר הנתון ברגיסטר ההזזה הועבר החוצה או פנימה ואין נתון חדש בחוצץ השידור או הקליטה. הביט חוזר ל 0 כאשר ביית מועבר אל רגיסטר ההזזה מחוצץ השידור או במעבר של SCK . במצב מסטר יש בביט 1 .

RXBMT – Receive Buffer eMpTy – חוצץ קליטה ריק : הביט תקף רק באופן עבד . הוא עולה ל 1 כאשר חוצץ הקליטה נקרא על ידי המיקרו ואין בחוצץ אינפורמציה חדשה. הביט ירד ל 0 כאשר יש נתון חדש בחוצץ הקליטה שעדיין לא נקרא על ידי המיקרו. במצב של מסטר יש בביט 1 .

הערה : באופן עבד , הנתון ב MOSI נדגם באמצע של כל ביט נתון. באופן מסטר הנתון ב MISO נדגם SYSCLK (פולס שעון מערכת) לפני סיום כל ביט נתון לאחר זמן settling מקסימאלי לרכיב העבד.

11.7.2 רגיסטר SPI0CN - SPI0 CoNtrol – בקרת SPI0 :

איור 11 מתאר את רגיסטר בקרת ה SPI0 . כמו ברגיסטר הקונפיגורציה, גם כאן השורות בטבלה העליונה מראה את שם הביט, האם הוא לקריאה או כתיבה או גם קריאה וגם כתיבה. השורה הרביעית בטבלה הזו מראה מה מצב הרגיסטר אחרי RESET . כתובת הרגיסטר F8H . בכל דפי ה SFR וניתן לגשת אליו במיעון סיבית (לכל ביט בנפרד).

הסבר הביטים:

SPIF - SPI0 Interrupt Flag – דגל פסיקה של SPI0 : הביט עובר ל 1 על ידי החומרה בסיום העברה של נתון. אם אפשרנו פסיקות SPI נקבל פסיקת SPI . הביט לא עובר אוטומטית ל 0 על ידי החומרה בסיום פסיקה ויש לאפס אותו בתוכנה.

WCOL - Write COLLision flag – דגל התנגשות כתיבה : הביט עולה ל 1 אם היה ניסיון לכתוב ל SPI0DAT כאשר הביט TXBMT=0 (מראה שחוצץ השידור עדיין לא ריק ומתבצע עדיין העברת נתונים), כלומר כותבים נתון חדש לשידור

לפני שהסתיים שידור הנתון הקודם. במצב כזה חוצץ השידור לא יקבל את הביית החדש. אם מאפשרים פסיקות SPI קבל פסיקה שמציינת שגיאת התנגשות כתיבה. הביט איננו מאופס אוטומטית על ידי החומרה ויש לאפס אותו בתוכנה.

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	MODF	RXOVRN	NSSMD[1:0]		TXBMT	SPIEN
Type	R/W	R/W	R/W	R/W	R/W		R	R/W
Reset	0	0	0	0	0	1	1	0

SFR Address = 0xF8; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7	SPIF	SPI0 Interrupt Flag. This bit is set to logic 1 by hardware at the end of a data transfer. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
6	WCOL	Write Collision Flag. This bit is set to logic 1 if a write to SPI0DAT is attempted when TXBMT is 0. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
5	MODF	Mode Fault Flag. This bit is set to logic 1 by hardware when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
4	RXOVRN	Receive Overrun Flag (valid in slave mode only). This bit is set to logic 1 by hardware when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
3:2	NSSMD[1:0]	Slave Select Mode. Selects between the following NSS operation modes: (See Section 25.2 and Section 25.3). 00: 3-Wire Slave or 3-Wire Master Mode. NSS signal is not routed to a port pin. 01: 4-Wire Slave or Multi-Master Mode (Default). NSS is an input to the device. 1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.
1	TXBMT	Transmit Buffer Empty. This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.
0	SPIEN	SPI0 Enable. 0: SPI disabled. 1: SPI enabled.

איור 10 : רגיסטר בקרת SPI0.

MODE - Mode Fault flag – דגל שגיאת אופן : הביט עובר ל 1 על ידי החומרה כאשר במצב מסטר התגלתה התנגשות בין 2 אופני עבודה.

כאשר עובדים במצב של NSS=0, MSTEN=1 ו NSSMD[1:0]=01 (ביטים 2 ו 3 ברגיסטר שיוסברו בהמשך). זהו אופן מרובה מעבדים וה SPI0 פועל **כעבד** עם 4 חוטים (כאשר NSS הוא כניסה !) ושמנו ברגיסטר הקונפיגורציה בביט MSTEN=1, כלומר נותנים אפשרות להפעיל **כמסטר**. נקבל שגיאה כי לא ניתן להפעיל גם כמסטר וגם כעבד. אם מאפשרים פסיקות SPI נקבל פסיקה. הביט איננו מתאפס אוטומטית על ידי החומרה ויש לאפס אותו בתוכנה.

RXOVRN - Receive Overrun Flag - דגל דריסת קליטה (תקף במצב עבד בלבד) : ביט זה עולה ל 1 על ידי החומרה כאשר בחוצץ הקליטה יש נתון מהעברה קודמת, שעדיין לא נקרא על ידי המיקרו והביט האחרון של ההעברה הנוכחית הוזז לתוך רגיסטר ההזזה של SPI0. אם אפשרנו פסיקות SPI נקבל פסיקה. הביט איננו מתאפס אוטומטית על ידי החומרה ויש לאפס אותו בתוכנה.

NSSMD[1:0] - Slave Select MoDe – אופן פעולה כעבד : 2 ביטים הבוחרים בין אופני הפעולה של קו ה NSS :

00 : אופן עם 3 חוטים כעבד או 3 חוטים כמסטר . הקו NSS לא מתחבר לאף הדק של פורט .

01 : אופן עם 4 חוטים כעבד או אופן מרובה מסטרים (זוהי ברירת המחדל). קו NSS הוא כניסה.

1x : אופן 4 חוטים כמסטר יחיד. הקו NSS הוא יציאה ויהיה בו את הערך של הביט NSSMD0 (הביט המסומן ב x ליד ה 1).

TXBMT - Transmit Buffer Empty – חוצץ שידור ריק : הביט עובר ל 0 כאשר המיקרו מעביר נתון חדש לחוצץ השידור. הביט עולה ל 1 כאשר הנתון יועבר מחוצץ השידור לרגיסטר ההזזה והוא מציין שניתן לשלוח נתון חדש לחוצץ השידור.

SPIEN - SPI0 Enable – אפשרור SPI0 : אם שמים בביט 0 לא מאפשרים עבודה עם יחידת ה SPI0 . אם נשים 1 – מאפשרים עבודה עם SPI0 .

11.7.3 רגיסטר SPI0CKR - SPI0 Clock Rate - קצב השעון של SPI0 :

בעזרת רגיסטר זה קובעים את קצב השעון בהדק ה SCK באופן עבודה כמסטר . יש לזכור שרק בעבודה כמסטר ה SPI0 הוא זה הנותן את פולסי השעון לרכיבי העבד.

גם כאן בטבלה העליונה רואים את 8 הביטים של הרגיסטר . הביטים נקראים SCR[7:0] . כתובת הרגיסטר A2H בכל הדפים של ה SFR . אחרי RESET כל הביטים שלו ב 0 .

האיור הבא מתאר את הרגיסטר :

Bit	7	6	5	4	3	2	1	0
Name	SCR[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA2; SFR Page = All Pages

Bit	Name	Function
7:0	SCR[7:0]	<p>SPI0 Clock Rate.</p> <p>These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where <i>SYSClk</i> is the system clock frequency and <i>SPI0CKR</i> is the 8-bit value held in the SPI0CKR register.</p> $f_{SCK} = \frac{SYSClk}{2 \times (SPI0CKR[7:0] + 1)}$ <p>for $0 \leq SPI0CKR \leq 255$</p> <p>Example: If <i>SYSClk</i> = 2 MHz and <i>SPI0CKR</i> = 0x04,</p> $f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$ $f_{SCK} = 200kHz$

איור 11 : רגיסטר SPI0CKR .

קצב פולסי השעון בהדק SCK נגזר על ידי חלוקה של תדר השעון של המיקרו. הוא נתון על ידי הנוסחה שבטבלה כאשר
 SYSCLK הוא תדר השעון של המיקרו. בדוגמה שבטבלה תדר השעון של המיקרו הוא 2MHz והמספר ששמנו ב SCR[7:0]
 הוא 4: B00000100.

דוגמה נוספת: עבור תדר שעון של 48MHz וחלוקה של 01000000B (חלוקה ב 64) נקבל:

$$f_{SCK} = 48000000 / 64 = 750\text{KHz}$$

הערה: יש לבדוק שרכיב העבד מסוגל לעבוד במהירות כזו.

11.7.4 רגיסטר SPI0DAT - SPI DATa – רגיסטר הנתון של ה SPI0 :

האיור הבא מתאר את הרגיסטר.

Bit	7	6	5	4	3	2	1	0
Name	SPI0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA3; SFR Page = All Pages

Bit	Name	Function
7:0	SPI0DAT[7:0]	SPI0 Transmit and Receive Data. The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.

איור 12: רגיסטר SPI0DAT

למעשה קיימים 2 רגיסטרים של נתון. האחד מכיל את הבייט לשידור והשני מכיל את הבייט הנקלט. שניהם נמצאים באותה כתובת ב SFR והגישה אליהם נבדלת בהדק כתיבה וקריאה פנימיים בתוך הרכיב. כתיבה לרגיסטר SPI0DAT שמה את הנתון בחוצץ השידור ומתחילה באופן מסטר העברה של נתונים. קריאה מ SPI0DAT מעבירה את הנתון שנקלט למיקרו.

11.8 תוכנה

נרשום פונקציה המקבלת נתון בן בגודל בייט ומשדרת אותו בקו MOSI אל העבד המתחבר לקו SS. לא נשתמש כאן בהדקי ה SPI של המיקרו בקר אלא נכתוב פונקציה עצמאית עם הדקים אקראיים שבחרנו. במקרה כזה לביטים CPOL ו COHA אין שימוש וקצב התקשורת תלוי בתוכנית שרשמנו.

נתאר את חיבורי ההדקים של המיקרו אל רכיב העבד. נצא מההנחה שהוגדרו ההדקים בקרוסבר והוצהרו הספריות המתאימות.

***** קבועים בגודל ביט *****

```
#define SS P1^5 // רגל הבחירה לרכיב הקול. פעילה בנמוך
#define MOSI P1^6 // רגל כניסת הנתונים הטורית לרכיב
#define SCLK P1^7 // רגל השעון של הרכיב.
```


נרשום את הפונקציה של שליחת נתון של 8 ביטים . הפונקציה מקבלת פקודה או נתון למשתנה בשם data8 מטיפוס תווי לא מסומן. הפונקציה איננה מחזירה ערך . בלולאת for אנחנו משדרים את הנתון מביט ה LSB אל ביט ה MSB . כל ביט מועבר בעליית השעון. לא ביצענו קליטה של הביט מהעבד אל המסטר בקו ה MOSI .

```
void send8bits(unsigned char data8)
```

```
{  
    unsigned char i;  
    SCLK=0;  
    SS=0; // הודעה לרכיב העבד "שמדברים אליו"  
    for(i=0;i<8;i++) // לולאה של 8 פעמים  
    {  
        MOSI=data8%2; // לשים את ביט ה LSB בקו ה MOSI  
        SCLK=1; // העלאת הדק השעון ל 1  
        data8=data8/2; // ( D1 עובר ל D0 )  
        SCLK=0; // הורדת הדק השעון ל 0 . הביט מוזז ברגיסטר ההזזה של העבד.  
    }  
}
```

הסברים נוספים על התקשורת ולהשוואה בין סוגי התקשורת UART , SPI , I2C ניתן למצא בקישור :

[serial_communication_comparison.pdf \(arikporat.com\)](http://www.arikporat.com/serial_communication_comparison.pdf)