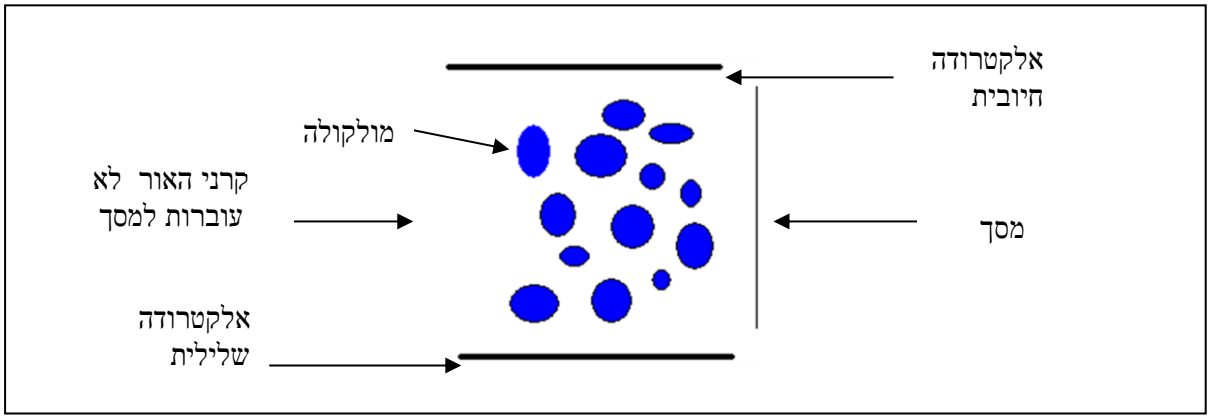
תצוגת TFT עם ארדואינו אונו עם הסבר על בקר HX8357-D00/D01

1. מהי תצוגת Liquid Crystal Display – LCD – תצוגת גביש נוזלי- עקרונות

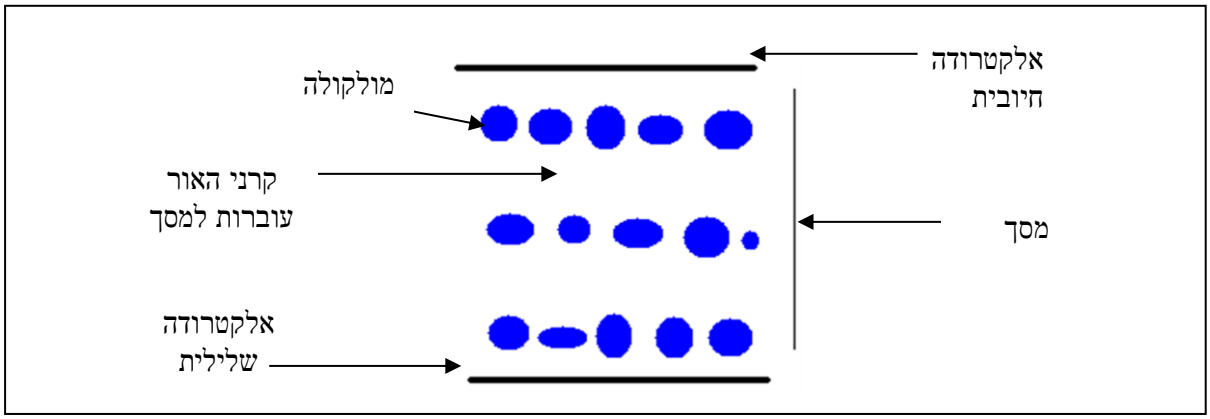
1.א. עקרון של תצוגת גביש נוזלי

באיור 1 מתוארים המולקולות של הגביש הנוזלי ללא שדה חשמלי. הקרנה של אור בצד אחד של התמיסה לכיוון הצד השני יגרום לקרני האור לפגוע במולקולות של הגביש ולחזור בחזרה ללא מעבר אל הצד השני של הגביש הנוזלי. לוח מסך שנמצא בצד השני יישאר חשוך כי לא מגיעות אליו קרני אור.



איור 1 – גביש נוזלי ללא שדה חשמלי

כאשר ניתן שדה חשמלי בין שני הלוחות שביניהן נמצא הגביש הנוזלי, יסתדרו המולקולות של הגביש בצורה מאד מסודרת וחלק מקרני האור שמוקרנים בצד אחד – יעבור אל הצד השני של הגביש הנוזלי ויאיר את המסך שנמצא בצד זה כפי שנראה באיור 2 .



איור 2 : גביש נוזלי בהשפעת שדה חשמלי.

באופן מעשי שמים את התמיסה של הגביש הנוזלי בין שני לוחות זכוכית. כל 2 לוחות כאלה נקראים סגמנט Segment. לוח זכוכית אחד מצופה בשכבת זהב דקיקה, והלוח השני בנוי מלוחות זהב דקיקים עד כדי שקיפות. התכונה המיוחדת של התמיסה היא שכאשר מופיע שדה חשמלי, המולקולות מסתדרות

בצורה גבישית, דבר הגורם לתמיסה שבתוך הסגמנט להפוך לשקופה. כלומר אם נספק אל אחד הלוחות העליונים שדה חשמלי ביחס ללוח הזהב התחתון יהפוך הסגמנט שמתחתיו לשקוף. אם יונח רקע צבעוני מתחתיו הוא יראה. ברגע שיופסק השדה החשמלי, יחזור הסגמנט להיות אטום והרקע יעלם.

1.א.1 מבנה של תצוגת גביש נוזלי

כל תצוגת גביש נוזלי מורכבת מ 2 חלקים : **א.** בקר תצוגה **ב.** תצוגת הגביש הנוזלי עצמו. הבקר הוא מעגל משולב (ג'וק) אשר מתחבר מצד אחד אל מיקרו בקר ומהצד השני אל התצוגה עצמה העשויה מלוח זכוכית שמתחתיה נמצאים הסגמנטים של הכביש הנוזלי . לכל הסגמנטים בלוח הזכוכית יש מוליכים המתחברים אל בקר התצוגה ומופעלים על ידו.

1.ב. תצוגת LCD פסיבית

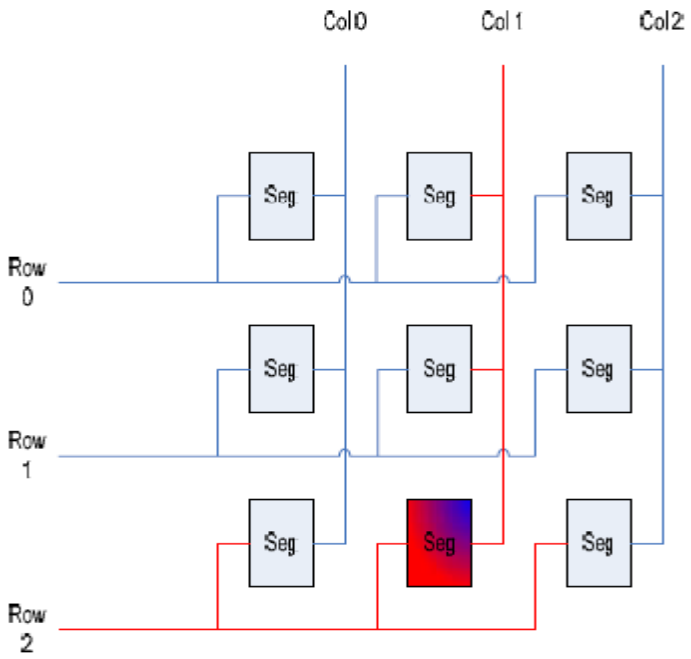
ישנם 2 סוגי תצוגות LCD

א. פסיבית – אין לה מקור אור משלה ויש לספק לה מקור אור חיצוני.

ב. אקטיבית – יש לה טרנזיסטורים הממתגים מתח לכל אחד מהפיקסלים. רוב מסכי המחשב הם מסוג זה.

1.ב. 1 תצוגת LCD פסיבית

תצוגת LCD פסיבית מורכבת מרשת של שורות ועמודות כמתואר באיור 3 .



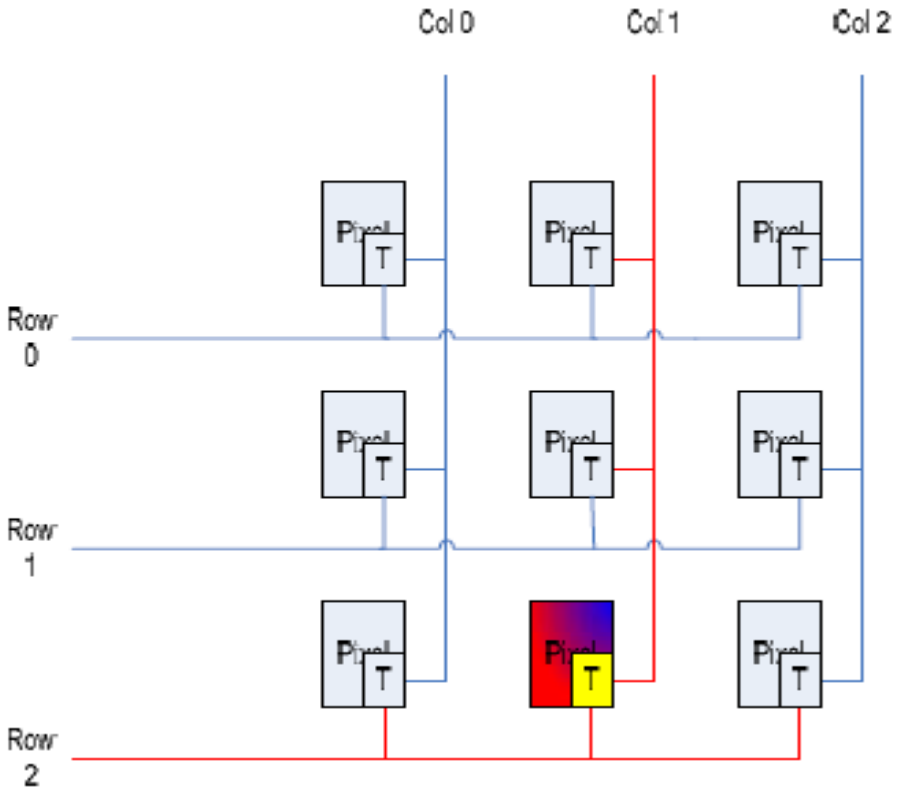
איור 3 - תצורת הסריג בתצוגת LCD פסיבית

השורות והעמודות מחוברות במאונך אחת כלפי השנייה ומחוברות לכל סגמנט (Seg). השורות והעמודות מחוברות לג'וק (מעגל אלקטרוני) שקובע איזו שורה ואיזו עמודה תיבחר. ממתח קטן מתחבר אל השורה והעמודה שנבחרו כדי ליצור שדה חשמלי בנקודת המפגש שלהם. כאמור השדה קובע אם המולקולות יהיו מסודרות ויהיה מעבר של אור.

2.ב.1 - תצוגת LCD אקטיבית

בנויה גם היא מרשת אותות חשמליים של שורות ועמודות . אלא שכאן השורות והעמודות מתחברות לטרנזיסטור בכל סגמנט של ה LCD . גם כאן , מעגל משולב (ג'וק) קובע מי מהשורות והעמודות ייבחרו. השורה והעמודה שנבחרו מאפשרים את הטרנזיסטור. במצב כזה מטען מאוחסן בטרנזיסטור וזה גורם לסדר את כל המולקולות בצורה אחידה ולמעבר של אור.

בתצוגה פסיבית נדרשת תאורה איורית חזקה יותר מזו שבתצוגה פסיבית.



איור 4 - תצורת הסריג בתצוגת LCD אקטיבית

תצוגות TFT - Thin Film Transistor - טרנזיסטור סרט דק – הן תצוגות אקטיביות.

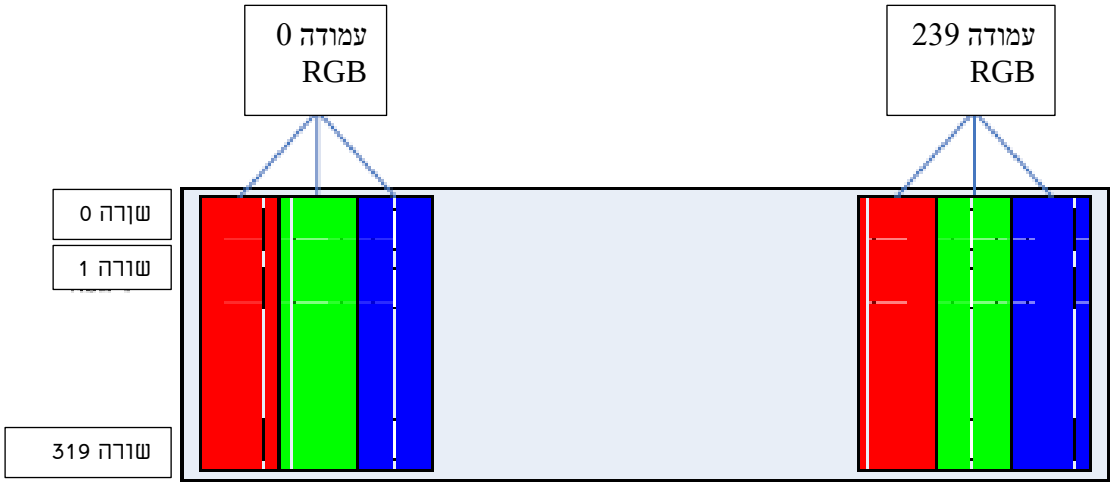
כל סגמנט בתצוגה מאפשר החזרה או יצירה של אור.

כדי ליצור צבעים צריך 3 סגמנטים. 3 סגמנטים אלו מעבירים אור דרך מסננת אדומה , ירוקה או כחולה.

כך נוצר פיקסל RGB . כלומר פיקסל מורכב מ 3 סגמנטים.

עבור תצוגה של 240*320 יש 3*320 סגמנטים (עמודות) ו 240 שורות.

באיור 5 רואים את סידור העמודות והשורות של תצוגת RGB של 320 שורות ו 240 עמודות.



איור 5 - תצוגת RGBשל 320 שורות ו 240 עמודות

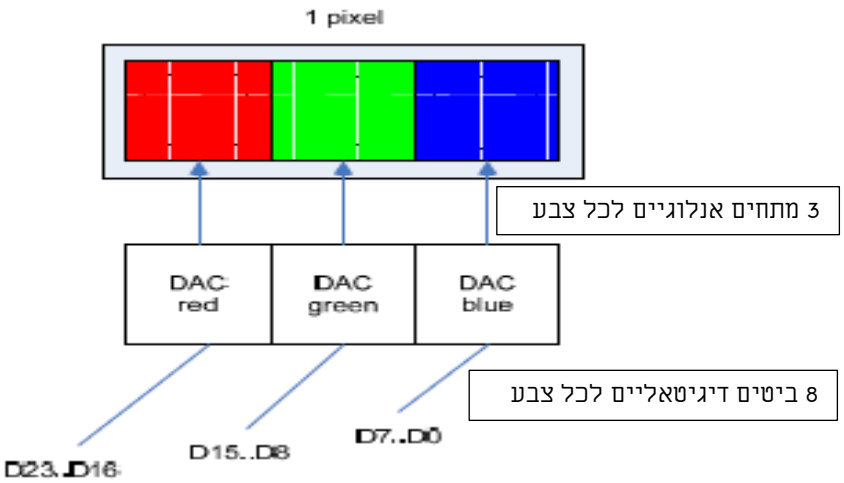
3.ב.1 יצירת צבע בתצוגת TFT (Thin Film Transistor)

תצוגות TFT יכולות לדחוף 3 סגמנטים (פיקסל אחד) בפולס שעון אחד עם שדה חשמלי משתנה חזק. גודל המתח האנלוגי בכל צבע קובע את עצמת הצבע המסוים. תצוגה כזו תומכת בצבעים רבים. רמות הצבע נקבעות על ידי מספר קווי הנתון בפנל התצוגה ובמספר קווי יציאות הנתון של הבקר .

האפשרויות הן : 24 ביט לפיקסל (bits per pixel – bpp) , 18bpp , 16bpp , 15bpp , 8bpp .

בממשק מקבילי נדרשים 320 פולסי שעון ל 320 פיקסלים.

באיור 6 רואים ממשק של 24 ביטים לפיקסל בשעון אחד.

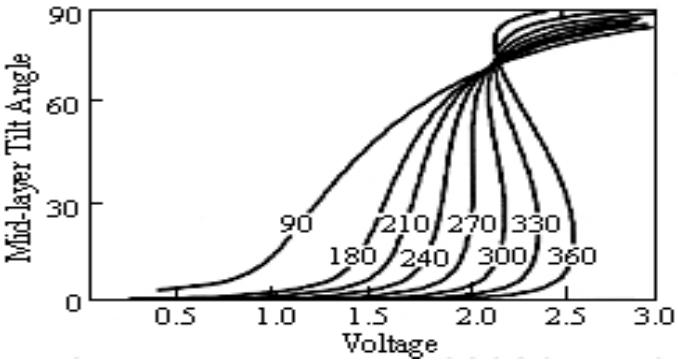


איור 6 – ממשק של 24 ביט לפיקסל בפולס שעון אחד.

איור 6 מתאר תצוגה המתחברת בממשק של 24 ביט. כל 8 ביטים מתחברים ל DAC הקובע את גודל המתח שיגיע לכל צבע בפיקסל מסוים. הצבע הכחול נשלט על ידי 8 הביטים , D0~D7 , הצבע הירוק על ידי הביטים D8~D15 והצבע האדום על ידי D16~D23 . 8 הביטים של המספר הדיגיטאלי קובעים את גודל המתח האנלוגי שיוצא מה DAC ואיתו את עצמת הצבע בכל אחד מ 3 צבעי ה RGB.

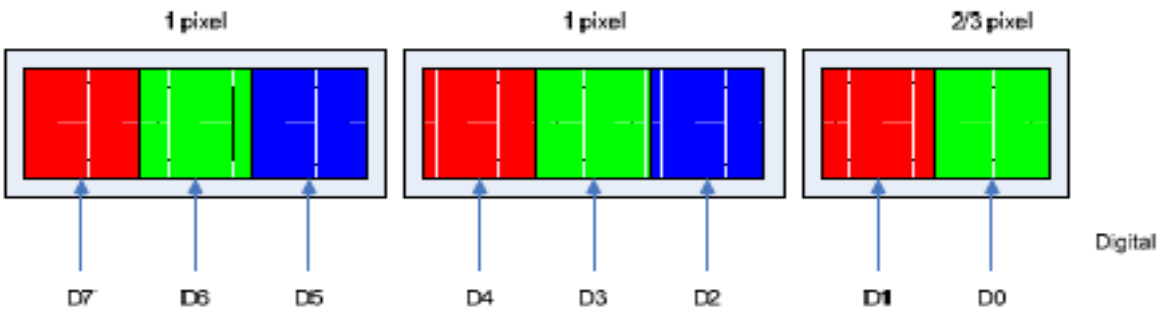
4.ב.1 - יצירת צבע בתצוגת STN (Super Twisted Nematic)

תצוגת STN שונה מתצוגה רגילה בזווית שבה המולקולות שלה בגביש הנוזלי יכולות להשתנות. בתצוגות הנקראות TN (Twisted Nematic) המולקולות יכולות לשנות את מצבן ב 90 מעלות. ב STN השינוי בזווית של המולקולות יכול להיות יותר מ 90 מעלות. בדרך כלל בין 180 ל 240 מעלות. תצוגות TN הן תצוגות פסיביות . תצוגות STN דורשות פחות הספק וקלות יותר לייצור מתצוגות TFT. באיור 7 רואים את שינוי זווית המולקולות בתלות במתח השדה החשמלי.



איור 7 - שינוי זווית המולקולות בתלות במתח השדה החשמלי בתצוגת STN.

התצוגה מתבססת על שבירת קרניים כפולה ועל ידי תאורה אחורית לבנה מגיעים לצבעי שחור לבן חדים יותר מתצוגה פסיבית. כאשר מוסיפים פסי RGB בתוך התצוגה מקבלים תצוגת RGB . תצוגות STN יכולות להפעיל סגמנט אחד או יותר בפולס שעון אחד (הפעלת עצמת שדה מקסימלית והפסקה של השדה). ניתן להפעיל מספר זעום של פיקסלים בפולס שעון. נדרשים 120 פולסי שעון להפעלת 320 פיקסלים כאשר בכל פולס שעון מופעלים 2-2/3 פיקסלים (עבור פס נתונים של 8 ביט) ו 240 פולסי שעון עבור 320 פיקסלים , כלומר 1-1/2 פיקסל לכל פולס שעון (עבור פס נתון של 4 ביט).
הסגמנטים מועברים בין מצבי ON ו OFF ליצירת עומק הצבע. זה יכול לקחת כמה מחזורי ריענון עד שמקבלים את הצבע הרצוי. לדוגמה : מנת דופק (duty cycle) של 50% בסגמנט ייתן 50% של עצמה. באיור 8 רואים הפעלה של 2 ו 2/3 פיקסלים המופעלים בפולס שעון אחד עבור פס נתונים של 8 ביט.



איור 8 - הפעלה של 2 ו 2/3 פיקסלים המופעלים בפולס שעון אחד עבור פס נתונים של 8 ביט

1.ג יצירת תמונה בתצוגה של 320*480

יצירת התמונה מתבצעת בדומה ליצירת תמונה במסך של טלוויזיה . מערך השורות והעמודות מחולק לפיקסלים. בתצוגה של 320*480 יש 153600 = 320*480 פיקסלים.

הבקר עובר על הפיקסלים בשורה הראשונה אחד אחרי השני ומפעיל כל פיקסל בצבע שנקבע לו.

בסיום השורה הראשונה הוא יורד ומפעיל בצורה דומה את הפיקסלים בשורה השנייה.

כך הוא עובר שורה אחר שורה עד השורה האחרונה. בסיום השורה האחרונה הוא חוזר אל השורה הראשונה.

היות והפעולה חוזרת על עצמה פעמים רבות בשנייה – **ריענון** - **Refreshing** - נדמה למסתכל שיש מסגרת שלמה של תמונה. מכאן ברור שיש ליצור אותות סנכרון אנכיים ואופקיים שייבחרו את השורה והעמודה הנכונים. בכל תצוגה יש בקר שתפקידו לרענן את הפיקסל המתאים. בבקר יש מצביע שורות ומצביע עמודות.

1 . ד אותות בקרה ותזמון

התצוגה צריכה את האותות הבאים :

VSYNC - Vertical SYNC – סנכרון אנכי לתצוגות TFT או FP - (Frame Pulse - פולס מסגרת) לתצוגות STN . משמש לאתחול מצביע השורות לחלק העליון של התצוגה.

HSYNC - Horizontal SYNC – סנכרון אופקי בתצוגות TFT או LP - Line Pulse (פולס שורה) בתצוגת STN המשמש לאתחול מצביע העמודות לקצה של התצוגה.

D0….Dxx - קווי הנתונים.

LCDCLK - **LCD CLOCK** - פולסי השעון של ה LCD להפעלת התצוגה.

אותות נוספים הם המתחים להפעלת התצוגה, תאורה אחורית , touchscreen (מסך מגע).

1.ה - תאורה אחורית - Backlight

התאורה האחורית איננה מתחברת דרך בקר התצוגה. ישנן מספר טכנולוגיות של תאורה אחורית בתלות בגודל פאנל התצוגה. פנלים גדולים משתמשים באור פלואורסצנט. פנלים קטנים משתמשים בלדים. ישנם כאלה המתחברים למקור זרם קבוע ואינם משנים את עצמת התאורה. יש כאלו שניתן להעביר אותם מ ON ל OFF ולהפך ויש כאלה שניתן לשנות את העצמה של התאורה בעזרת PWM .

רוב השימוש של תצרוכת ההספק היא על ידי התאורה האחורית. תצוגת QVGA קטנה צורכת 300mW ואפילו יותר. המתח של התאורה האחורית - Vbl- הוא כ 29 וולט (מעלים את מתח ה 3.3 וולט) והזרם כ 18 מילי אמפר.

1.ו - חוצצי המסגרת - Frame Buffers

חוצץ המסגרת הוא הזיכרון שבו יש את הנתונים המשמשים לריענון של התצוגה. הזיכרון המוקצה לחוצצי המסגרת , משותף , בדרך כלל , לרכיבי מערכת נוספים כמו ה CPU , DMA , רשת וכו'.

הזיכרון מאורגן כמערך של ביטים, בייטים , חצאי מילים או מילים. הדבר תלוי בעומק הצבע הנבחר ובארגון הביטים של הצבע.

נתוני פיקסל יכולים להיות ביטים מצופפים (8 פיקסלים ל בייט) או בייטים (עד 256 צבעים), או חצאי מילים של 16 ביט (עד 64K צבעים) או מילים של 24 ביטים (עד 16M צבעים).

גודל חוצצי המסגרת מחושב כך :

COLOMNS * ROWS * SIZEOF (PIXEL DATA)

לדוגמה :

בתצוגה של 800*600 העובדת עם 16bpp – 16 ביטים לפיקסל - (חצי מילה לפי 5 ביטים לאדום , 6 ביטים לירוק ועוד 5 ביטים לכחול) צריך : 600 עמודות 800 שורות ו 2 בתים לפיקסל :

600 * 800 * 2 = 960000 BYTES

בתצוגה של 24 ביטים (8 ביטים לכל צבע) מאחסנים את הנתונים במבנה של 32 ביט (הבייט הגבוה , הרביעי , לא נלקח בחשבון).

דוגמה נוספת :

בתצוגה של 320*480 העובדת עם 16bpp – 16 ביטים לפיקסל - (חצי מילה לפי 5 ביטים לאדום , 6 ביטים לירוק ועוד 5 ביטים לכחול) צריך : 320 עמודות 480 שורות ו 2 בתים לפיקסל :

320 * 480 * 2 = 307200 BYTES

1. ז . תבנית הביטים במרחב צבע RGB

ערכי הצבע של פיקסל מסודרים בשדות של ביטים לפי האפשרויות הבאות : הסימון U לא בשימוש.

RGB332 - 8bpp- המסודרים כך : 3 ביטים לאדום, 3 לירוק ו 2 לכחול. RRRGGGBB.

RGB555 - 16bpp - 5 ביטים לכל צבע. מסודרים - URRRRRGGGGGBBBBB.

RGB565 – 16bpp - 5 ביטים לאדום, 6 לירוק 5 לכחול. RRRRRGGGGGGBBBBB .

RGB888 – 24bpp – 8 ביטים לכל צבע. UUUUUUUURRRRRRRRGGGGGGGGBBBBBBBB

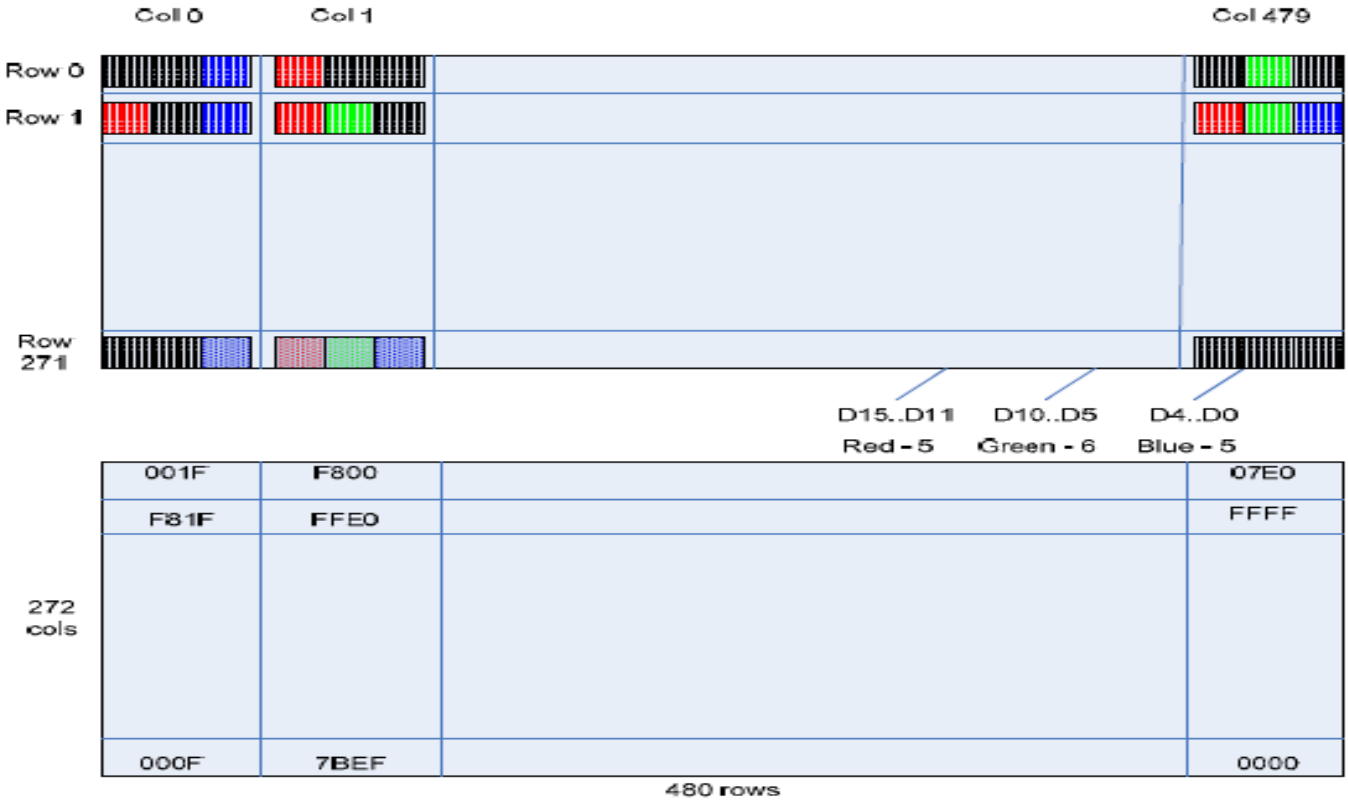
1. ח . דוגמה לנתוני חוצץ המסגרת ומיפוי ה LCD

באיור 9 רואים כיצד יושבים הנתונים בחוצץ המסגרת. מדובר בתצוגה של 272*480 , כלומר 272 שורות ו 480 עמודות עם 16bpp , כלומר 16 ביטים לפיקסל המסודרים בתבנית RGB565 , כלומר 5 ביטים לאדום , 6 ביטים לירוק ו 5 ביטים לכחול.

באיור העליון רואים את הצסע בכל פיקסל. באיור התחתון רואים את הנתון הנמצא בכל כתובת.

מהאיור רואים ש 16 הביטים הראשונים שייכים לשורה 0 ועמודה 0 . 16 הביטים הבאים שייכים לורה 0 עמודה 1 וכך הלאה. כאשר מסתיימים 16 הביטים של עמודה 479 מתחילים 16 הביטים של שורה אחת.

5 הביטים D0~D4 שייכים לצבע האדום. 6 הביטים D5~D10 שייכים לירוק ו 5 הביטים D11~D15 שייכים לצבע הכחול.



איור 9 - מיפוי הזיכרון בחוצץ המסגרת.

לדוגמה : בשורה 0 ועמודה 0 רואים שהפיקסל בצבע כחול. הנתון שנמצא בכתובת המתאימה בחוצץ הוא : 001FH . בבינארי זה : D0..... D4 D5 .....D10 D11.....D14 D15

0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1

, כלומר לצבע האדום יש 5 אפסים , לירוק 6 אפסים ולכחול חמש פעמים 1 . מכאן שנקבל צבע כחול.

דוגמה נוספת : בשורה השנייה , בעמודה השנייה רואים את הצבעים אדום וירוק. בזיכרון נמצא הנתון : FFE0H שהוא בבינארי : 1111 1111 1110 0000 . 5 הביטים הגבוהים של האדום הם '1' . 6 הביטים הבאים של '1' הם של הירוק ו 5 הביטים הנמוכים הם '0' .

1.ט . עומק הצבע וטבלאות המרה

עומק צבע חוצץ המסגרת ועומק צבע ה LCD לא צריכים להיות שווים. ניתן להשתמש בטבלת המרה הממפה תת קבוצה לערכים של צבע LCD. לדוגמה : הבקר יכול להיות מתוכנת ל 4bpp (4 ביטים לפיקסל) בעוד שממשק ה LCD הוא 16bpp . במקרה כזה נוכל להשתמש ב 16 צבעים שונים. באיור 10 ניתן לראות כיצד עוברים מ 3bpp בעזרת ל 8 צבעים שונים על ידי המרה ל RGB565 . ניתן לראות שעבור 3 ביטים של 0 נקבל צבע שחור (16 אפסים). מצב 001 נותן 0X001F שהוא צבע כחול וכך הלאה עד הלבן שבמצב של 111 נקבל 0XFFFF שבו כל 16 הביטים הם 1 .

Frame buffer value	Lookup table value (16-bit RGB565 output)	System color	
R0, G0, B0 (0)	0x0000	black	
R0, G0, B1 (1)	0x001F	blue	
R0, G1, B0 (2)	0x07E0	green	
R0, G1, B1 (3)	0x07FF	cyan	
R1, G0, B0 (4)	0xF800	red	
R1, G0, B1 (5)	0xF81F	magenta	
R1, G1, B0 (6)	0xFFE0	yellow	
R1, G1, B1 (7)	0xFFFF	white	

איור 10 : טבלת המרה של המרה מ 3bpp ל RGB565

}

2. הבקר HX8357-D00/D01

רכיב הבקר תומך ברזולוציה HVGA - Half size Video Graphics Array – מערך חוזי גרפי בגודל חצי - ונותן פתרון של רכיב אחד המשלב דרייבר דוחף זרם ומעגל ספק כוח המתחבר אל TFT-LCD עם 320*480 פיקסלים.

קיימים בקרים ל QVGA – Quarter VGA (רבע VGA) מתאים לתצוגות 240*320 פיקסלים. הרכיב יכול לפעול עם מתח נמוך והמעגלים שלו ייצרו את המתחים להפעלת התצוגה. בנוסף לכך יש אפשרות להוריד את תצרוכת ההספק של בעזרת תוכנה. דוגמה לבקר כזה הוא ILI9325 שעובד על עיקרון עבודה דומה. בקר נוסף הוא ILI9481 .

הרכיב תומך במספר אופני ממשק הכוללים MPU MIPI DBI מטיפוס B ואופני ממשק MIPI (Mobile Industry Processor Interface - ממשק מעבד לתעשיית הניידים), DBI – (Display Bus Interface - ממשק תצוגת פס) ו DPI (ממשק תצוגת פיקסלים – Display Pixel Interface) מטיפוס C . גרסת D01 של הבקר מאפשרת ממשק MIPI DSI (Display Serial Interface) – ממשק תצוגה טורי. אופני הממשק נבחרים בעזרת חומרה חיצונית המתחברת להדקים IM0 IM1 IM2. הרכיב מתאים להפעלה גם מסוללה במערכות טלפון סלולארי .

3. תכונות

3.1 תצוגה

*** רזולוציה**

320(H) x RGB(H) x 480(V) -- H– אופקי , V– אנכי . RGB-Red Green Blue.

- **Display Color Modes – אופני עבודה של תצוגת הצבע**

א. Normal Display Mode ON – אופן תצוגה רגיל

1. ממשק DBI (Display Bus Interface - תצוגת ממשק פס).

א. 65K צבעים (5 ביטים לאדום 6 לירוק ו 5 לכחול 2^{16}= 65536 צבעים שונים RGB 565).

ב. 262K צבעים (6 ביטים לכל צבע 2^{18} = 262144 צבעים שונים).

ג. 24 ביטים של צבע וירטואלי (8 ביטים לכל צבע 2^{24} = 16777216 צבעים שונים - 16M).

2. ממשק DPI (ממשק תצוגת פיקסלים – Display Pixel Interface)

א. 65K צבעים (5 ביטים לאדום 6 לירוק ו 5 לכחול).

ב. 262K צבעים (6 ביטים לכל צבע).

ג. 16M צבעים (8 ביטים לכל צבע).

3. ממשק MIPI (Mobile Industry Processor Interface - ממשק מעבד לתעשיית הניידים רק לרכיב HX8357/D01)

א. 65K צבעים (5 ביטים לאדום 6 לירוק ו 5 לכחול).

ב. 262K צבעים (6 ביטים לכל צבע).

ג. 24 ביטים של צבע וירטואלי (8 ביטים לכל צבע 16M צבעים שונים).

3.1 . אופן סרק IDLE mode on.

8 צבעים (ביט 1 לכל צבע).

3.2 - מודול התצוגה

- אזור מסגרת התמונה 320 אופקי x 480 אנכי x 18 ביט .
- במודול קיים ממיר DC ל DC
- מתחי הפעלה שונים :

- VSP = 5.2 V for two time pump (Power supply for driver circuit range)
- VSP = 6.2 V for three time pump (Power supply for driver circuit range)
- VSN = -5.2 V for two time pump (Power supply for driver circuit range)
- VSN = -6.2 V for three time pump (Power supply for driver circuit range)
- VSPR =3.3V to 5.8V (Positive Source output voltage range)
- VSNR= -3.3V to -5.8V (Negative Source output voltage range)
- VGH = +10.0 to +15.3V (Positive Gate output voltage range)
- VGL = -7.5 to -12.8V (Negative Gate output voltage range)
- VCOM=-2.5V to 0V

3.3 ממשק הבקרה לתצוגה

- ממשק מקבילי MIPI-DBI של 8/9/16/18 ביטים.
- ממשק העברת נתונים טורי MIPI-DBI
- ממשק חוזי מקבילי של פיקסלים MIPI-DPI של 8/16/18/24 קווי נתונים
- ממשק טורי MIPI-DSI (לוורסיות 1.0.1 ו 1.00).

3.4 - כניסות ספק

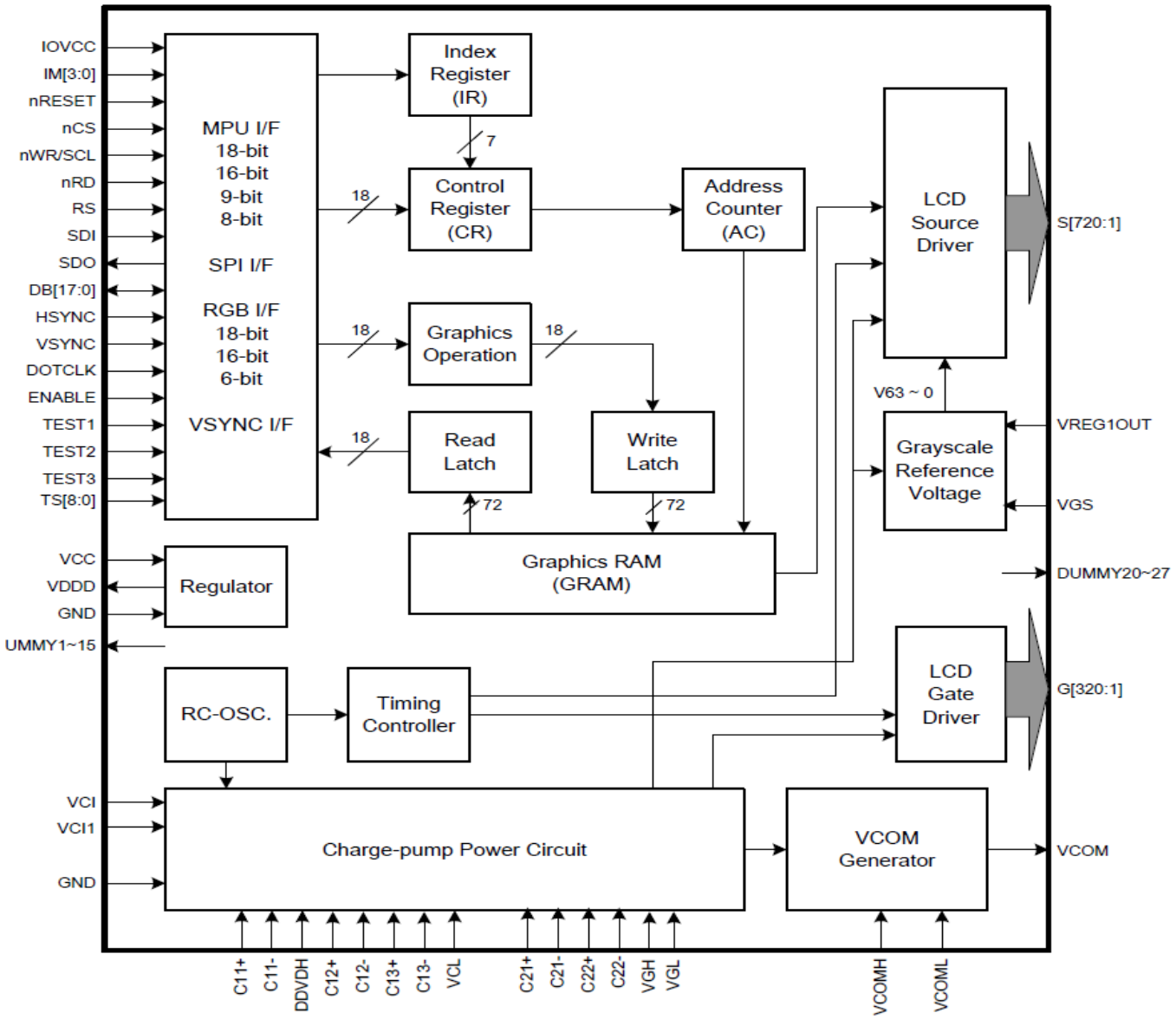
- ספק כוח ללוגיקה (IOVCC) : 1.65V ~ 3.3V
- ספק כוח אנלוגי (VCI) : 2.5V~3.3V

3.5 – שונות

- תצרוכת הספק נמוכה. מתאים למערכות המופעלות על ידי סוללה.
- פונקציית מניעת image sticking (שרואים שאריות של מסך קודם כשמכבים את התצוגה).
- כניסות תואמות CMOS.
- פתרון של רכיב אחד עבור COG assembly (Chip On Glass). אין צורך להדביק רכיבים נוספים למסך.
- תמיכה בכניסת פונקציית image של 24 ביט .
- תמיכה ב 1/2/4 נקודות לעמודה להיפוך / היפוך זיג זג
- תמיכה ב"גלגול" אזור.
- תמיכה באופן תצוגה חלקי.
- תמיכה במצב standby עמוק.
- תמיכה במצב normal black/normal white .
- תמיכה בתצוגת מבט זוויות רחב (ניתן לראות את התצוגה בזוויות גדולות – לא רק מלמעלה).
- אפשרות לתכנות חד פעמי (One Time Programming-OTP) ותכנות 4 פעמים של ה ID ו רגיסטר VCOM.
- ספק כוח לצריבה חד פעמית ברכיב עצמו.
- תמיכה בפונקציית בקרת בהירות מסתגלת .
- תמיכה בפונקציית העשרת צבעים.
- תמיכה בפונקציית 3-Gamma דיגיטאלית.
- טמפרטורת הפעלה מ 40- עד 85 מעלות צלסיוס.

4. א. סכמת המלבנים של בקר התצוגה

כאמור, הבקר מתחבר מצד אחד אל מעבד ומצדו השני אל פנל הזכוכית עם הסגמנטים. באיור 9 מתוארת הסכמה המלבנית של בקר התצוגה.



איור 9 : סכמת המלבנים של הבקר

4.ב - תיאור ההדקים

בצד שמאל למעלה יש את 2 ההדקים אליהם מתחברים מתחים של ספק הכוח. האדמות שלהם נמצאות למטה משמאל.

IOVCC3 – לכאן מתחבר מתח ספק הכוח הדיגיטאלי. **VSSD** – האדמה של המתח הדיגיטאלי.

VCI – לכאן מתחבר המתח של ספק הכוח האנאלוגי. **VSSA** – האדמה של המתח האנאלוגי .

1.ב.4 - הדקי ממשק לוגי

IM0 IM1 IM2 - 3 קווים המתחברים אל מלבן ה MPU בחלקו התחתון והם בוררים את ממשק המערכת לפי הטבלה הבאה :

IM2	IM1	IM0	Interface
0	0	0	DBI TYPE-B 18-bit/24-bit
0	0	1	DBI TYPE-B 9-bit
0	1	0	DBI TYPE-B 16-bit
0	1	1	DBI TYPE-B 8-bit
1	0	0	Not use
1	0	1	DBI TYPE-C Option 1
1	1	0	MIPI DSI (For HX8357-D01 only)
1	1	1	DBI TYPE-C Option 3

טבלה 1 : בחירת ממשק המערכת.

במצב '1' ההדק מתחבר אל IOVCC ובמצב של '0' ל VSSD .

הערה : במודול התצוגה שקונים באינטרנט במחיר של כ 30 ₪ החיבור הוא 011 , כלומר : 8 bit DBI TYPE – B . - ממשק תצוגת פס של 8 ביטים.

Chip Select - **CSX** - קו בחירת הרכיב. אם נשים 0 בהדק – זה אומר שניתן לפנות לרכיב. 1 – לא ניתן לדבר עם הרכיב.

WRX_SCL - הדק הכתיבה – WR - לרכיב במצב עבודה TYPE B או הדק השעון הטורי SCL - במצב - TYPE C . היות ואנחנו עובדים ב TYPE B אז זהו הדק הכתיבה. עליה מ 0 ל 1 מבצעת את הכתיבה של הפקודה או הנתון לתוך הרכיב.

RDX - הדק הקריאה – RD – מהרכיב. גם כאן המעבר מ 0 ל 1 מבצע את הקריאה מפס הנתונים/פקודה של הרכיב.

DCX - בעזרת הדק זה קובעים האם מדובר בפקודה (COMMAND) או נתון (DATA). אם שמים 0 – פונים לרגיסטר הפקודה. אם שמים 1 - פונים לרגיסטר הנתון.

VSYNC - Vertical Syncronizing - אות כניסת סנכרון אנכי במצב DPI (ממשק תצוגת פיקסלים). אנחנו לא נשתמש בהדק.

HSYNC - Horizontal Syncronizing - אות כניסת סנכרון אופקי במצב DPI (ממשק תצוגת פיקסלים).

DE - Data Enable - אות אפשור נתון באופן DPI I/F (ממשק תצוגת פיקסלים) .

PCLK - אות אפשור נתון באופן DPI .

RESX – RESet pin – הדק איפוס. פעיל בנמוך .

DIN_SDA - הדק הנתון הטורי בממשק פס טורי . הנתון נכנס בעלייה של השעון.

DOUT – יציאת הנתון הטורי. אם SDO_EN=0 ההדק לא בשימוש. אם SDO_EN=1 ההדק הוא יציאה של נתון טורי.

TS4_DB23 TS3_DB22 TS2_DB21 TS1_DB20 TS0_DB19 DB18 DB17~0

24 ביטים דו כיווניים של פס הנתונים.

4.ב.2 . יציאות

S1~S961 - מתחי יציאה המסופקים אל הגביש הנוזלי

G1~G480 - הדקי יציאה של שער הדוחפים. בהדקים יש מתח גבוה או נמוך VGH או VGL בהתאמה.

VCOM – מתח ספק הכוח המשותף של ה TFT . מתחברת אל האלקטרודה המשותפת בפאנל של ה TFT .

TE - Tearing Effect

CABC_PWM1 – יציאת אות בקרת PWM של תאורת הרקע backlight מתחבר אל ה MPU.

CABC_PWM2 - יציאת אות בקרת PWM של תאורת הרקע backlight מתחבר אל מעגל תאורת רקע.

CABC_ON - יציאה אל דוחף הזרם של לד תאורת הרקע. לא לחבר אם לא משתמשים בהדק זה.

4.ב.3 . כניסות ויציאות

C11A,C11B,C13A,C13B - לחבר אל קבלי ה step-up (קבלי האצה) בפעולת מעגל 1 (VSP).

C12A,C12B,C14A,C14B - לחבר אל קבלי ה step-up בפעולת מעגל 3 (VSN).

C22A,C22B - לחבר אל קבלי step-up במעגל מספר 1 (VGH/VGL).

4.ב.4 חלקי ספק כוח

IOVCC - זהו פד של ספק הכח ל IO הדיגיטאליים.

VCI – ספק הכוח האנלוגי

VSSD – האדמה הדיגיטאלית.

VSSA – האדמה האנלוגית.

VDDD – למתחים הלוגיים הפנימיים.. לחבר אל קבלי הייצוב.

NVDDDOUT – עבור מתח לוגי פנימי.

VSPROUT – מתח מחולל כוח פנימי מיוצב ליחידת דחיפת זרם מקור.

VSNROUT - מתח מחולל כוח פנימי מיוצב ליחידת דחיפת זרם מקור.

VSP – יציאה ממעגל האצה 1 . יש לחבר קבל ייצוב בין הדק VSP ל VSSA.

VSN – יציאה ממעגל האצה 3 . יש לחבר קבל בין VSN ו VSSA .

VGH – יציאת ספק חיובית ממעגל האצה 2 עבור מעגל דחיפה לשער .

VGL – יציאת ספק שלילית ממעגל האצה 2 עבור מעגל דחיפה לשער.

VCOMDC – יציאה מרמת VCOM

4.ב.5 . ממשק High Speed

HIS_DP , HIS_DN - הדקי קלט/פלט של ממשק אות הפרשי במהירות גבוהה.

HIS_CP , HIS_CN - הדקי השעון של ממשק אות הפרשי במהירות גבוהה.

VSS_HSI - האדמה האנלוגית של ממשק אות הפרשי במהירות גבוהה. כאשר משתמשים בשיטת GOC יש לחבר את VSSA של FPC כדי למנוע רעש.

LDO_HSI – הדק יציאה של מייצב ממשק מהירות גבוהה. מחוברת לקבל ייצוב בין VSS_HSI ו HIS_LDO .

4.ב.6 . הדקי בדיקה ועוד.

TEST1 – הדק כניסה לבדיקה (יש לו pull down פנימי).

TEST2 – הדק כניסה לבדיקה (יש לו pull down פנימי).

TEST3 – הדק כניסה לבדיקה (יש לו pull down פנימי).

OSC – הדק כניסה לבדיקה.

TESTDP – הדק יציאה לבדיקה.

VTEST – הדק יציאה לבדיקה

NVTEST – הדק יציאה לבדיקת מתח ה GAMMA של התצוגה.

DUMMY1~30 , DUMMY31~69 - פדים דמה.

DUMMY_VCL – פדים דמה.

4.ב.7 - הסבר הסכמה המלבנית

נסביר את המלבנים העיקריים.

1. MPU I/F - בבקר יש מיקרו פרוססור (בצד שמאל) השולט על כל הפעולות בתוך הבקר . הוא תומך ב 3 מערכות שונות של ממשק מהיר :

א. ממשק פס מקבילי של 8 ביט, 9 ביט, 16 ביט , 18 ביט ו 24 ביט.

ב. ממשק טורי היקפי – SPI - **S**erial **P**eripheral **I**nterface - של 16 ביט , 18 ביט ו 24 ביט.

ג. ממשק DBI (Display Bus Interface - ממשק פס תצוגה) .

הממשק נבחר בעזרת 3 הדקים של הרכיב : IM2 IM1 IM0 כמתואר בטבלה 1 שבעמודים קודמים.

2. INTERNAL REGISTER - מימין למיקרו פרוסור. זהו רגיסטר בן 16 ביט השומר אינפורמציה של אינדקס מרגיסטרי הבקרה ומה GRAM הפנימי.

3. GRAM CONTROL - זהו רגיסטר של 18 ביט המאחסן נתונים זמניים שאותם כותבים ל GRAM או קוראים מה GRAM.

4. **GRAM** – Graphic RAM – זיכרון RAM גרפי. זהו הזיכרון שבו שומרים את הצבעים עבור כל פיקסל של התצוגה. גודל הזיכרון בבייטים הוא : (ה 24 בחישוב זה מספר הביטים המקסימלי לפיקסל)

480 * 320 * 24 / 8 = 460800 Bytes

5. MODE SELECTION - בחירת אופן . בעזרת מלבן זה קובעים את

6. GRAYSCALE VOLTAGE GENERATOR - מחולל מתח של סקלת האפור – (בצד ימין של הסכמה) . מעגל זה יוצר מתח דחיפה לגביש הנוזלי בהתאמה לסט הנתונים של סקלת האפור במעגל Gamm adjusting circuit שמתחתיו – מעגל כיוון גמא - (הסבר בסעיף הבא) . המחולל יכול להוציא 64 רמות מתח שונות של עצמת הארה עבור כל אחד מ 3 הצבעים (Red Green Blue).

grayscale מוגדר כטווח של גוונים של אפור ללא צבע גלוי. הגוון החשוך ביותר הוא שחור שהוא חוסר של אור מוקרן או מוחזר. הגוון המואר ביותר הוא לבן שבו יש שידור או החזרה מלאים של אור בכל אורכי הגל הנראים לעין.

7. **GAMMA ADJUSTING CIRCUIT** – מעגל כיוון גמא. היות והגרף של עצמת ההארה של צבע מסוים בפיקסל כתלות במתח המסופק לאותו סיגמנט איננו ליניארי, נותנים סט של נתונים למעגל **Digital Gamma Correction** (משמאל למלבן grayscale voltage generator) כדי "ליישר" את האופיין.

8+9. TIMING CONTROL – בקרת תזמון – זהו מעגל היוצר אותות תזמון לפעולת המעגלים הפנימיים כמו ה GRAM וכמו יחידת בקרת השער **GATE CONTROL UNIT** - של מערך הטרנזיסטורים בסיגמנט שהיציאות שלו הולכות אל השערים של טרנזיסטורי ה FET מ G1~G480 .

10+11+12. DATA LATCH - נעילת נתונים - הנתונים ממעגל תיקון הגמא ננעלים בנועלים אלו ועוברים אל מעגל **D/A converter circuit** - מעגל ממיר מדיגיטאלי לאנאלוגי - ההופך את המספר הדיגיטאלי שהתקבל למתח אנלוגי העובר אל **SOURCE DRIVER** שהם ההדקים המתחברים אל ה source של טרנזיסורי ה FET בסגמנט . סה"כ יש לנו 320*3 = 960 הדקי source מ S1~S960.

13. RC OSC - מתנד RC - בצד שמאל למטה - מעגל מתנד המורכב מנגד וקבל ליצירת אותות תזמון לרענון מסגרת התמונה.

14 . DC / DC CONVERTER - ממיר מ DC ל DC - ממיר את המתח של הספק למספר מתחים נוספים הנדרשים להפעלת ה LCD . בין השאר יש להעלות את המתח של ה 3.3 וולט למתח של כ 29 וולט למעגל ההארה האחורית backlight.

15. REFERENCE VOLTAGE - מתח ייחוס - מעגל ספק נוסף למתחים עבור ה source של טרנזיסטורי ה FET .

16. CABC FUNCTION - Content Adaptive Brightness Control – בקרת הארה מותאמת תוכן - (למעלה במרכז האיור ו 2 מלבנים מתחתיו) שהם מתייחסים לתצוגה האחורית - backlight - של התצוגה. ניתן להפעיל אותה או לכבות אותה בעזרת הדק CABC_ON וכמו כן ניתן לעבוד איתה ב Pulse Width Modulator – PWM - אפנון רוחב דופק – לקביעת עצמת ההארה האחורית.

5. ממשק

הבקר HX8357 תומך בממשקי **MIPI** . MIPI - **M**obile **I**ndustry **P**rocessor **I**nterface

כלומר ממשק מעבד לתעשיית הניידים (לטלפונים ניידים) :

DBI - Display Bus Interface - ממשק תצוגת פס .

DPI – Display Pixel Interface - ממשק תצוגת פיקסלים .

DSI – Display Serial Interface - ממשק תצוגה טורית .

ממשק תצוגת פס – DBI , תומכת בממשק של 24 , 18 , 16 , 9 ו 8 ביטים. הממשק מקבילי TYPE B וממשק טורי TYPE C עם אופציות 1 או אופציה 3 . הממשק נקבע על ידי ההדקים IM2-0 כמתואר בטבלה מספר 2 המתארת את בחירת הממשק:

IM2	IM1	IM0	Interface	WRX_SCL	Data Bus use	
					Command/ Parament	GRAM
0	0	0	DBI TYPE-B 18-bit (DB_EN='0')	WRX	DB7-DB0	DB17-DB0: 18-bits Data
0	0	0	DBI TYPE-B 24-bit(DB_EN='1')	WRX	DB7-DB0	DB23-DB0: 24-bits Data
0	0	1	DBI TYPE-B 9-bit	WRX	DB7-DB0	DB8-DB0: 9-bits Data
0	1	0	DBI TYPE-B 16-bit	WRX	DB7-DB0	DB15-DB0: 16-bits Data
0	1	1	DBI TYPE-B 8-bit	WRX	DB7-DB0	DB7-DB0: 8-bit Data
1	0	0	Not use	-	-	
1	0	1	DBI TYPE-C Option 1	SCL	SDA	
1	1	0	MIPI DSI (For HX8357-D01 only)	-	HSI_CP/N , HSI_D0P/N	
1	1	1	DBI TYPE-C Option 3	SCL	SDA	

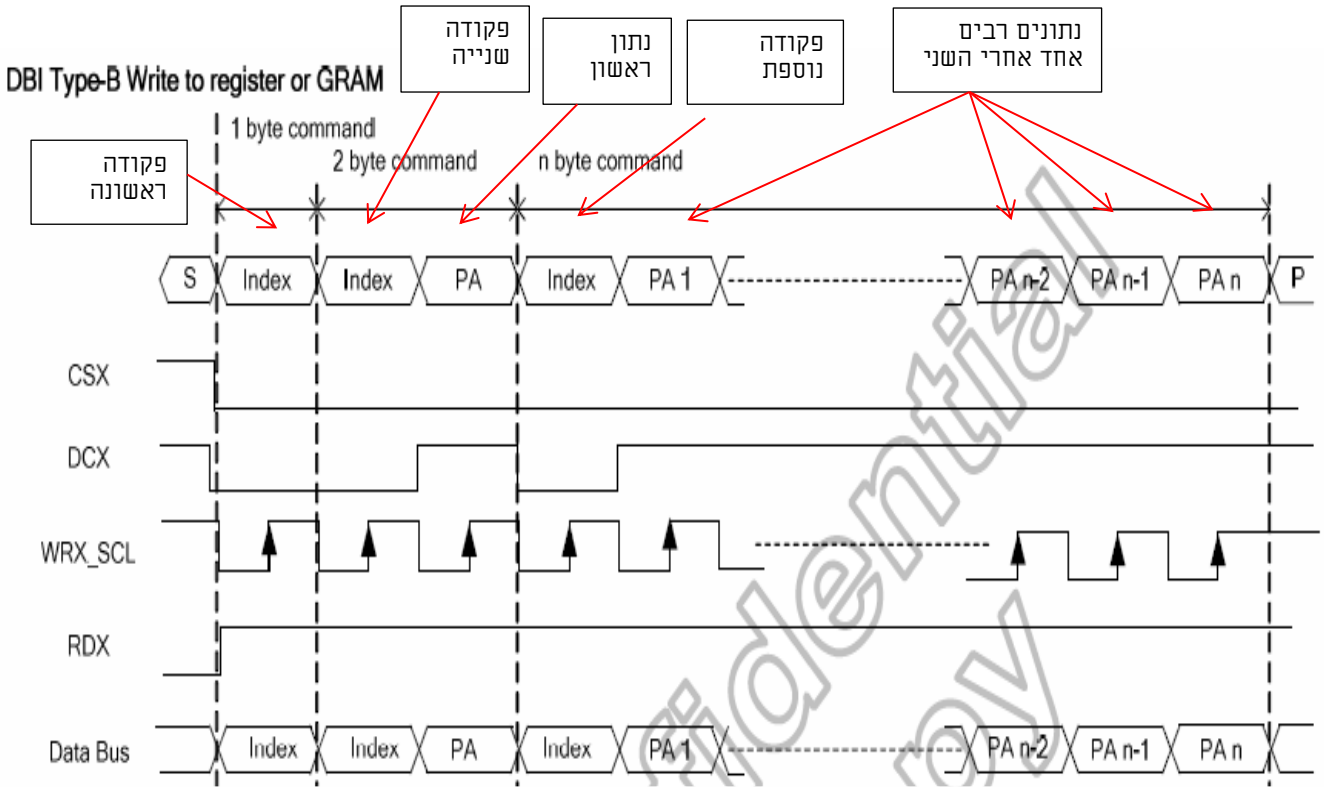
טבלה 2 : בחירת הממשק

בבקר יש רגיסטר אינדקס - Index Register – IR שמאחסן את נתוני האינדקס של רגיסטר הבקרה הפנימי ושל ה GRAM (Graphic Ram) רגיסטר זה דומה לתפקיד רגיסטר הפקודה בתצוגת LCD "רגילה" . כאשר בהדק DCX (הדק הבוחר פקודה – 0 או נתון -1) נשים 0 , הפקודה דרך ממשק ה DBI נכתבת לרגיסטר. כאשר נשים 1 בהדק ה DCX אז נתונים נכתבים מפס הנתונים ל GRAM דרך רגיסטר R2Ch . כאשר הנתונים נכתבים ל GRAM מהמעבד , הנתון נכתב תחילה לנועל כתיבה – write data latch , ואז באופן אוטומטי נכתב ל GRAM על ידי פעולות פנימיות.

במצב קריאת נתון , הנתון שנקרא מה GRAM עובר תחילה דרך נועל קריאה read data latch .
בפעול הקריאה הבאה הנתון יעבור מנועל הקריאה לפס הנתונים ולמעבד. כך שפעולת הקריאה הראשונה
איננה תקפה ונקראת דמה - DUMMY והיא תקפה רק מפעולת הקריאה השנייה.

5 .1.א - ממשק MIPI DBI-B - מחזור כתיבה

הממשק ה DBI נקבע בעזרת ההדקים בחירת הממשק IM2~0 . דיאגרמת הזמנים של הממשק
המקבילי עבור מחזור כתיבה מתוארים באיור 10 .



איור 10 : פרוטוקול ממשק DBI B . כתיבה לרגיסטר או ל GRAM.

באיור 10 רואים שכדי לכתוב פקודה או נתון יש להוריד את הדק CSX (בחירת רכיב) ל 0 . רואים
זאת בצורת הגל העליונה מיד אחרי מלבן ה S (Start – התחלה). בסוף צורת גל זו רשום P (Pause
– עצור והכוונה סיום).
בעזרת ההדק DCX אומרים לתצוגה האם כותבים לה פקודה (0) או נתון (1) .
הדק RDX (קריאה) חייב להיות ב 1 כל זמן הכתיבה.

הפקודה או הנתון ננעלים לתצוגה בעליית ההדק WRX_SCL (הדק הכתיבה).

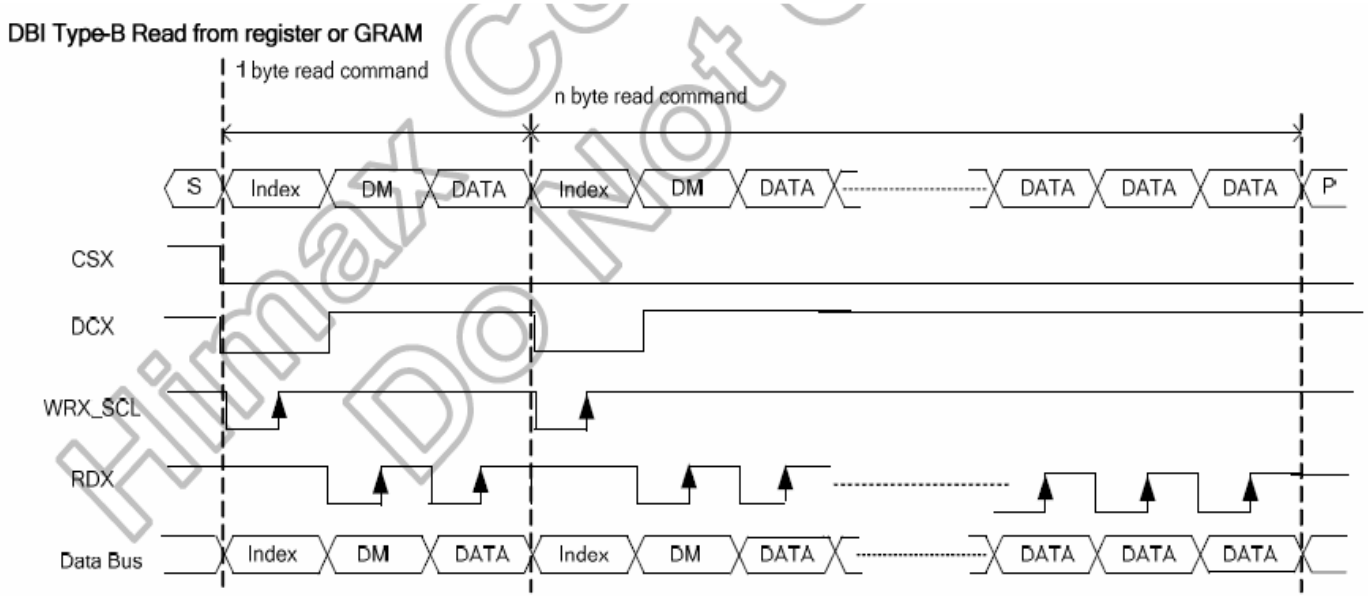
באיור רואים שניתן לשלוח פקודה (Index) ראשונה ולאחריה פקודה (Index) שנייה נוספת. כל פקודה ננעלת על ידי העלאת קו הכתיבה מ 0 ל 1 .

לאחר מכן מראים שניתן לשלוח נתון ראשון (PA - קיצור של פרמטר). עושים זאת על ידי העברת ההדק DCX ל 1 ושוב העלאת קו הכתיבה מ 0 ל 1.

בהמשך רואים שלאחר הפקודה השנייה ניתן לשלוח פרמטרים (נתונים) אחד אחרי השני. הם ננעלים לתוך הבקר בעליית הדק הכתיבה מ 0 ל 1 .

5 . 1.ב - ממשק MIPI DBI-B - מחזור קריאה

דיאגרמת הזמנים של הממשק המקבילי עבור מחזור קריאה מתוארים באיור 11.



איור 11 : פרוטוקול ממשק DBI B . קריאה מרגיסטר או מה GRAM.

מאיור 11 ניתן לראות שיש להוריד את הקו CSX ל 0 כדי להודיע לרכיב שהוא פעיל.

בהתחלה כותבים לבקר פקודה (קו DCX ב 0) . הפקודה אומרת מה רוצים לקרא. האם מצב רגיסטר כלשהו או נתונים מה GRAM .

לאחר מכן שולחים פקודת קריאה (העלאת קו הקריאה RDX מ 0 ל 1 . קריאה ראשונה היא תמיד קריאת דמה (DUMMY) והקריאה השנייה מעבירה את הנתון אל המעבד.

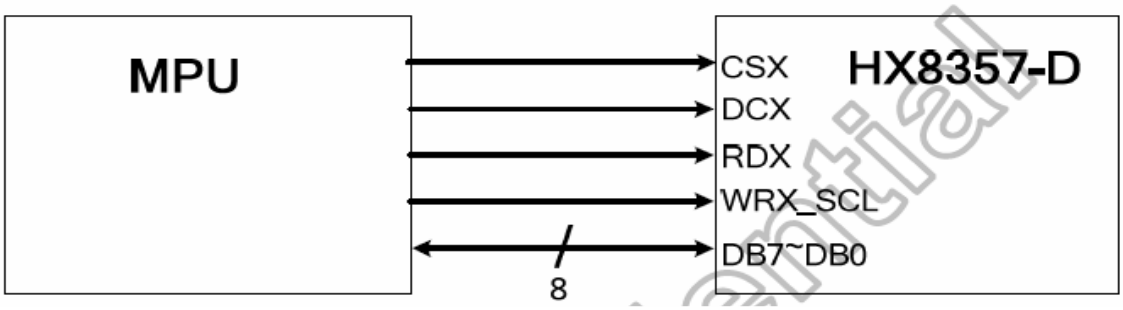
לאחר מכן מראים קריאה של נתונים אחד אחרי השני. גם כאן אחרי שליחת הפקודה (Index) יש קריאת דמה אחת ואח"כ קריאה רצופה של נתונים (ללא קריאת דמה נוספות).

2. 5 - ממשק DBI TYPE B - ממשק מקבילי עם 8 ביט .

במצב 011 של ההדקים IM2-0 עובדים עם ממשק DBI של 8 ביט .

כאשר מחברים בין מעבד (MPU) ובקר תצוגה גרפית HX8357-D בחיבור מקבילי של 8 ביט אז בנוסף לפס הנתונים מתחברים עוד 4 קווי בקרה.

באיור מספר 12 מתוארים חיבור ההדקים בין מעבד ובקר התצוגה .



איור 12 - ממשק מקבילי DBI TYPE B עם 8 ביט .

הדק CSX – הדק בחירת הרכיב - Chip Select - המעבד קובע האם הבקר פעיל (0) או לא (1).

הדק DCX - הדק שבו המעבד אומר לרכיב האם הוא שולח פקודה (0) או נתון (1).

הדק RDX – הדק הקריאה . המעבד שם בהדק 1 כאשר איננו קוראים מהרכיב. במעבר מ 0 ל 1 מתבצעת קריאה (בתנאי שהדק CSX=0).

WRX_SCL – הדק הכתיבה של הרכיב. כאשר לא מתבצע מחזור כתיבה ההדק ב 1 . במעבר מ 0 ל 1 מתבצעת כתיבה(בתנאי שהדק CSX=0).

הדקים DB7~DB0 – 8 ההדקים שעליהם יש את הפקודה או הנתון.

1.2. 5 ממשק פס נתונים של 8 ביט לפורמט של 16 ביטים לפיקסל (RGB 5 6 5) עבור 65K צבעים

הפקודה שיש לשלוח כדי לעבוד עם פיקסלים היא Interface Pixel Format . הפקודה הנשלחת היא 3Ah (set_pixel_format) ולאחריה נשלח פרמטר. הערך שבפרמטר מאפשר עבודה עם 16 או 18 או 24 ביט לפיקסל. בדרך כלל עושים זאת בהתחלת העבודה באתחול של התצוגה. אם הערך של הפרמטר הוא 05H נקבל תבנית צבעים של 16 ביט. $2^{16} = 65536$ צבעים שונים . אם נשלח פרמטר 06H נקבל תבנית צבעים של 18 ביט (6 ביטים לכל צבע) כלומר $2^{18}= 262144$ צבעים שונים (או בקרוב 262K) אם נשלח 07H נקבל תבנית צבעים של 24 ביט שהם $2^{24} = 16777216$ תבניות צבע שונות (או בקרוב 16M).

באיור 13 רואים כיצד שולחים את הצבעים לפיקסל כלשהו .היות ועובדים עם ממשק של 16 ביט אז הצבעים הנשלחים הם בפורמט R5G6B5.

מהאיור רואים שבהתחלה שולחים את הפקודה 2CH שאומרת שכותבים לזיכרון המסגרת (frame memory), כלומר ל RAM עבור הפיקסל המסוים. ההדקים של התצוגה מקבלים :

הדק של ה RESX (איפוס) חייב להיות ב '1' (לא פעיל).

הדק CSX - בחירת רכיב – שמים ב 0 כדי להגיד לתצוגה שהיא פעילה.

בהתחלה הדק DCX ב 1 כדי להגיד שמוציאים פקודה. לאחר מכן ההדק ירד ל 0 כאשר נוציא נתונים.

הדק WRX – הדק הכתיבה – יעלה ל '1' בכל פעם שנרצה לכתוב. העלייה מ '0' ל '1' נועלת את הפקודה או הנתון .

לאחר הוצאת הפקודה קו DCX עולה ל '1' ומציין שמוציאים נתונים. 5 הביטים הגבוהים בנתון הראשון הם של האדום (RED) ו 3 הביטים הנוספים (הנמוכים) הם של הירוק. בנתון הבא 3 הביטים הגבוהים הם המשך הביטים של הירוק ועוד 5 הביטים הנמוכים הם של הכחול. 2 הבתים הבאים שבאיור הם הצבעים עבור הפיקסל הבא.

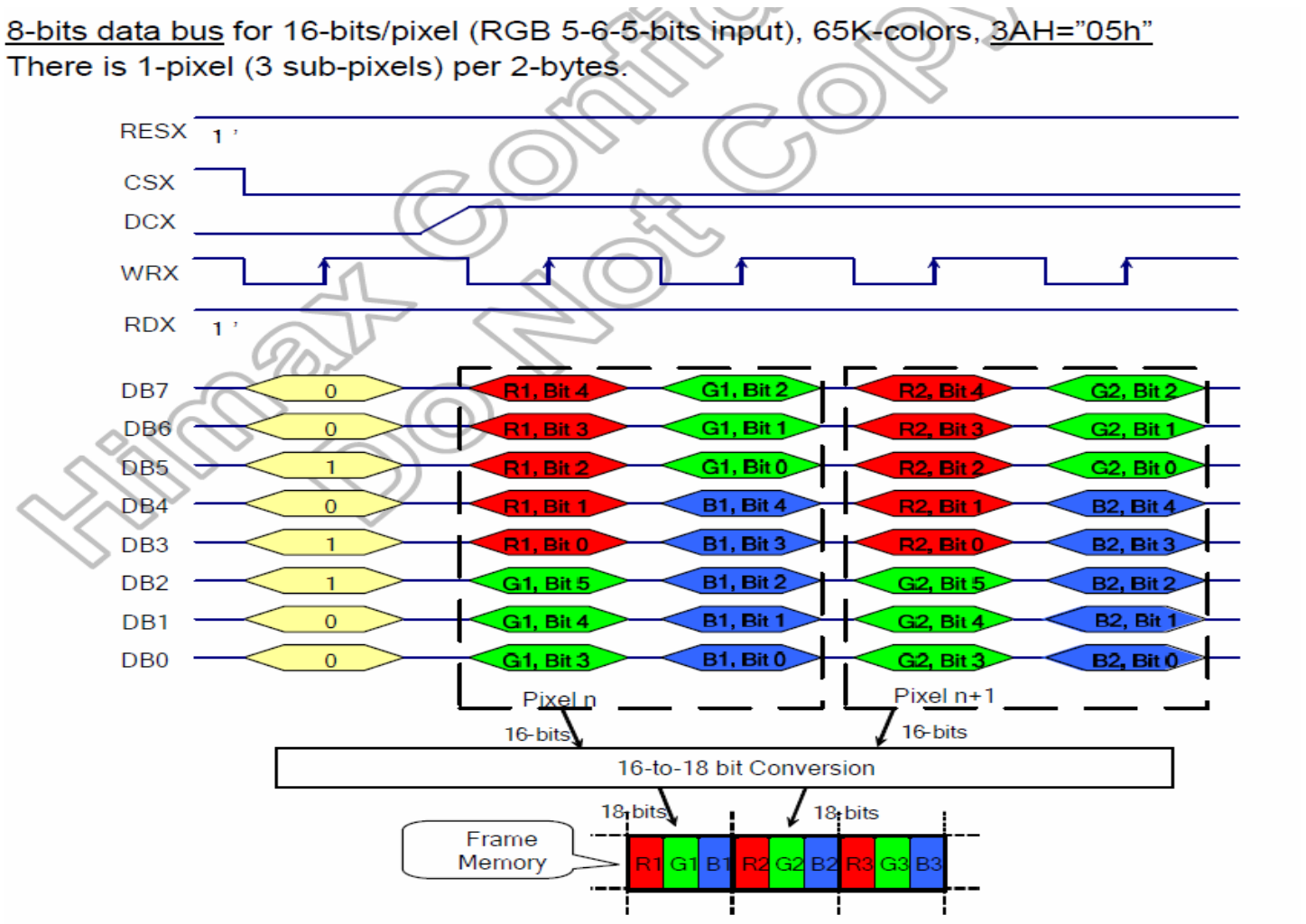


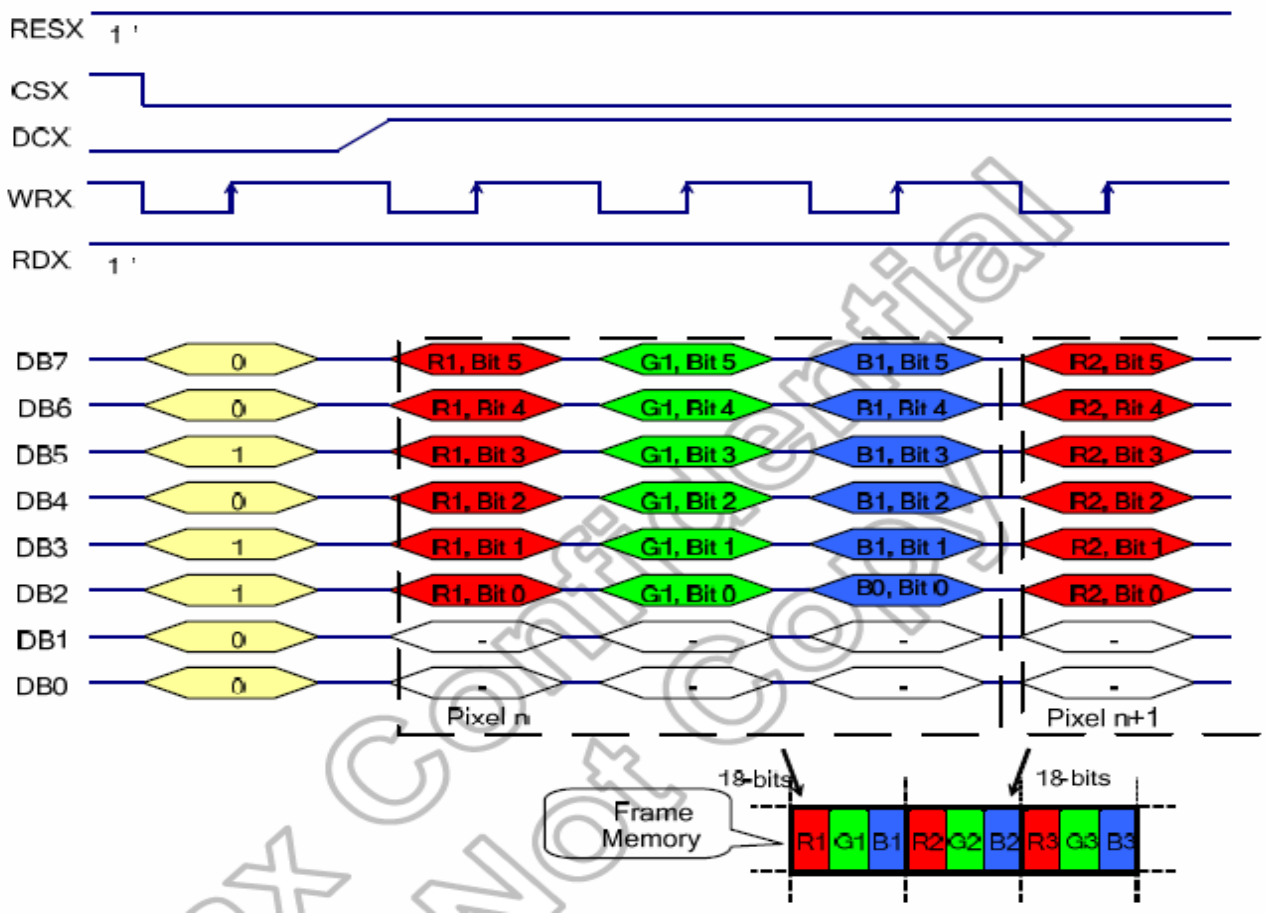
Figure 4.4: Write data for RGB 5-6-5 (65k colors) bits input in 8-bit parallel interface

איור 13 : ממשק פס נתונים של 8 ביט עם פורמט של 16 ביטים של צבע.

1. 3.5 - ממשק פס של 8 ביט עם פורמט של 18 ביטים לפיקסל

במקרה הזה אחרי הפקודה 2CH שאומרת לכתוב לזיכרון מוציאים 3 בייטים של הצבעים. כל בייט הוא 8 ביטים . מבנה הצבעים R6 G6 B6 . סה"כ ניתן לקבל $2^{18} = 262144$ צבעים שונים. הבייט הראשון הוא של 6 ביטים של הצבע האדום והביטים D0 D1 הם ללא תפקיד. בבייט הבא שולחים שוב 6 ביטים גבוהים עבור הצבע הירוק וגם כאן 2 הביטים הנמוכים לא נחשבים. גם בבייט השלישי, של הכחול 6 הביטים הגבוהים קובעים את הצבע ו 2 הביטים הנמוכים ללא תפקיד.

איור 14 מתאר את הוצאת הצבעים במקרה הזה:



איור 14 : ממשק פס של 8 ביט עם פורמט של 18 ביטים (RGB 6 6 6)

2.3.5 - ממשק פס של 8 ביט עם פורמט של 24 ביטים לפיקסל

במקרה הזה אחרי הפקודה 2CH שאומרת לכתוב לזיכרון מוציאים 3 בייטים של הצבעים. כל בייט הוא 8 ביטים . מבנה הצבעים R8 G8 B8 . סה"כ ניתן לקבל $2^{24} = 16777216$ צבעים שונים.

טבלה 3 מסכמת את הנאמר בסעיפים האחרונים.

- Parallel 8-Bits Bus Interface (IM2,IM1,IM0="011")

Register Command	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Command
	X	X	X	X	X	X	X	X	X	X	0	0	1	0	1	1	0	0	2CH
3AH	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Color
05h	X	X	X	X	X	X	X	X	X	X	R4	R3	R2	R1	R0	G5	G4	G3	65K-Color (1-pixel/ 2-bytes)
	X	X	X	X	X	X	X	X	X	X	G2	G1	G0	B4	B3	B2	B1	B0	
06h	X	X	X	X	X	X	X	X	X	X	R5	R4	R3	R2	R1	R0	X	X	262K-Color (1-pixel/ 3bytes)
	X	X	X	X	X	X	X	X	X	X	G5	G4	G3	G2	G1	G0	X	X	
	X	X	X	X	X	X	X	X	X	X	B5	B4	B3	B2	B1	B0	X	X	
07h	X	X	X	X	X	X	X	X	X	X	R7	R6	R5	R4	R3	R2	R1	R0	16M-Color (1-pixel/ 3bytes)
	X	X	X	X	X	X	X	X	X	X	G7	G6	G5	G4	G3	G2	G1	G0	
	X	X	X	X	X	X	X	X	X	X	B7	B6	B5	B4	B3	B2	B1	B0	

טבלה 3 : טבלה המסכמת ממשק פס של 8 ביט עם 3 אפשרויות תבנית הצבעים.

4.5 ממשק פס של 9 ביט עם 3 אפשרויות תבנית הצבעים

בטבלה 4 מתואר ממשק של 9 ביט עם 3 אפשרויות תבנית הצבעים.

- Parallel 9-Bits Bus Interface (IM2,IM1,IM0="001")

Register Command	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Register
	X	X	X	X	X	X	X	X	X	X	0	0	1	0	1	1	0	0	2CH
3AH	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Color
05h	X	X	X	X	X	X	X	X	X	X	R4	R3	R2	R1	R0	G5	G4	G3	65K-Color (1-pixel/ 2-bytes)
	X	X	X	X	X	X	X	X	X	X	G2	G1	G0	B4	B3	B2	B1	B0	
06h	X	X	X	X	X	X	X	X	X	R5	R4	R3	R2	R1	R0	G5	G4	G3	262K-Color (1-pixel/ 2bytes)
	X	X	X	X	X	X	X	X	X	G2	G1	G0	B5	B4	B3	B2	B1	B0	

טבלה 4 : ממשק פס של 9 ביט עם תבנית 3 אפשרויות הצבעים.

5.5 : ממשק פס של 16 ביט עם 3 אפשרויות תבנית הצבעים

בטבלה 5 מתוארת תבנית הצבעים עם ממשק פס של 16 ביט עם 3 אפשרויות תבנית הצבעים.

- Parallel 16-Bits Bus Interface (IM2,IM1,IM0="010")

Register Command	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Command
	X	X	X	X	X	X	X	X	X	X	0	0	1	0	1	1	0	0	2CH
3AH	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Color
05h	X	X	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	65K-Color
06h	X	X	R5	R4	R3	R2	R1	R0	X	X	G5	G4	G3	G2	G1	G0	X	X	262K-Color (2-pixels/ 3bytes)
	X	X	B5	B4	B3	B2	B1	B0	X	X	R5	R4	R3	R2	R1	R0	X	X	
	X	X	G5	G4	G3	G2	G1	G0	X	X	B5	B4	B3	B2	B1	B0	X	X	
07h	X	X	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	16M-Color (2-pixels/ 3bytes)
	X	X	B7	B6	B5	B4	B3	B2	B1	B0	R7	R6	R5	R4	R3	R2	R1	R0	
	X	X	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0	

טבלה 5 : ממשק פס של 16 ביט עם 3 אפשרויות תבנית הצבעים.

6.5 : ממשק פס של 18 ביט

בטבלה מספר 6 מתואר ממשק פס של 18 ביטים. כאן יש אפשרות לתבנית צבעים אחת בלבד של 18 ביטים (6 לכל צבע – סה"כ 262K צבעים שונים).

- Parallel 18-Bits Bus Interface (IM2,IM1,IM0="000"and DB_EN='0')

Register Command	DB17	DB16	DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Register
	X	X	X	X	X	X	X	X	X	X	0	0	1	0	1	1	0	0	2CH
3AH	DB17	DB16	DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Color
06h	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0	262K-Color

טבלה 6 : ממשק פס של 18 ביטים עם תבנית צבעים של 18 ביטים.

7.5 : ממשק פס של 24 ביטים

בטבלה 7 מתואר ממשק פס של 24 ביטים. כאן יש רק תבנית צבע של 24 ביטים.(16M צבעים).

- Parallel 24-Bits Bus Interface (IM2,IM1,IM0="000" and DB_EN='1')

Register Command	DB23	DB22	DB21	DB20	DB19	DB18	DB17	DB16	DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Register
	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	1	0	1	1	0	0	2Ch
3Ah	DB23	DB22	DB21	DB20	DB19	DB18	DB17	DB16	DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Color
07h	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0	16M-Color

טבלה 7 : ממשק פס של 24 ביט עם 16M אפשרויות צבע .

6. ממשק תקשורת טורי MIPI DBI TYPE-C

הבקר HX8357 D00/01 תומך ממשק העברת נתונים טורית של 3 או 4 חוטים. ממשק טורי נקרא **MIPI DBI TYPE-C** .

בחירת הממשק הטורי נקבעת בעזרת ההדקים IM2-0 (העזר בטבלה 2 בעמודים קודמים). אם נשים בהדקים אלו 101 נקבל ממשק של 3 חוטים והוא נקרא : **MIPI DBI TYPE-C option 1** .

אם נשים 111 נקבל ממשק של 4 חוטים והוא נקרא : **MIPI DBI TYPE-C option 3**.

בממשק של 3 חוטים יש את הקווים **CSX** - **C**hip **S**elect - המשמש לבחירת הרכיב , **SDA** - **S**erial input/output **DA**ta המשמש כקו הנתון הטורי (יכול להיות קלט או פלט) והחוט השלישי הוא **WRX_SCL** **S**erial **CL**ock המשמש כהדק השעון לסנכרון הנתון הטורי מהמעבד אל הבקר או מהבקר אל המעבד.

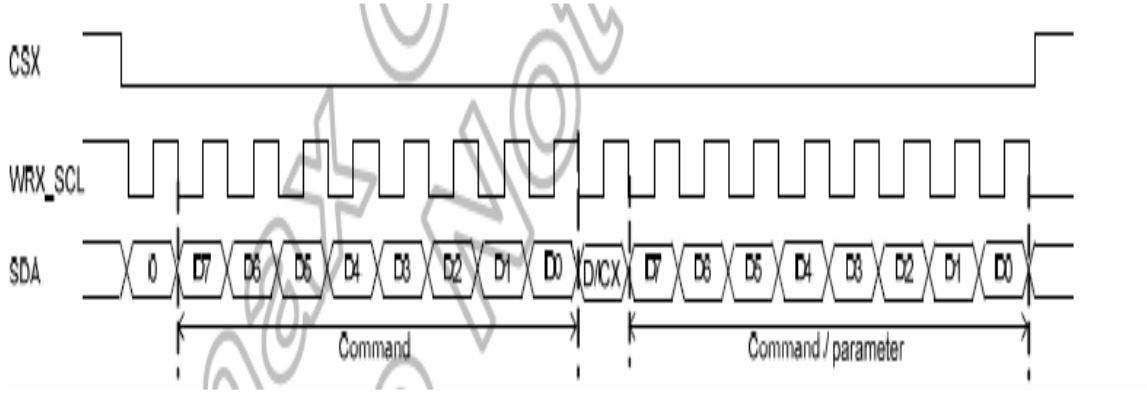
בממשק עם 4 חוטים יש בנוסף לשלושת הקווים הקודמים קו הנקרא **DCX** - **D**ata **C**ommand שמראה האם מדובר בפקודה או נתון. כאן קו הנתון הטורי נקרא **SDA/SDO** .

1.6 : כתיבה בתקשורת טורית

בתקשורת טורית עם 3 קווים כוללת חבילת התקשורת ביט בקרה D/CX ואחריו יש בייט . אם ביט הבקרה D/CX ב 0 אז הבייט הוא פקודה ואם ביט הבקרה ב 1 אז הבייט מאוחסן ברגיסטר הפקודה של ה GRAM . ביט ה MSB משודר ראשון. כאשר הקו CSX בגבוה אין לפולסי השעון או הנתון הטורי כל השפעה. ירידה מ 1 ל 0 בקו CSX מאפשרת את התקשורת הטורית ומציינת התחלת תקשורת נתונים.

1 . 1.6 - תקשורת טורית עם 3 קווים

איור 15 מתאר צורות גלים של תקשורת טורית עם 3 קווים.

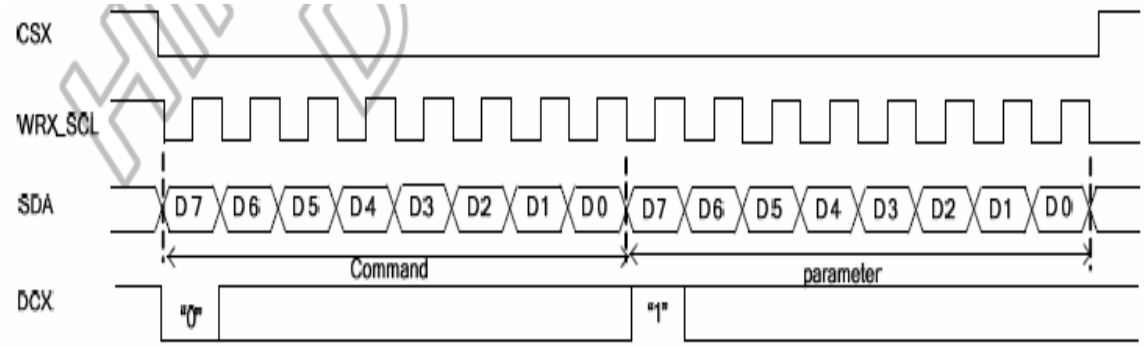


איור 15 : תקשורת טורית עם 3 קווים.

ניתן לראות שהתקשורת הטורית מתחילה רק כאשר הקו CSX יורד ל 0 .
הביט הראשון בקו SDA ב 0 ומציין שמדובר בפקודה. לאחר מכן יש את 8 הביטים של הפקודה.
הנתון הטורי בקו SDA ננעל לתוך בקר התצוגה בכל עלייה של פולס השעון בקו WRX_SCL .
לאחר כתיבת הפקודה מופיע ביט המציין אם מדובר בפקודה (בביט יש 0) או בנתון (בביט יש 1) ומיד אחרי הביט יש את הפקודה או הנתון.

2 . 1.6 - תקשורת טורית עם 4 קווים

איור 16 מתאר תקשורת טורית עם 4 קווים.



איור 16 : תקשורת טורית עם 4 קווים

קו ה DCX בהתחלה ב 0 ומציין שמדובר בפקודה.

גם כאן התקשורת מתחילה בירידת הקו CSX מ 1 ל 0 .

הנתון הטורי בקו SDA ננעל לתוך בקר התצוגה עם כל עליית שעון בקו WRX_SCL .

חסכנו את הביט הראשון בכל שידור בקו SDA המציין האם מדובר בפקודה או נתון.

ניתן לראות שאחרי שליחת הביט D7 הקו DCX יכול לשנות את מצבו . אחרי שידור ביט D0 יש לשים בקו DCX שוב את הערך 0 (לפקודה) או 1 (עבור נתון) לפני שמעלים את הדק השעון ל1 לנעילת הביט D7 של הפקודה או הנתון הבאים.

2.6 קריאה בתקשורת טורית בממשק טורי היקפי SPI

בפעולת קריאה בממשק טורי היקפי המעבד שהוא המסטר שולח לבקר התצוגה (שמשמש כעבד) פקודה ואז הבייט הבא נשלח מהבקר (העבד) אל המעבד (מסטר). באופן הקריאה יש 2 פקודות קריאה . אחת של 8 ביט והשנייה של יותר מ 8 ביט. לקריאה מה GRAM יש רק פקודה אחת.

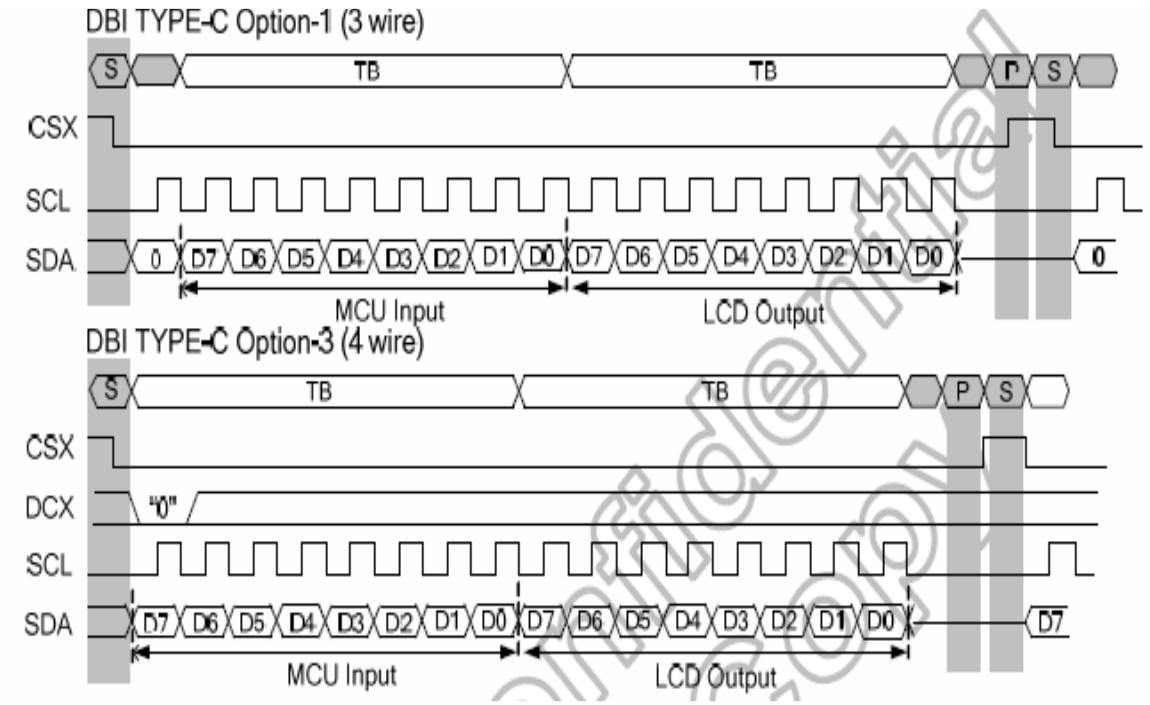
איור 17 מתאר את צורת הגלים בפקודת קריאה של 8 ביטים . 4 צורות הגל הראשונות מתארות תקשורת עם 3 קווים. 5 צורות הגל הבאות מתארות קריאה עבור תקשורת עם 4 קווים.

בארבע צורות הגל העליונות רואים את אופן הקריאה עבור 3 קווים בין המעבד לבקר התצוגה.

שוב, התקשורת מתחילה כאשר קו CSX יורד ל 0 .

המעבד שולח לבקר התצוגה פקודה . הביט הראשון בקו ה SDA הוא 0 ולאחריה פקודה בת 8 ביטים המבקשת לקרא פקודה או נתון הנמצאים בבקר התצוגה.

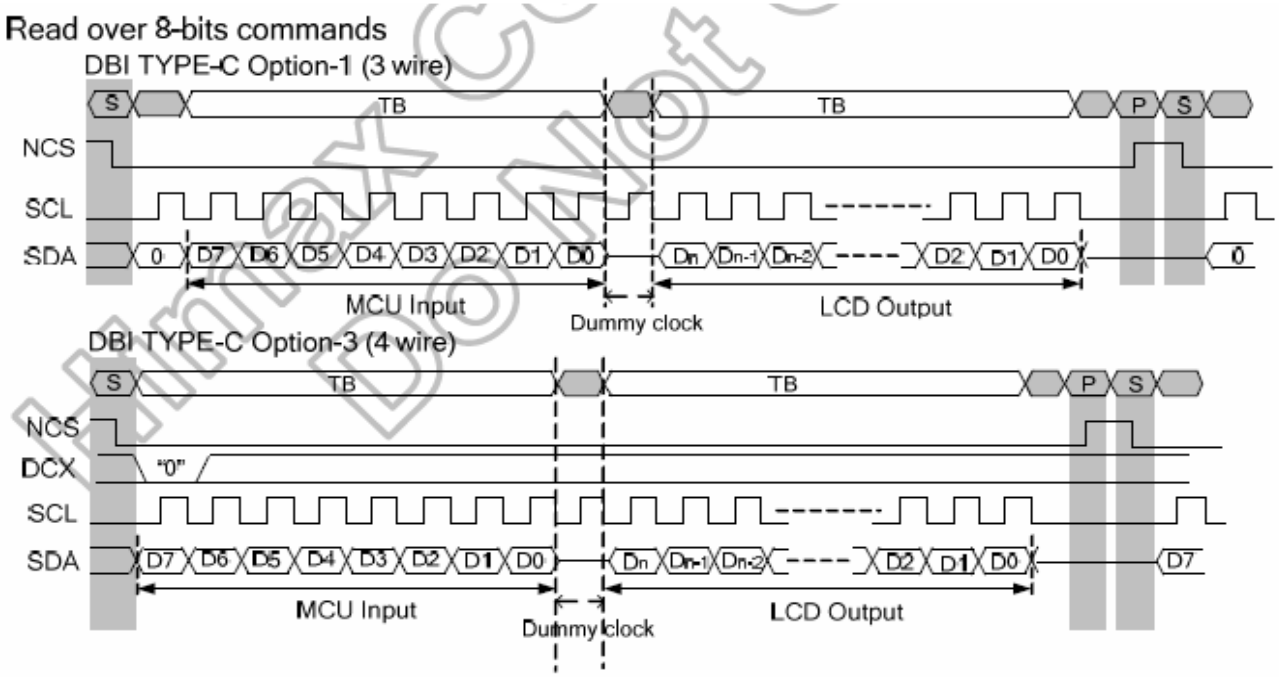
אחרי שידור הביט האחרון D0 של הפקודה יש להפוך את כיוון הקו SDA ועכשיו הוא הופך להיות קלט. בקר התצוגה שולח נתון אל המעבד. את פולסי השעון ממשיך ליצור המעבד ובכל פולס נכנס ביט אל המעבד. הביט נכנסים אל המעבד מהגבוה אל הנמוך.



איור 17 : קריאה של 8 ביטים בתקשורת טורית עם 3 קווים ועם 4 קווים.

בצורות הגל התחתונות נראית התקשורת עם 4 קווים. בקו DCX שמים בהתחלה 0 שמראה ששולחים פקודה ולאחר מכן התהליך דומה לזה של 3 קווים.

באיור 18 רואים קריאה של יותר מ 8 ביט גם עבור ממשק של 3 קווים ומתחתיו ממשק של 4 קווים.

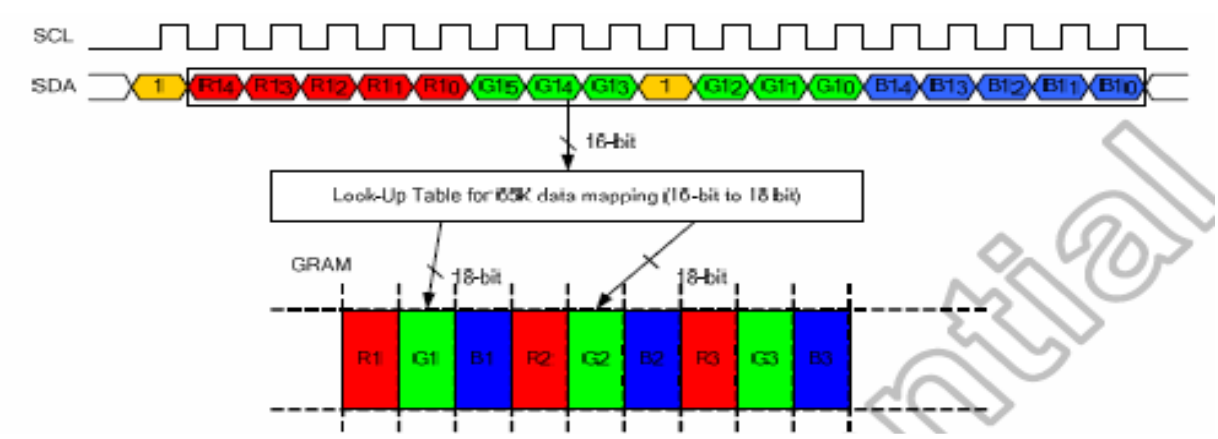


איור 18 : קריאה של יותר מ 8 ביטים בממשק טורי

ניתן לקבוע כמה ביטים נרצה לקרא ברצף בתקשורת טורית. לשם כך נשלח פקודת קריאה מהמעבד אל בקר התצוגה. בסיום הפקודה (לאחר שליחת הביט D0) יש לשלוח פולס נוסף של שעון שאין לו תפקיד. מיד לאחר מכן נהפוך את כיוון הקו SDA ונקלוט ברצף את כמות הביטים שביקשנו לקרא.

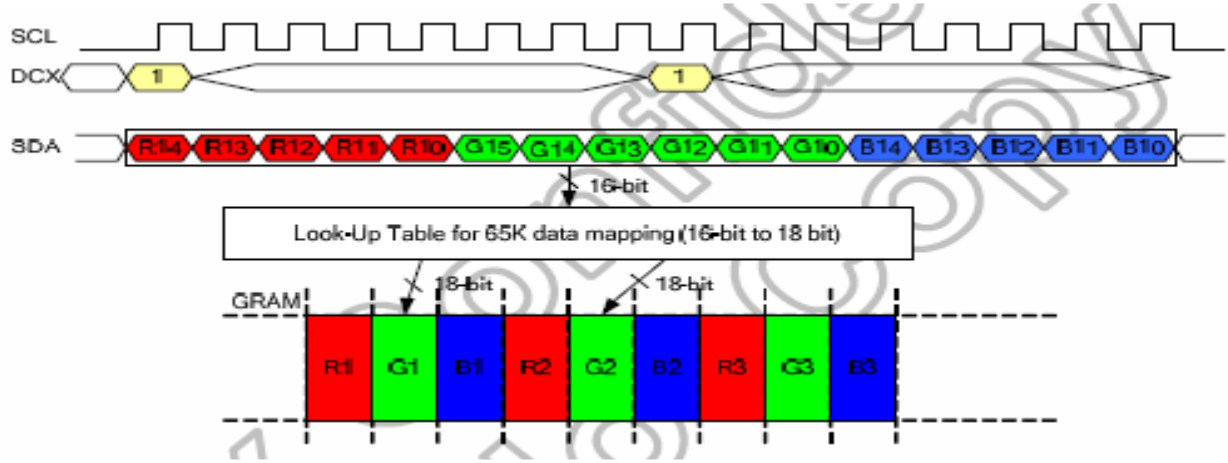
3.6 : כתיבת נתון לתצוגה בתקשורת טורית

איור 19 א ו ב מראים 2 דרכים לכתוב נתוני תצוגה בממשק טורי . איור א עבור ממשק של 3 קווים ואיור ב עבור ממשק טורי של 4 קווים.



איור 19א : ממשק טורי עבור 3 קווים עבור קוד צבע R5 G6 B5

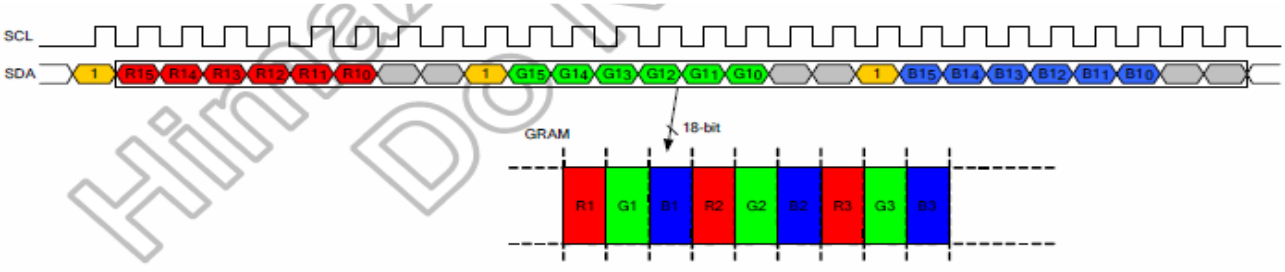
ב 2 האיורים מתארים רק את הקו SDA ו SCL. כמובן שיש להוסיף את הקו CSX בהתאמה.
בהתחלה שולחים ביט של 1 המציין שמדובר בנתון. לאחר מכן שולחים את 5 הביטים של הצבע האדום ו3 הביטים הגבוהים של הירוק. שוב שמים 1 לציין שמדובר בנתון ושולחים את 3 הביטים הנמוכים של הירוק ועוד 5 הביטים של הכחול.
באיור ב' מראים את צורות הגל עבור ממשק של 4 קווים.



איור 19 ב : ממשק טורי עבור 4 קווים עבור קוד צבע R5 G6 B5

באיור 19ב הקו DCX חוסך את ביט ה 1 בתחילת כל בייט שידור . מבנה הצבע כמו עם 3 קווים.

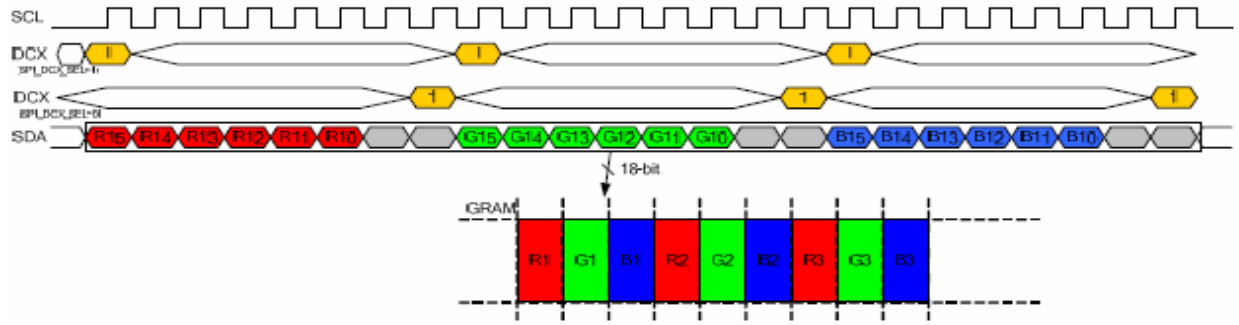
באיור 20א נראה ממשק טורי ל 3 קווים עבור קוד צבע של 18 ביט R6 G6 R6. באיור 20ב נראה את אותו קוד צבע עבור ממשק של 4 קווים.



איור 20א : ממשק טורי של 3 קווים עם קוד צבע R6 G6 B6 .

באיור רואים שלמעשה שולחים 3 בייטים. בייט ראשון שבו ביט ראשון הוא 1 מסומן בצבע צהוב המציין נתון. מיד אחריו שולחים 6 ביטים של הצבע האדום (צבועים בצבע אדום) . 2 פולסי השעון הבא הכרחיים אבל לא מכניסים את הנתון ב SDA לשום מקום (בצבע אפור). הביט הבא הוא 1 בצבע צהוב כדי לציין שמדובר בנתון. 6 הביטים הבאים (בצבע ירוק) הם הביטים של הצבע הירוק. שוב 2 ביטים אפורים ללא משמעות ולאחריהם ביט בצבע צהוב המציין נתון. 6 ביטים (בצבע כחול) של הצבע הכחול ו 2 ביטים נוספים ללא משמעות.

באיור 20ב מתואר ממשק טורי של 4 קווים עם אותו קוד צבע של R6 G6 B6 .



איור 20ב : ממשק טורי של 4 קווים עם קוד צבע R6 G6 B6 .

באיור 20ב רואים שהקו DCX חוסך בכל בייט את הצורך לציין בעזרת ביט של 1 (צבוע בצהוב באיור 20א).

טבלה 8 מסכמת את ההסבר שנתנו ומתארת כיצד נכתבים הביטים ל GRAM בתקשורת הטורית.

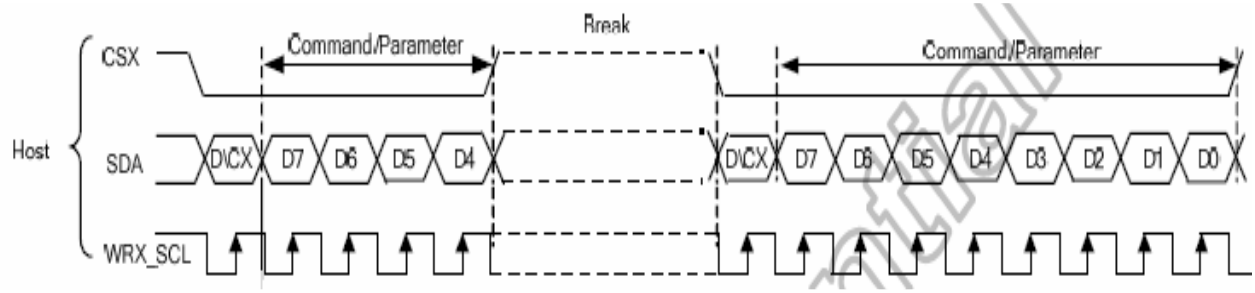
Register Command	D7	D6	D5	D4	D3	D2	D1	D0	Command
3Ah	0	0	1	0	1	1	0	0	2CH
05h	R4	R3	R2	R1	R0	G5	G4	G3	65K-Color (1-pixel/ 2-transfer)
	G2	G1	G0	B4	B3	B2	B1	B0	
06h	R5	R4	R3	R2	R1	R0	x	x	262K-Color (1-pixel/ 3-transfer)
	G5	G4	G3	G2	G1	G0	x	x	
	B5	B4	B3	B2	B1	B0	x	x	

טבלה 8 : תקשורת טורית בממשק DBI TYPE-C המכמת את הכתיבה ל GRAM.

מתוך הטבלה ניתן להבין שלא ניתן לשלוח קוד צבע של 16M בתקשורת טורית.

4.6 שבירה והפסקה של העברת נתונים

ניתן לעצור או להפסיק את שליחת הנתונים באמצע שידור. הדבר מתבצע על ידי העלאת קו CSX ל 1 . אם ההפסקה מתבצעת באמצע שליחת פקודה או נתון אז המודול מתעלם מהפקודה ובהפעלה הבאה יש לשלוח את הפקודה מחדש (כשנוריד את CSX פעם נוספת ל 0). הדבר מתואר באיור 21 . אם העצירה/הפסקה היא בסוף בייט אז ניתן לשדר בייט חדש בסיום ההפסקה/ עצירה.

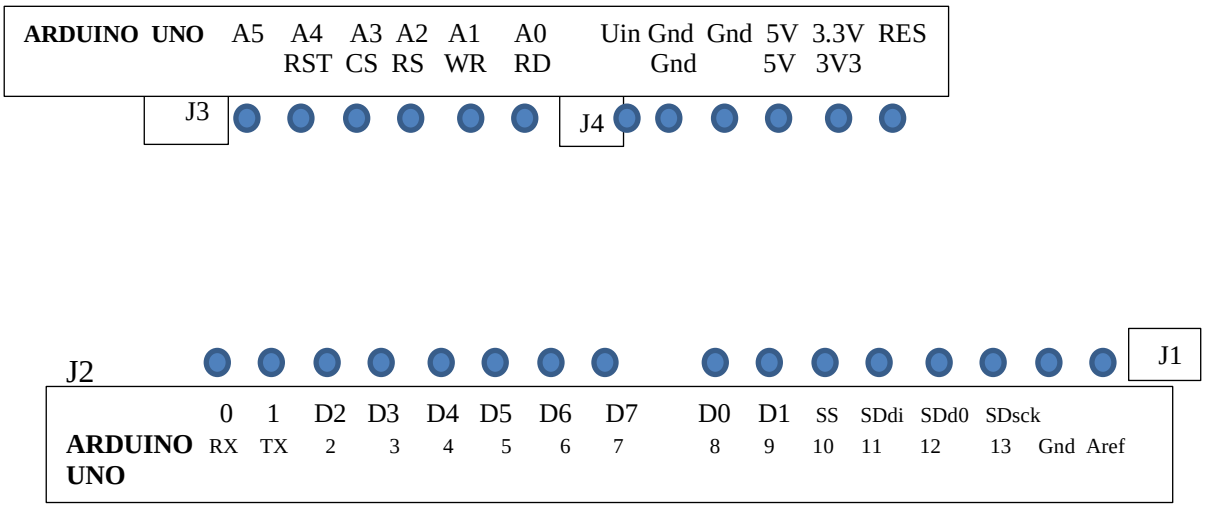


איור 21 : שליחה חוזרת של פקודה לאחר עצירה

אם אחד או יותר מהפרמטרים הנשלחים בפקודה נשלח ומתבצעת עצירה בשליחת פרמטר לפני שהמסטר שלח את הפרמטר האחרון והמסטר שולח פקודה חדשה במקום לשלוח שוב את הפרמטר שבו עצר אז הפרמטרים האחרונים שנשלחו מאוחסנים והפרמטר שבו הייתה עצירה לא מתקבל.

5. תצוגת TFT המונחת על כרטיס ארדואינו אונו

באיור 22 נראים הדקי כרטיס התצוגה כשהוא מונח על כרטיס הארדואינו.



איור 22 - תאור הדקי כרטיס התצוגה שמונח על כרטיס הארדואינו אונו.