

FPGA - (Field Programmable Gate Array) (מערך שערים נתכנת שדה)

מה זה FPGA?

רכיב זה הוא רכיב אלקטרוני המאפשר תכנות מחדש של חומרה, הוא מורכב ממערכי שערים לוגיים שניתן לתכנת מחדש ולחבר אותם יחד בצורות שונות כדי ליצור רכיבי חומרה שממלאים מטרות שונות, זהו סוג של מעגל משולב שאפשר לתכנת אותו לאחר ייצורו כדי לבצע מגוון רחב של פעולות בניגוד למעבדים רגילים שיש להם ארכיטקטורה קבועה ופונקציות מוגדרות, את הFPGA ניתן לתכנת מחדש כדי לבצע משימות ספציפיות.

איך FPGA עובד?

ה-FPGA מורכב מבלוקים לוגיים, שערים לוגיים, ומטריצות של קווי חיבורים שניתן להתאים אותם לפי צורך. תכנות ה-FPGA נעשה באמצעות שפות תיאור חומרה כגון VHDL או Verilog. המתכנת כותב קוד המתאר את הרכיב הלוגי אותו הוא רוצה ליצור. קוד זה עובר הידור (compilation) לקובץ שנקרא נתיב (bitstream), אשר נטען לתוך ה-FPGA ומתכנת אותו. הנתיב מגדיר כיצד כל שער לוגי ורכיב ב-FPGA יחובר לשערים ורכיבים אחרים.

מה השימושים של FPGA

ייצור מעגלים ייעודיים עבור משימות מסוימות, כגון:

- עיבוד אותות ותקשורת - מערכות טלקום, רשתות אלחוטיות ועוד. היכולת לשנות את התוכנה מאפשרת שדרוג והתאמה לתקנים ופרוטוקולים חדשים.
- מערכות בקרה ומדידה - בקרת מנועים, בקרת רובוטים, מכשור רפואי.
- עיבוד תמונה ווידאו - מצלמות, סורקים רפואיים ועוד.
- חישוב מקבילי בביצועים גבוהים - עיבוד נתוני Big Data, כריית מטבעות דיגיטליות ועוד.
- אבטיפוס מהיר של מוצרים חדשים.

דוגמאות למוצרים מבוססי FPGA:

- כרטיסי רשת וניתוב תקשורת של חברות שונות.
- מערכות לומדה אוטונומיות.
- ציוד בקרה ואוטומציה של מפעלים.
- מכשירי אולטרה-סאונד רפואיים.
- ציוד לכריית ביטקוין.

FPGA של יתרונות

- **גמישות:** FPGA הם רכיבים גמישים ביותר וניתן לתכנת אותם מחדש ולהגדיר אותם לביצוע משימות או פונקציות שונות. הם בעצם לוחות ריקים שניתן להתאים ליישומים ספציפיים באמצעות שפות תיאור חומרה כמו VHDL או Verilog.
- **טיים טו מרקט:** FPGA מספקים זמן הגעה קצר יותר לשוק מכיוון שניתן לתכנת אותם במהירות וליצור אב טיפוס. זה יתרון לפיתוח ובדיקות מהירים.
- **עלות:** FPGA הם בדרך כלל חסכוניים עבור יצירת אב טיפוס והרצות ייצור בנפח נמוך עד בינוני מכיוון שהם מבטלים את העלויות הגבוהות מראש הקשורות לתכנון וייצור מעגלים כמו ASIC.
- **עלויות NRE (הנדסה לא חוזרת):** ל-FPGA יש עלויות NRE מינימליות, מה שהופך אותם למתאימים לפרויקטים בתקציב נמוך או מחקר.

חסרונות של FPGA

ביצועים: FPGA מציעים ביצועים טובים עבור מגוון רחב של יישומים, אך הם עשויים שלא להתאים לביצועים הגולמיים של מעגלים כמו ASIC מעוצבים בהתאמה אישית עבור משימות מיוחדות במיוחד.

יעילות חשמל: FPGA הם בדרך כלל פחות חסכוניים בצריכת החשמל מכיוון שהם מכילים אלמנטים לוגיים הניתנים לתכנות שאינם מותאמים למשימה ספציפית.

איך מפעילים FPGA

מחברים את הרכיב למקור כוח (לדוגמא מחשב).
פותחים תוכנת תכנון מותאמת (לדוגמא QUARTUS).
מתכננים את המעגל הנדרש. (לדוגמא: בעזרת קוד VHDL).
מגדירים את המיקומים בשבב (לכניסות וליציאות).
מצב מבצעים הרצה (קומפילציה).
הורדת הקובץ המתקבל אל תוך השבב.

CPU - (Central Processing Unit) (יחידת העיבוד המרכזית)

מה זה CPU?

יחידות מעבד מרכזי הידועות בכינוי CPU הן רכיבי חומרה חיוניים במחשבים ובמכשירים טכנולוגיים רבים, תפקידן המרכזי הוא לבצע הוראות ולבצע חישובים מתמטיים ולוגיים הנדרשים להפעלת תוכנות ומערכות הפעלה. יחידות אלו מתפקדות כ'מוח' של המערכת, כאשר הן מקבלות קלט, מעבדות אותו, ומייצרות פלט בהתאם.

מבנה ה-CPU

המבנה הבסיסי של CPU כולל ליבה אחת או יותר. כל ליבה היא מעין יחידת עיבוד עצמאית שמסוגלת לבצע חישובים מופרדים, מה שמאפשר למעבדים הרב-ליבתיים לבצע מספר משימות במקביל ביעילות רבה יותר. ככל שמספר הליבות גדל, כך היכולת לבצע עיבוד מקבילי משתפרת, וזה מספק ביצועים מתקדמים יותר במשימות מורכבות. מעבדים מודרניים כוללים גם טכנולוגיות מתקדמות כגון Hyper-Threading שמאפשרות לכל ליבה להתמודד עם מספר תהליכים בו-זמנית על ידי חלוקת משאבים בצורה חכמה. בנוסף, טכנולוגיות כמו Turbo Boost מעלות את התדר של המעבד בזמן הצורך, כדי לספק ביצועים מיטביים בעת עומס. הערה: HYPERTHREADING זה טכנולוגיה שפותחה על ידי INTEL שמאפשר לליבת מעבד לפעול כאילו הוא 2 ליבות, המעבד מטפל ב-2 פעולות בו-זמנית ובכך מגדיל את היעילות והביצועים שלו.

מאפיין נוסף שחשוב לציין הוא המהירות השעונית של ה-CPU, הנמדדת ב-GHz. מהירות זו מציינת את מספר מחזורי השעונים שהמעבד יכול לבצע בשנייה. ככל שהמהירות השעונית גבוהה יותר, כך המעבד מסוגל לבצע פעולות מהר יותר, אם כי יש לקחת בחשבון גם את מספר הליבות ופקטורים אחרים המשפיעים על הביצועים הכוללים. מערכת זיכרון המטמון (Cache) היא רכיב נוסף שאחראי על שיפור ביצועי ה-CPU. זיכרון המטמון נמצא קרוב מאוד לליבות ומאפשר גישה מהירה לנתונים חשובים, מה שמקטין את זמן ההמתנה ומגביר את היעילות הכוללת של המעבד.

איך CPU עובד?

- מתחת למעבד יש מלא סיכות שמשמשות לקליטה ושליחה של מידע, ה-CPU נמצא במה שידוע כלוח אם (מעין מעגל שמאפשר לכל הרכיבים במחשב / מכשיר להתחבר אחד לשני) בתוך ה-CPU ישנם כמה חלקים:
- יחידת הבקרה - מקבל פקודות מה-RAM בצורת הוראות ומפרק את ההוראות לדרישות ספציפיות בשביל מרכיבי המעבד השונים
 - היחידה האריתמטית לוגית (ALU) - אחראית על כל הפעולות המתמטיות בגוף המעבד כגון חיבור, חיסור ואפילו השוואה
 - אוגרים - תאי זיכרון קטנים ומהירים מאוד בתוך ה-CPU. משמשים לאחסון זמני של נתונים ופקודות במהלך ביצוע חישובים.
 - שעון מערכת - שולח פולסים חשמליים בקצב קבוע, מסנכרן את כל פעולות המעבד לפי הפולסים של השעון.
 - זיכרון מטמון - זיכרון קטן ומהיר מאוד, שנמצא בתוך או קרוב ל-CPU. שומר נתונים ופקודות בשימוש תדיר כדי להאיץ את הביצועים.
- הערה: ההוראות של המערכת נשמרות סט הפקודות שלו בהתאם לארכיטקטורה שלו

יתרונות של CPU לעומת FPGA

יעילות אנרגטית:

בשנים האחרונות, היעילות האנרגטית של רכיבי FPGA ושל מעבדים מסורתיים (CPUs) הפכה לפרמטר ביצועים מרכזי. גודל הבעיות והיישומים גדל באופן משמעותי מדי שנה, מה שמוביל לנפחי עיבוד נתונים עצומים. רכיבי FPGA מציעים איזון בין יכולת תכנות לבין חיסכון באנרגיה. עומסי עבודה בעלי צריכת חשמל נמוכה הם שימושים נפוצים למעבדים, במיוחד כאשר נדרשת יכולת חישוב עם נקודה צפה (floating-point). כתוצאה מכך, מערכת CPU כזו צורכת רק 225 וואט לצורך ביצוע אותם תהליכים.

מערכת הפעלה:

ה-CPU יכול להריץ מערכות הפעלה מודרניות, FPGA אינו יכול לעשות זאת.

חסרונות של CPU לעומת FPGA

• מורכבות:

מעבדים מורכבים יותר מ-FPGA בגלל השיצור שלהם כבר הגדיר את המבנה הפנימי של המערכת ואת שיטת העבודה שלה, לעומת FPGA שהמשתמש מגדיר את המבנה הפנימי בעזרת המערכים הפנימיים.

• עיבוד קריטי לזמן:

מעבדים לעיתים מתמודדים עם מגבלות ולכן אינם הבחירה הטובה ביותר למשימות קריטיות לזמן במיוחד אם המשימות חורגות מיכולותיהם.

• התאמה חומרתית:

המבנה הפנימי של CPU קבוע מראש ולא ניתן לשנותו לעומת FPGA שגמיש יותר

הבדלים נוספים

המעבד FPGA הוא מקבילי כלומר כל הפעולות שלו קורות בו זמנית לעומת זאת מעבד CPU הוא רציף והפעולות בו קורות אחד אחרי השני

איך מפעילים CPU?

מחברים את המעבד למקור כוח מתאים (למשל: ספק כוח 3.3V או 5V, תלוי בדגם).
הערה: נבחר מתח של 3.3V כסטנדרט כדי לגשר על הפער בין לוגיקת TTL של 5V ישנה לבין הצורך בצריכת חשמל נמוכה יותר. זה מאפשר תאימות טובה עם מערכות 5V קיימות תוך הצעת יתרונות של פעולה במתח נמוך יותר.

מוסיפים רכיבי עזר חיוניים כמו קריסטל (Clock), נגדי Pull-Up לפין Reset, ולפעמים קבלים.
מכנים זיכרון תוכנה – מעבדים מסוימים קוראים את הקוד מתקן חיצוני כמו Flash או ROM.
מחברים התקני קלט/פלט (לדוגמה: לחצנים, לדים, תצוגה).
כותבים קוד תואם (ב-C או Assembly) שמגדיר את פעולת המעבד.
מריצים את הקוד בעזרת תוכנת פיתוח (כמו MPLAB, STM32CubeIDE או Keil).
מורידים את הקוד למעבד בעזרת מתאם תכנות (כגון ST-Link או USBasp).
מאפסים את המעבד (על ידי לחצן או פין RESET) והוא מתחיל להריץ את הקוד.
הערה: נגד PULL-UP הוא בעצם מתג שתמיד מספק 1 לוגי וכאשר נלחץ הוא יורד ל-0 לוגי, כדי להבטיח שהוא יהיה במצב לוגי מוגדר HIGH או LOW.

ארכיטקטורת מעבדים

בתחום מדעי המחשב ישנו מושג הנקרא 'סט פקודות' (כמו המילון של המעבד ומכיל את כל הפקודות שהמעבד יכול לבצע ואיך המעבד יכול לבצע אותן).

ARM

ARM היא ארכיטקטורת מעבד בשימוש נפוץ עבור מכשירים ניידים כגון סמארטפונים, טאבלטים ומערכות משובצות. מעבדי ARM ידועים ביעילות האנרגיה שלהם ובצריכת החשמל העיצובית הנמוכה יותר, מה שהופך אותם לבחירה אידיאלית עבור מכשירים המופעלים על ידי סוללה.
ארכיטקטורת ARM היא ארכיטקטורה שחברות כמו AMD ו-INTEL כיום משתמשות בה.
מעבדי ARM מבצעים פקודות פחות מסובכות. לכן, מעבדים בארכיטקטורה זו יכולים להסתפק בפחות טרנזיסטורים

סט הפקודות של ARM נקרא RISC - Reduced Instruction Set Computing או RISC. הוא מכיל פקודות יותר פשוטות אשר יכולות להתבצע בטיק שעון אחד.
המטרה ב-ARM היא לבצע פחות פקודות, כל אחת בצורה מהירה ויעילה.

X86

X86 היא ארכיטקטורת מעבד בשימוש נפוץ במחשבים שולחניים ומחשבים ניידים. מעבדי X86 הם בדרך כלל חזקים יותר ממעבדי ARM ומשמשים במגוון רחב של יישומי מחשוב ממחשבים אישיים ועד לשרתים.
ארכיטקטורה זו ידועה גם כ-8086
מעבדי x86 מבצעים פקודות יותר מסובכות ודורשים יותר טרנזיסטורים. בגלל שמעבדי ARM משתמשים בפחות טרנזיסטורים הם גם משתמשים בפחות חשמל.
סט הפקודות CISC - Complex Instruction Set Computer או CISC, כשמו, מכיל פקודות יותר מורכבות אשר דורשות יותר זמן (או טיקים בשעון המעבד) אך מהוות תחליף לביצוע מספר פקודות פשוטות יותר.
המטרה ב-86 היא לבצע יותר בפחות שורות קוד – כלומר, לפשט את עבודת המפתח גם אם הפקודה הבודדת היא מורכבת יותר.

VHDL - VHSIC Hardware Description Language

תהליך PROCESS - למה משמש התהליך?

מאפשר תיאור של התנהגות שקשורה לזמן או לתנאים מסוימים.
מאפשר שימוש בפקודות סדרתיות כמו if, case, loop, הקשות ליישום ב-architecture רגיל.
מתאים במיוחד לתיאור מערכות סינכרוניות (שעובדות לפי שעון) וא-סינכרוניות
הsyntax הנכון לprocess הוא

```
Label process (sensitivity list)
begin
    vhd code
end process;
```

Sensitivity list -

מתארת אילו אותות (signals) גורמים להפעלת התהליך (process) מחדש — כלומר, אילו אותות "רגישים" וגורמים להרצת הקוד שבתוך ה-process בכל פעם שהם משתנים.

סוגי תהליכים:

1. תהליך אסינכרוני (combinational logic)
לא תלוי בשעון.

לדוגמה: ממיר, מקודד, דקודר, אלוגריתם לוגי.

חייב להכיל את כל האותות הקוראים או נקראים בתוכו ברשימת הרגישות – אחרת עלולות להתרחש שגיאות סימולציה.
דוגמה:

```
sync_proc : process (clk, reset)
begin
    if reset = '1' then
        Q <= '0';
    elsif rising_edge(clk) then
        Q <= D;
    end if;
end process;
```

2. תהליך סינכרוני (clocked process)

פועל עם אות clk ולעיתים גם reset.

מתאים לתיאור דלגלגים (FF) או רשמי מצבים (FSM).
דוגמה:

```
sync_proc : process (clk, reset)
begin
    if reset = '1' then
        Q <= '0';
    elsif rising_edge(clk) then
        Q <= D;
    end if;
end process;
```

הערה: שימוש ב-`rising_edge(clk)` עדיף על `if clk = '1' and clk'event` כי הוא ברור יותר מבחינה לוגית ונחשב לסטנדרט.

דגשים חשובים:

בתוך process, הפקודות מתבצעות בסדר כתיבתן, בניגוד ל-concurrent statements שמבוצעים במקביל. ניתן להכריז בתוך תהליך על משתנים (variable), בניגוד ל-signal שמוגדרים מחוץ לתהליך. עדכון של signal בתוך תהליך לא נכנס לתוקף מיידית – אלא רק בסיום התהליך. משתנה (variable) לעומת זאת, מתעדכן מיידית – שימוש בו מתאים לחישובים זמניים בתוך התהליך.

Constraints

כשמתכנתים FPGA בתוכנות כמו Xilinx Vivado, צריך להודיע לתוכנה אילו פינים פיזיים על ה-FPGA משויכים לאותות שקוד ה-HDL שלך מגדיר. במיקרו-בקרים, הפינים לרוב קבועים מראש ואי אפשר לשנות לאיזה פיין מחברים את ה-SPI, למשל. אבל ב-FPGA, אפשר למפות את האותות כמעט לכל פיין שתרצה, מה שנותן גמישות גבוהה לעיצובי חומרה מותאמים. עם זאת, אי אפשר לשנות את החיבורים הפיזיים של הפינים ללוח, כלומר אם יצרן הלוח חיבר פיין מסוים של ה-FPGA ל-HDMI, אי אפשר לשנות את זה בתוכנה – אפשר רק למפות את האות בקוד לפיין הזה. קובץ ה-Constraints של Xilinx נקרא XDC (Xilinx Design Constraints). כתיבת קובץ כזה לבד יכולה להיות קשה למתחילים, כי צריך לדעת לאיזה פינים פיזיים מחוברים רכיבים בלוח ואילו תקני מתח וחשמל הם תומכים.

קבצי ה-XDC הראשיים עבור לוחות Digilent זמינים במאגר Github בשם digilent-xdc. שם אפשר להוריד את הגרסאות המעודכנות. קובץ ה-XDC הראשי כולל את כל הפינים של ה-FPGA שמחוברים פיזית ללוח, ומסודרים לפי קבוצות רכיבים. רוב הפינים בקובץ מסומנים כ-commented (#) כדי שתוכל להפעיל רק את אלו שאתה צריך. אם בפרויקט שלך אין קובץ XDC, צריך להוסיף אותו ידנית – הקובץ מגדיר אילו פינים מחוברים לרכיבים, תדרי שעון, מתחים ועוד. בדרך כלל תשנה רק את השם של הפיין שמוגדר אחרי הפקודה get_ports כך שיתאים לשמות האותות שהגדרת בקוד ה-HDL שלך. בנוסף, תצטרך להסיר את סימן ה-# מהשורות של הפינים שאתה רוצה להשתמש בהם (כלומר, לבטל את ההערה).

CLOCK

שעון הוא אות דיגיטלי שמתהפך (מתחלף בין 0 ל-1 וחוזר על עצמו) בקצב קבוע, לדוגמה כל 10 ננו-שניות. הוא משמש כסיגנל סינכרוני שמאפשר תזמון אחיד לפעולות שונות במערכת דיגיטלית.

למה משתמשים בשעון?

- סינכרון של נתונים
עדכון ערכים במעגל רק כאשר השעון מתחלף (לרוב עליית השעון).
לדוגמה, רישום (Register) שומר ערך חדש רק ב-posedge (עליית) השעון.
- ספירה וזמנים
בניית סופרים לספירת מחזורים (cycles) של השעון, ליצירת דיליי מדויק או לוגיקה של טיימר.
- סינכרון פעולות מורכבות
הזרמת נתונים בשלבים (Pipeline), סינכרון בין רכיבים, שליטה בממשקים.

דוגמה פשוטה ל-process סינכרוני עם שעון ב-VHDL:

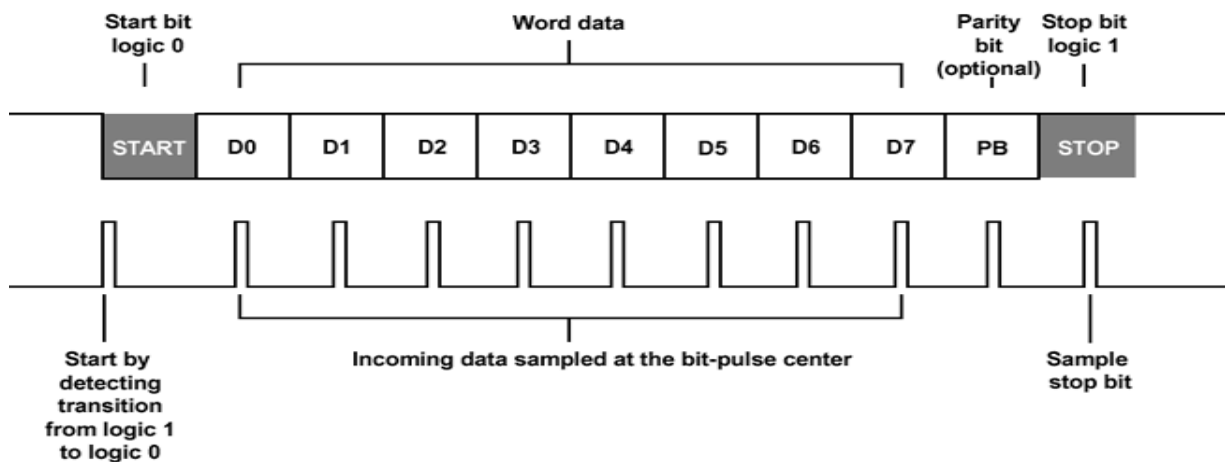
```
process(clk)
begin
    if rising_edge(clk) then
        register <= register + 1;
    end if;
end process;
```

שעון מערכת ב-art7-10

לוח ה-art7-10 מגיע עם שעון חיצוני של 125 MHz. השעון ממוקם בפין בשם PS_CLK, הוא מגיע מרכיב Oscillator חיצוני.

UART protocol - Universal Asynchronous Receiver-Transmitter

פרוטוקול UART הוא מודול חומרה לתקשורת קווית לטווח קצר בין מערכות דיגיטליות



כשהכוונה בדרך כלל להעברת מידע בין מיקרו-בקרים או שבבים אחרים, שנמצאים על גבי אותו לוח או מקסימום באותו חדר.

הסבר והגדרות כלליות

פרוטוקול UART הוא דיגיטלי, טורי, דו-סטרי וא-סינכרוני:

- דיגיטלי, כיוון שיחידת המידע הבסיסית ביותר בו היא ביט, שערכו ("1" או "0") נקבע על ידי מתח חשמלי גבוה או נמוך בלבד. אין משמעות לערכי ביניים.
- טורי, כיוון שהביטים מגיעים בזה אחר זה לפי התור, ואין מידע נוסף שעובר בו-זמנית באותו כיוון שידור. עם זאת, הוא-
- דו-סטרי – הוא כולל שני קווי מידע, אחד לכל כיוון שידור, ובכיוונים שונים כן יכול לעבור מידע בו-זמנית. התקשורת לא חייבת להיות דו-סטרי (אפשר למשל רק לשל הקו ש"יוצא" מהרכיב, שמשדר מידע ממנו החוצה, נקרא TX, ואילו הקו בכיוון ההפוך – ש"נכנס" לרכיב ומביא אליו מידע מבחוץ – נקרא RX. הסימונים האלה הם תמיד ביחס לרכיב הספציפי, ולכן TX של רכיב אחד יתחבר תמיד ל-RX של הרכיב האחר.

תקשורת UART היא כמעט תמיד בין שני רכיבים בלבד. אפשר אמנם לחבר גם שלושה או יותר לאותם קווים (לדוגמה, רכיב מנהל ששולח פקודות לשני רכיבי-משנה), אך בדרך כלל אין מנגנון מובנה בחומרה לניתוב מידע, או שהוא פרימיטיבי מאוד, ואף פעם אין מנגנון למניעה של התנגשויות בין שידורים. שילוב של שלושה רכיבים או יותר הוא נדיר וגם מחייב זהירות רבה בתכנון החשמלי. בשביל יישומים כאלה המציאו פרוטוקולים אחרים.

מימוש חשמלי

איך תקשורת UART מתבצעת בפועל?

אנחנו מתחילים מ"קו", בעצם מוליך חשמלי כלשהו שמחבר את הצד המשדר לצד הקולט, והמשדר (TX) אחראי להחזיק אותו במתח גבוה. תלוי במערכת עצמה, במיקרו-בקרים מודרניים זה יהיה בדרך כלל 5 או 3.3 וולט. אנחנו נקרא לזה פשוט HIGH. תיאורטית, המתח הזה מאפשר לנו לדעת, גם בזמן שאין שום שידור, שהקו עצמו תקין. אבל הבדיקה הזו לא מובנית בפרוטוקול או בחומרה. כיוון ש"ברירת המחדל" של הקו היא HIGH, והתקשורת היא כאמור דיגיטלית, הדרך היחידה להעביר מידע היא להוריד את הקו זמנית ל-LOW, מתח 0V (אדמה). אבל מתח LOW מציין ביט שערכו "0".

מה לעשות אם המידע שאני רוצה להעביר מתחיל דווקא ב-"1"?

כדי לפתור את הבעיה הזו המציאו מה שנקרא "ביט התחלה" (Start bit), שהוא תמיד "0", אבל הוא לא חלק מהמידע עצמו אלא חלק מהמסגרת החיצונית (frame) שבתוכה המידע נשלח. משך הזמן של הביט הזה, ושל כל הביטים האחרים בשידור, חייב להיות ידוע ומוסכם על שני הצדדים. אחרי ביט ההתחלה מגיעים הביטים של הנתונים. ברוב המערכות המודרניות משדרים בכל מסגרת שמונה ביטים = בייט אחד. הביט הראשון שמשודר הוא ה-LSB, והבאים אחריו כמובן בסדר חשיבות עולה. אחרי הביטים של הנתונים, יש אופציה להוסיף ביט זוגיות (Parity). זהו ביט שעוזר לזהות שגיאות בשידור. החומרה סופרת כמה ביטים של נתונים במסגרת הנוכחית היו "1", ומוסיפה את הערך של ביט הזוגיות. לביט זוגיות זוגי (Even) הסכום הכולל אמור להיות מספר זוגי, ולביט זוגיות אי-זוגי (Odd) הסכום אמור להיות אי-זוגי. אם הוא לא, סימן שמשהו בדרך השתבש ואי אפשר לסמוך על הביטים האלה. גם כאן, השימוש בביט הזוגיות והאופי שלו צריכים להיות מוסכמים מראש. בסיום המסגרת, המתח על הקו חייב לחזור ל-HIGH, ולשם כך יש Stop bit חובה שערכו "1". הוא יכול להימשך פרק זמן של ביט רגיל, או פי 1.5 או פי 2 – כרגיל, איך ששני הצדדים מחליטים וקובעים מראש. פרוטוקול UART לא מכיר הודעות ארוכות יותר מיחידה אחת, כפי שתוארה למעלה. כדי לשלוח נתונים נוספים צריך לעטוף כל בייט במסגרת בדיוק באותו אופן.

יש ארבעה פרמטרים של אופן השידור

- מהירות נמדד ב-Baud
- מספר ביטים של נתונים בכל מסגרת
- ביט זוגיות
- אורך ביט עצירה

RS232

תקן הוא מסמך המתאר אוסף של כללים, מפרטים ודרישות מוסכמות המשמשים להבטחת איכות, אמינות ובטיחות של מוצר, שירות או תהליך. פרוטוקול תקשורת, לעומת זאת, הוא אוסף של כללים ונהלים המאפשרים למערכות תקשורת שונות להחליף מידע ביניהן בצורה יעילה וברורה. הוא תקן תקשורת של העברת תקשורת טורית של מידע בינארי בין שני מכשירים כיום צורת התקשורת הזאת נפוצה ומשמשת לחיבור של אין ספור מכשירים אחרים המחבר הסטנדרטי הנפוץ של תקשורת RS232 נקרא DB9 והוא מורכב ממחבר בעל 9 פינים, חלק ממחשבי ה-PC שנמכרים היום מכילים בתוכם מחבר DB9 נקבה.

תקן תקשורת זה מאפשר תקשורת דו כיוונית מלאה, כלומר ישנם קווים נפרדים לאותות נכנסים ולאותות יוצאים – מה שמאפשר זרימת נתונים בו זמנית לשני הכיוונים.

כמו שצוין קודם לכן ישנם לא מעט מכשירים המתקשרים בתקשורת זו, אך לתקשורת זו יש כמה חסרונות:

- תקשורת זו מוגבלת לרוב למקסימום 15 מטרים.
- לא ניתן לפצל את התקשורת, כלומר התקשורת נעשית רק בין שני מכשירים בלבד.

כאשר מוצר בחיבור RS232 מחובר למחשב, צריכה להיות מותקנת במחשב תוכנה שיודעת "להאזין" לחיבור ה RS232 ולקבל ממנו נתונים (נקרא במחשב יציאת COM).

כעת, אם נרצה לחבר מספר חיישנים למחשב ואין לנו מספיק כניסות כנראה שנצטרך להשתמש בסוג מסויים של ממיר תקשורת. אם נרצה לחבר את המוצר במקום מרוחק מהמחשב, נצטרך להשתמש באיזשהו ממיר תקשורת לרשת.

ישנם שלוש אופציות לעבוד עם מתאם התקשורת:

- IP ישיר - בצורת עבודה של IP ישיר ישנה תוכנת PC המותקנת על אחד המחשבים שברשת ויודעת לשלוח נתונים ולקבל נתונים מכתובת IP, כך הממיר יודע לקבל נתונים מהרשת ולהמיר אותם ל-RS232 ולהפך.

- יצירת COM port וירטואלי - ישנם הרבה תוכנות שיודעות "להאזין" לחיבור ה-RS232 שבמחשב, היא בעצם מאזינה ושולחת נתונים לחיבור ה-COM שבמחשב, ניתן לראות למשל במנהל ההתקנים תחת חיבורי COM את כל חיבורי ה-COM הפיזיים שיש למחשב, בעזרת כמה הגדרות פשוטות בתוכנת ה Tibbo Connection wizard ניתן ליצור COM וירטואלי אשר יוצר COM נוסף במנהל ההתקנים, כך המחשב "חושב" שהמוצר שמחובר לממיר שברשת מחובר ישירות למחשב.

- קישור טורי וירטואלי - הקישור נעשה בעצם בין 2 ממירי התקשורת ולא בין ממיר למחשב, כך אפשר להאריך את תקשורת ה-RS232 ע"י שימוש ברשת המחשבים, בעזרת התוכנה DS Manager ניתן לזהות ברשת את כל הממירים המחוברים ולבצע קישוריות בין 2 מהם כך שכל נתון שאחד שולח יגיע לממיר השני.

RS422

תקן זה דומה לRS232 אך יש לו מספר הבדלים חשובים:

- אורך הכבל המקסימלי הוא 1200 מטר
 - ניתן לחבר מכשירים מרובים אך רק אחד מהם יפעל כמסדר והשאר מקליטים.
- שיטת תקשורת זו משתמשת בארבעה חוטי חשמל בתוך הכבל, היא אינה נפוצה במיוחד כיוון שלרוב משתמשים בRS232 או RS485.
- עם מספר הבדלים RS485 ול RS323 זהו חיבור הדומה במעט ל Recommended Standard 422 חשובים:
- אורך הכבל המקסימלי הוא 1,200 מטר וניתן לחבר מכשירים מרובים אך רק אחד מהם יפעל כמסדר והשאר יהיו מקליטים.
- מאפיינים ותכונות:
- שיטת תקשורת זו משתמשת בארבעה גידים (חוטי חשמל) בתוך הכבל, היא אינה נפוצה במיוחד כיוון שלרוב RS232 או RS485 משתמשים בתקשורת.

למה RS232 נפוץ יותר?

- **RS-232 קל מאוד למימוש** — הוא דורש שני חוטים בלבד (TX, RX, ו-GND). לעומת זאת, RS-422 משתמש בקווים דיפרנציאליים (לפחות 4 חוטים), מה שמסבך את החיווט.
- **תאימות לאחור** - RS-232 היה הסטנדרט הנפוץ ביותר למחשבים, מדפסות, מודמים וציוד תעשייתי במשך עשרות שנים.
- המון ציוד שתוכנן בעבר עובד רק עם RS-232 — ולכן נשמרת תאימות לאחור.
- **שימושים נפוצים** - RS-232 מתאים מאוד לתקשורת נקודה-לנקודה .
- ביישומים שבהם צריך רק חיבור קצר ופשוט, אין יתרון ממשי ל-RS-422.
- **זמינות חומרה** - ברוב המיקרו-בקרים, מחשבים ישנים, וציוד מדידה – יש ממשק RS-232 כבר מובנה. לעומת זאת, RS-422 דורש רכיבים חיצוניים (טרנסיבררים דיפרנציאליים), שלא תמיד זמינים מובנים.

קו דיפרנציאלי

קו תקשורת המעביר אותות חשמליים באמצעות זוג חוטים המעבירים את האות ואת השלילי שלו, במקום באמצעות חוט אחד. היתרון העיקרי של שיטה זו הוא הפחתת רעשים ושיפור איכות האות, במיוחד במרחקים ארוכים או בסביבות רועשות.

יתרונות תקשורת דיפרנציאלית:

עמידות גבוהה לרעשים:
הפרעות חיצוניות, כגון רעשים אלקטרומגנטיים, משפיעות בדרך כלל באופן דומה על שני המוליכים. כאשר המקלט מבצע חיסור בין האותות, ההפרעות מתבטלות, מה שמאפשר תקשורת אמינה יותר בתנאי סביבה קשים.

מרחקים גדולים יותר:

בגלל עמידותה הגבוהה בפני רעשים, תקשורת דיפרנציאלית מאפשרת העברת נתונים למרחקים גדולים יותר מאשר שיטות תקשורת חד-קוטביות.

מהירות גבוהה יותר:

לעיתים קרובות, תקשורת דיפרנציאלית מאפשרת שידור נתונים במהירויות גבוהות יותר.

איך תקשורת דיפרנציאלית מפחיתה רעשים?

- שידור של אות דיפרנציאלי
במקום לשדר אות על קו יחיד ביחס לאדמה (כמו בתקשורת רגילה/חד-קוטבית), משדרים שני אותות זהים בעוצמה אך הפוכים בפאזה:
לדוגמה: אם קו A משדר $1V+$, קו B ישדר $1V-$.
הפער ביניהם ($A - B$) הוא האות הלוגי – כלומר, במצב הזה: $2V+$.
- חדירה של רעש משותף – Common-mode noise
כשהשניים עוברים באותו מסלול פיזי (כבלים צמודים או זוג שזור), כל שדה מגנטי חיצוני משפיע באופן כמעט זהה על שני הקווים:
נניח שהפרעה של $0.5V+$ נכנסת לשני הקווים.
האותות יהפכו להיות:
 $A = +1.5V$
 $B = -0.5V$
ההפרש $A - B = +2V$, שזה בדיוק כמו קודם – האות המקורי נשמר!
- כיצד הביטול מתבצע
הצד הקולט (Receiver) מבצע חיסור של האותות:
 $V_{OUT} = V+ - V-$
וכך כל רעש שהשפיע באותו אופן על שני הקווים – נחתך החוצה:
זה עובד גם לרעשים אלקטרומגנטיים וגם לקפיצות מתח רגעיות.

משפחות לוגיות

TTL

טרנזיסטור-טרנזיסטור לוגי (TTL) היא טכנולוגיית מעגלים דיגיטליים המשתמשת בטרנזיסטורי מגע דו-קוטביים כדי ליצור שערים לוגיים.
היא משתמשת בטרנזיסטורים לבצוע פעולות לוגיות והגברה, ומאופיינת במהירות מיתוג גבוהה יחסית.

שימוש בטרנזיסטורים:

TTL משתמשת בטרנזיסטורים כדי לבצע את פעולות הלוגיקה (כמו AND, OR, NAND, NOR) וגם כדי להגביר את האותות.

שערים לוגיים:

שער NAND הוא השער הבסיסי ביותר ב-TTL, וממנו ניתן לבנות שערים לוגיים מורכבים יותר.

מתח אספקה:

TTL פועלת בדרך כלל עם מתח אספקה של $5V$.

מהירות מיתוג:

TTL מאופיינת במהירות מיתוג טובה, מה שהופך אותה מתאימה ליישומים הדורשים ביצועים מהירים.

תת-משפחות:

קיימות תת-משפחות שונות של TTL, כל אחת עם מאפיינים שונים, כגון צריכת חשמל נמוכה או מהירות גבוהה.

שימושים:

TTL שימשה במגוון רחב של יישומים, כולל מחשבים, ציוד בדיקה, סנטיסייזרים ועוד.

השוואה עם RTL

ב-RTL, הלוגיקה מבוצעת בעיקר באמצעות נגדים. TTL מציעה ביצועים טובים יותר מ-RTL.

יתרונות TTL:

מהירות מיתוג טובה, יכולת הגברת אותות, יציבות יחסית.

חסרונות TTL:

צריכת הספק יחסית גבוהה, רגישות יחסית למתח אספקה.

LVTTTL low voltage transistor transistor logic

תקן לרמות מתח בלוגיקה דיגיטלית שמתבסס על הטכנולוגיה של TTL אבל עובד במתחים נמוכים יותר, לרוב 3.3V במקום 5V. הוא נפוץ מאוד במערכות דיגיטליות מודרניות במיוחד בלוחות FPGA, מיקרו-בקרים, זיכרונות, וכו'.

מאפייני LVTTTL

רמות לוגיות

0 לוגי - עד כ-0.8 וולט

1 לוגי לפחות 2.4 וולט לעיתים 2.4 וולט ומעלה

תאימות

תואם ל-TTL רגיל במובן שרמות הלוגיקה נשמרות אבל לא תואם מבחינת המתח המוזן כי TTL עבד על 5V

מתי משתמשים ב-LVTTTL?

- כשיש צורך בתקשורת במהירות גבוהה יותר ועם פחות חימום וצריכת חשמל.
- כשעובדים עם רכיבים מודרניים כמו:
- לוחות פיתוח (כמו Xilinx Arty, Intel/Altera)
- מיקרו-בקרים
- זיכרונות (RAM/Flash)
- חיישנים עם יציאות דיגיטליות 3.3V
- ממשקי תקשורת כמו UART, SPI, I2C

LVTTTL בהגדרת פינים

בכלי תכנון כמו Vivado, כשרוצים לחבר פין לכניסה/יציאה של רכיב חיצוני, חייבים להגדיר את ה-IOSTANDARD. דוגמה:

```
set_property IOSTANDARD LVTTTL [get_ports my_output]
```

בנוסף, צריך לוודא ש-Vcc0 של הבנק מחובר ל-3.3V (כי זה מה ש-LVTTTL דורש).

CMOS (Complementary Metal-Oxide-Semiconductor)

CMOS היא משפחת לוגיקה דיגיטלית המשתמשת בשילוב של טרנזיסטורי PMOS ו-NMOS לבניית שערים לוגיים. השילוב של PMOS ו-NMOS בשער לוגי מבטיח שבאף מצב לא יהיו שני טרנזיסטורים מוליכים בו זמנית בין ההזנה לקרקע – כך נמנעת זרימת זרם קבועה, ומכאן נובעת החיסכון בצריכת חשמל.

NMOS – טרנזיסטור שנפתח כשהמתח בכניסה גבוה (לוגי 1).

PMOS – טרנזיסטור שנפתח כשהמתח בכניסה נמוך (לוגי 0).

CMOS משתמשת בטרנזיסטורים מסוגים משלימים כדי לממש לוגיקה בצורה חסכונית בחשמל. מאפיינים של CMOS:

צריכת הספק נמוכה מאוד: הצריכה אפסית כמעט כשאין שינוי במצב הלוגי (כלומר, כל עוד לא מתבצע מיתוג).

מהירות גבוהה: בדגמים המודרניים, מהירות המיתוג דומה ואף גבוהה מ-TTL.

מתח עבודה גמיש: לרוב 3.3V או 5V, ויש גרסאות שעובדות גם עם 1.8V ו-1.2V.

עמידות בפני רעש גבוהה יותר לעומת TTL.

קל לייצר אותו בצפיפות גבוהה, ולכן כל המעבדים והזיכרונות המודרניים מבוססים עליו.

למה עברו מ-TTL ל-CMOS?

הסיבה העיקרית היא **חסכון בצריכת חשמל וצפיפות גבוהה**,

מה שקריטי בעולם של שבבים מודרניים כמו טלפונים, מחשבים ומערכות משובצות מחשב. TTL

עדיין קיים, אבל ברוב השימושים המודרניים CMOS שולט.

LVC MOS low voltage complementary metal-oxide-semiconductor

היא גרסה מתקדמת של טכנולוגיית CMOS הפועלת במתחי עבודה נמוכים יותר, ומתאימה בעיקר למעגלים מודרניים בהם נדרשת תקשורת מהירה וצריכת חשמל נמוכה עוד יותר.

LVC MOS היא משפחת אותות (I/O Standard) שמיועדת לפעול במתחי עבודה נמוכים – לרוב בין 1.2V ל-3.3V – תוך שמירה על תכונות ה-CMOS כמו צריכת הספק נמוכה והתנהגות לוגית דיגיטלית אמינה. היא נפוצה במיוחד בתכנון של **FPGA, ASIC**, מערכות משובצות, כרטיסים אלקטרוניים מודרניים, וממשקי תקשורת.

ASIC (ראשי תיבות של: Application-Specific Integrated Circuit) הוא מעגל משולב שתוכנן במיוחד עבור יישום מסוים. בניגוד למעבדים כלליים (כמו CPU או FPGA) שיכולים לבצע מגוון משימות, ASIC נבנה מראש כדי לבצע משימה אחת מוגדרת ויעילה במיוחד.

מאפיינים של LVC MOS:

מתחי עבודה נמוכים יותר מ-CMOS רגיל:

סטנדרטים נפוצים:

LVC MOS33 3.3V

LVC MOS25 2.5V

LVC MOS18 1.8V

LVC MOS12 1.2V

שיפור בצריכת הספק:

ככל שמתח העבודה יורד, גם ההספק הדינמי ($P = CV^2f$) יורד בצורה דרמטית.

תאימות למעגלים מודרניים:

מתאימה יותר לשבבים הפועלים במתחי ליבה נמוכים (core voltage).

נפוצה ב-FPGA ובממשקים מהירים

LVC MOS אינה טכנולוגיה חדשה של טרנזיסטורים, אלא הגדרה של רמות אות (I/O standards) עבור ממשקי קלט/פלט הפועלים במתחי CMOS נמוכים. כלומר, השבב כולו עשוי להיות CMOS, אבל קווי הקלט/פלט שלו יוגדרו לפי LVC MOS כדי להתאים למתחי עבודה מודרניים. אפשר לתאר את LVC MOS כ"שפת הדיבור" של הפינים במעגל – אותה שפה שצריכה להיות מוסכמת בין הרכיבים כדי שיוכלו להבין זה את זה ולתקשר בלי שגיאות או נזק.

Zynq 7000

הוא משפחה של מערכות על שבב (SoC) מבית Xilinx, המשלבת ליבת עיבוד מבוססת ARM עם לוגיקה ניתנת לתכנות (FPGA). זהו פתרון המשלב את הגמישות של FPGA עם כוח העיבוד של מעבד ARM, ומאפשר ביצועים גבוהים ויכולת התאמה אישית ליישומים מגוונים. רכיב אלקטרוני המשלב רכיבים מרובים של מחשב, כמו מעבד, זיכרון ובקרים, על גבי שבב סיליקון - SoC יחיד.

ארכיטקטורה משולבת:

Zynq 7000 מציע שילוב של מעבד ARM Cortex-A9 דו-ליבתי עם לוגיקה מתוכנתת (AMD 7 Series FPGAs). זה מאפשר למתכננים לנצל את היתרונות של שתי הפלטפורמות: יכולת עיבוד כללית של המעבד וגמישות התאמה אישית של FPGA.

יכולות עיבוד:

המעבד ARM Cortex-A9 מספק כוח עיבוד עבור משימות כלליות, מערכות הפעלה (כגון Linux) ויישומי תוכנה מורכבים.

גמישות FPGA:

ה-FPGA מאפשר למתכננים להתאים אישית את החומרה עבור משימות ספציפיות, כגון האצה של אלגוריתמים, עיבוד אותות, בקרת מנועים ועוד.

יתרונות השילוב:

השילוב של ARM ו-FPGA מאפשר ביצועים גבוהים בצריכת הספק נמוכה, מה שהופך את Zynq 7000 למתאים ליישומי בקרה, אנליטיקה, I/O וזיכרון.

פיתוח מקביל:

Zynq 7000 מאפשר פיתוח תוכנה במקביל עבור המעבד וה-FPGA, מה שמקצר את זמן הפיתוח ומאפשר גמישות רבה יותר בעיצוב.

יישומים:

Zynq 7000 משמש במגוון רחב של יישומים, כולל תעשייה, אוטומציה, רכב, תקשורת, רפואה ועוד.

PS – Processing System (מערכת עיבוד)

ה-PS הוא החלק ה"תכנותי" בשבב, ובדרך כלל כולל מעבד ARM (כמו Cortex-A9 או Cortex-A53). זהו אזור שתוכנן לבצע עיבוד סדרתי כללי, כמו כל מעבד קלאסי, ומריץ תוכנות כמו לינוקס, מערכות זמן אמת, או תוכנה ברמת bare-metal.

מאפיינים עיקריים של ה-PS:

- מעבדים כלליים (General-Purpose Processors): לרוב זוג ליבות ARM (כמו Cortex-A9), לפעמים יותר.
- יכולת הרצת מערכות הפעלה: כמו Linux, FreeRTOS, או תוכנות זמן אמת אחרות.
- גישה לזיכרון: כולל DDR, SRAM, Flash, Cache וכו'.
- היקפי תקשורת סטנדרטיים: UART, SPI, I2C, CAN, USB, Ethernet וכו' – כל אלו מופעלים מהצד של המעבד.
- שימושים נפוצים: ניהול מערכת, אינטראקציה עם המשתמש, עיבוד נתונים לא-קריטי, שליטה כללית על הרכיב.

יתרונות:

- קל לתכנות בשפות עיליות (C/C++/Python וכו').
- תמיכה רחבה בסביבות פיתוח ובמערכות הפעלה.
- מאפשר גישה נוחה למשאבים חיצוניים ופריפריות.

PL – Programmable Logic (לוגיקה ניתנת לתכנות)

ה-PL הוא החלק של ה-FPGA – לוגיקה שניתנת לתכנות על ידי המשתמש לצורך יצירת חומרה מותאמת אישית. כאן כותבים קוד ב-VHDL, Verilog או בשפות ברמה גבוהה (HLS – High-Level Synthesis) כדי ליישם מעגלים לוגיים, יחידות עיבוד מקבילי, בקרי תקשורת מהירה, וכו'.

מאפיינים עיקריים של ה-PL:

- לוגיקה הניתנת לתכנות: שערים לוגיים, Flip-Flops, Multiplexers, לוחות ניתוב (Routing), ועוד.
- יחידות DSP מובנות: לצורך עיבוד אותות, FFT, פילטרים וכו'.
- Block RAM: זיכרון פנימי מהיר וניתן לתצורה.
- Clock Management Units: יחידות ניהול תזמון (PLL, MMCM).
- אפשרות ליצירת פריפריות מותאמות אישית: כמו SPI, Ethernet, AXI, CAN וכו', או אפילו מאיצים (accelerators) לחישובים כבדים.
- שימושים נפוצים: עיבוד אותות, עיבוד תמונה, תקשורת מהירה, בקרת חומרה מדויקת, רובוטיקה, ויישומים שדורשים זמן תגובה מהיר או עיבוד מקבילי כבד.

יתרונות:

- ביצועים גבוהים – במיוחד בעיבוד מקבילי.
- שליטה מלאה על מבנה החומרה.
- יכולת ליישם מעגלים מותאמים ספציפית לצורך מסוים.
- ניתן להשתמש בו כמאיץ חומרה (hardware accelerator) לפונקציות מתוך ה-PS.

PS ו-PL – עבודה משולבת

מה שמייחד רכיבים כמו Zynq הוא היכולת לשלב בין PS ל-PL בצורה הדוקה:

- AXI Interconnect: מאפשר תקשורת דו-כיוונית בין ה-PS ל-PL דרך ממשקים סטנדרטיים.
- DMA Controllers: מאפשרים העברת נתונים מהירה בין זיכרון (בצד של ה-PS) ללוגיקה (בצד של ה-PL).
- Interrupts: ניתן לשלוח פסיקות (Interrupts) בין הצדדים.

דוגמה לשיתוף פעולה בין PS ל-PL:

הצד של ה-PS מריץ לינוקס ומטפל בקלט מהמשתמש. לינוקס שולח קובץ וידאו לצורך עיבוד. הצד של ה-PL מבצע עיבוד תמונה (למשל סינון או דחיסה) באופן מקבילי ומהיר. הנתונים המעובדים נשלחים חזרה ל-PS או ישירות ל-Display.

כיצד FPGA והמעבד מתקשרים

AXI Advanced eXtensible Interface

הוא מפרט ממשק המשמש בתחום האלקטרוניקה, בעיקר בתכנון מערכות משובצות מחשב. הוא מתאר את האופן שבו רכיבי IP שונים מתקשרים זה עם זה בתוך מערכת. פרוטוקול AXI הוא חלק ממשפחת הפרוטוקולים ARM ונועד לספק ביצועים גבוהים וגמישות בעיצוב מערכות מורכבות. הערוצים ב-AXI נפרדים ולכן ניתן להשתמש בהם בצורה מקבילית – למשל לשלוח בקשת כתיבה תוך כדי קריאת מידע אחר.

תכונות עיקריות של AXI:

- תמיכה ברוחב פס גבוה – מאפשר העברת מידע ביעילות גבוהה מאוד.
- ערוצים נפרדים לכתיבה וקריאה – כולל ערוצי כתובת, נתונים, ואישורים.
- תמיכה בבקשות מקביליות (out-of-order) – מאפשר לבקשות להגיע ולחזור בסדר שונה.
- תמיכה ב-burst transfers – העברת בלוקים של נתונים ברצף אחד.
- חיבור בין רכיבי IP בצורה סטנדרטית – מאפשר חיבור פשוט של רכיבי צד שלישי.

הסבר מפורט:

ממשק תקשורת:

AXI הוא מפרט המגדיר את ה"שפה" שבה רכיבי IP שונים מדברים זה עם זה. הוא קובע את הפקודות, הכתובות, הנתונים והאיתותים המשמשים להעברת מידע בין הרכיבים.

תאימות וגמישות:

AXI תוכנן להיות גמיש ומודולרי, כך שניתן לשלב בקלות רכיבי IP שונים ממקורות שונים לתוך מערכת אחת. הממשק האחיד מקל על האינטגרציה ומפחית את הצורך בשינויים מורכבים בקוד.

ביצועים גבוהים:

AXI תומך במספר תכונות המאפשרות העברת נתונים מהירה ויעילה, כולל תמיכה ברוחב פס גבוה, תזמון מורכב ושימוש במספר ערוצי נתונים במקביל.

מנהל ומנוהל:

AXI מגדיר שני סוגים עיקריים של ממשקים: מנהל (Manager) ומנוהל (Subordinate). המנהל הוא הרכיב שמתחיל את העברת הנתונים, והמנוהל הוא הרכיב שמקבל או שולח את הנתונים. כל חיבור AXI הוא בין ממשק מנהל לממשק מנוהל.

בהקשר של ZYNQ 7000 המנהל הוא המעבד המנוהל הוא הFPGA

יישומים:

AXI נמצא בשימוש נרחב בתכנון של מערכות משובצות מחשב, כולל מעבדים, בקרים, זיכרון, יחידות עיבוד גרפי (GPU), מערכות מולטימדיה ועוד.

בדיקות איכות:

AXI משמש גם בתהליכי בדיקה של מערכות אלקטרוניות, כגון בדיקת AXI Inspection (AXI), המשמשת לזיהוי תקלות ופגמים בלוחות מעגלים מודפסים (PCBs).

דוגמה ב-Vivado Block Design:

כאשר אתה מוסיף את ZYNQ7 Processing System ל-Block Design: יש אפשרות להפעיל AXI Master interfaces בצד ה-PS ואז לחבר IP מותאם אישית דרך AXI Interconnect אפשר גם לחבר AXI DMA להעברת מידע מהירה (למשל, לווידאו או אודיו)

איך AXI עובד

חמישה ערוצים נפרדים ב-AXI

פעולת כתיבה כוללת שלושה ערוצים:

Address Write – המעבד שולח את כתובת הכתיבה
Write Data – שולח את הנתון לכתיבה
Write Response – מקבל אישור שהכתיבה הצליחה

פעולת קריאה כוללת שני ערוצים:

Address Read – המעבד שולח את כתובת הקריאה
Read Data – מקבל את הנתון המבוקש

דוגמא:

- AW - המעבד שולח כתובת (למשל 0x1000)
- W - המעבד שולח את הנתון (למשל 0xDEADBEEF)
- B - respond Back - המנוהל מחזיר תגובה (OKAY)
- AR - המעבד מבקש כתובת קריאה
- R - המנוהל שולח את הנתון מהכתובת

תכונות מתקדמות

Burst Transfers:

שליחה/קריאה של מספר מילים ברצף אחד – יעיל לזיכרון.

Out-of-Order Support:

אפשר לשלוח כמה בקשות ולהחזיר תשובות בסדר שונה (אם המערכת תומכת).

Valid/Ready Handshake:

מנגנון Handshake הוא חלק מרכזי בפרוטוקול AXI ומאפשר תיאום מדויק ואמין של העברת נתונים בין רכיבים, תוך שמירה על גמישות וביצועים גבוהים גם כאשר קיים הבדל במהירות או בעומס בין הצדדים. כל אחד מחמשת הערוצים AW, W, B, AR, R משתמש בזוג אותות:

VALID – נשלח על ידי הצד שמספק את המידע (השולח), ומסמן שהמידע זמין ותקין להעברה.
READY – נשלח על ידי הצד שמקבל את המידע (המקבל), ומסמן שהוא מוכן לקלוט את המידע.
העברת המידע מתבצעת רק כאשר גם VALID וגם READY הם ב-'1' באותו מחזור שעון.

תכונות עיקריות של Handshake:

מאפשר עבודה בקצבים שונים בין הרכיבים – כל צד קובע מתי הוא מוכן.
משפר גמישות מערכתית – ניתן "לעצור" זמנית שליחה או קבלה מבלי לאבד נתונים.
כל ערוץ פועל עצמאית – למשל, ניתן להמתין ל-READY ב-W תוך כדי שממשיכים לשלוח AR.
מאפשר פעולה יציבה גם בעומסים כבדים או בתקשורת לא סימטרית.

GIT

גיט זה מערכת לניהול גרסאות (Version Control System) בקוד פתוח, שמאפשרת לעקוב אחרי שינויים בקבצים לאורך זמן. היא פותחה על ידי לינוס טורבאלדס (יוצר לינוקס) בשנת 2005.

למה משתמשים ב-Git?

- לשמור היסטוריה של קוד ושינויים.
- לעבוד בצוות בלי לדרוך אחד על העבודה של השני.
- לחזור לגרסאות קודמות בקלות.
- לאחד שינויים ממספר מקורות (branches).

מושגים בסיסיים בגיט:

Repository - תיקייה שבה מנוהל הקוד עם היסטוריית השינויים
Commit - שמירה של סט שינויים עם הודעה שמתארת מה שונה
Branch - גרסה מקבילה של הקוד שנועדה לפיתוח מבלי לפגוע בקוד הראשי
Merge - איחוד בין שני ענפים
Clone - העתקה של מאגר קיים מהמחשב או מהאינטרנט (כמו מ-GitHub)
Pull - הורדת שינויים מהשרת למחשב המקומי
Push - שליחת שינויים מהמחשב לשרת

כלים משלימים:

GitHub / GitLab / Bitbucket – שירותי אירוח לפרויקטים עם ממשק גרפי וניהול צוותים.
VS Code / SourceTree – כלים עם ממשק גרפי לעבודה עם Git.

פקודות בסיסיות בגיט:

- git init – יצירת מאגר Git חדש בתיקייה קיימת.
- git status – מציג את מצב הקבצים (מה השתנה, מה מוכן ל-commit).
- git add – מוסיף קבצים ל-"stage" לפני commit (למשל git add file.txt או git add .).
- git commit -m "הודעה" – יוצר commit עם ההודעה הנתונה.
- git log – מציג את היסטוריית ה-commits.
- git checkout – מעבר בין ענפים (branches) או שיחזור קובץ לגרסה מסוימת.
- git branch – מציג את הענפים הקיימים או יוצר ענף חדש.
- git merge – מאחד ענף נבחר עם הענף הנוכחי.
- git remote – מציג או מגדיר כתובת של מאגר מרוחק (כמו GitHub).
- git pull – שואב שינויים ממאגר מרוחק וממזג אותם.
- git push – שולח שינויים למאגר מרוחק.

מצבים של קובץ בגיט:

- Untracked – קובץ ש-Git לא עוקב אחריו עדיין.
- Staged – קובץ שמוכן ל-commit.
- Modified – קובץ ששונה אך עדיין לא עבר ל-stage.
- Committed – שינויים שכבר נשמרו בהיסטוריה.

פיפו (First In First Out)

FIFO הוא מבנה נתונים שבו סדר הכתיבה והקריאה נשמר בדיוק כפי שהנתונים נכנסו – הראשון שנכנס הוא הראשון שייצא.

במערכות חומרה כמו FIFO, FPGA משמש לביצוע תקשורת פנימית בין מודולים, תוך שמירה על הסדר ופתרון בעיות של מהירויות שונות בין חלקים שונים של המערכת.

תפקידי FIFO:

- Buffer זמני לנתונים זורמים.
- חיץ בין רכיבים עם קצבים שונים.
- מעבר נתונים בין תחומי שעון שונים (CDC – Clock Domain Crossing).

איך FIFO עובד?

FIFO בנוי ממספר רכיבים בסיסיים:

- זיכרון פנימי – לרוב מערך בגודל קבוע (למשל 16 תאים), בו כל תא מאחסן מילה בגודל קבוע (למשל 8 ביטים).
- מונה כתיבה (Write Pointer) – מצביע על מיקום הכתיבה הבא. לאחר כל כתיבה, המונה עולה.
- מונה קריאה (Read Pointer) – מצביע על מיקום הקריאה הבא. לאחר כל קריאה, המונה עולה.
- אותות בקרה (Control Signals):
 - full – מסמן שהזיכרון מלא, ואין מקום לכתוב עוד.
 - empty – מסמן שאין נתונים.

מתי משתמשים ב-FIFO?

- הפרדת שעונים (CDC) – לדוגמה, כשמודול UART מקבל נתונים משעון מערכת אחר.
- איזון קצבים – כמו למשל, בין עיבוד מהיר של DSP לבין גישה איטית לזיכרון.
- מערכות תקשורת – FIFO כמעט תמיד נמצא בממשקי תקשורת: SPI, UART, AXI-Stream, Ethernet.
- תזמון פנימי של מערכת – לדוגמה, כשממתינים להפעלת תהליך על פי סדר.

סוגי FIFO

סינכרוני - קריאה וכתבייה באותו שעון
אסינכרוני - קריאה וכתבייה בשני תחומי שעון נפרדים

איך משתמשים ב-FIFO ב-VHDL?

- אפשרות ראשונה – יצירת FIFO ידנית בקוד VHDL
אתה מגדיר מערך, pointers, תנאים ל-full ו-empty, ומכניס לוגיקה של קריאה וכתבייה.
- אפשרות שנייה – שימוש באשף IP של Vivado (FIFO Generator)
Vivado מספק IP מוכן לשימוש:
נכנסים ל-IP Catalog
בוחרים ב-FIFO Generator
מגדירים: רוחב הנתונים, גודל, מצב סינכרוני/אסינכרוני, וכו'
יוצרים wrapper VHDL שמתחבר ל-IP

דוגמה בסיסית לקוד FIFO (סינכרוני):

```
signal fifo_mem : array(0 to 15) of std_logic_vector(7 downto 0);  
signal wr_ptr, rd_ptr : integer range 0 to 15 := 0;  
signal full, empty : std_logic;
```

בכל פעולת write, תכתוב לתא ש-wr_ptr מצביע עליו, ואז תעלה אותו (אם לא מלא).
בכל read, תקרא מהתא ש-rd_ptr מצביע עליו, ואז תעלה אותו (אם לא ריק).

טיפול במעגליות (Wrap Around)

ב-FIFO קלאסי, כשה-pointers מגיעים לסוף המערך, הם צריכים "להתגלגל" חזרה להתחלה:
 $wr_ptr \leq (wr_ptr + 1) \bmod 16;$
 $rd_ptr \leq (rd_ptr + 1) \bmod 16;$
זו טכניקה שמבטיחה שהמצביעים יישארו תמיד בטווח 0–15 (במקרה של 16 תאים).

מצבים מיוחדים

Almost Full / Almost Empty – שימושי במיוחד כאשר יש צורך להתריע מראש לפני שהתור יתמלא או יתרוקן לגמרי.
Level Indicator – מצביע כמה תאים מלאים בפועל (יכול לשמש לניטור עומסים או דיבאגינג).

איך הקובץ שנוצר על ידי ה-generate bitstream מגיע ל-FPGA?

במהלך תהליך ה-Bit Stream התוכנה ממירה את הלוגיקה של התוכנית לקובץ של ביטים, הקובץ בדרך כלל עם הסיומת .BIT.
כאשר לוחצים על הכפתור PROGRAM DEVICE, התוכנה שולחת דרך הכבל micro usb בעזרת ממשק JTAG את הקובץ אל תוך ה-FPGA.
- ממשק JTAG הוא פרוטוקול תקשורת סדרתי שנועד במקור לבדיקות חומרה אבל עם הזמן הפך לכלי מרכזי גם בצריבת ה-FPGA, דיבוג ותכנות מעבדים.
הנתונים נטענים לתוך זיכרון הקונפיגורציה של ה-FPGA שהוא בדרך כלל SRAM פנימי.
זיכרון זה הוא נדיף, כלומר ברגע שמכבים את החשמל ה-FPGA שוכח את התוכן.
אם אתה רוצה שהקוד ישמר גם אחרי כיבוי, עלינו להשתמש ב-add configuration memory device כדי לתכנת את ה-FLASH החיצוני (לרוב SPI FLASH).

פלטפורמת Vitis

וִיטִיס היא סביבת פיתוח תוכנה מאת חברת AMD המיועדת לפתח יישומים המשלבים FPGA ומעבדי ARM. היא מאפשרת למפתחים לבנות ולמטב יישומים מואצים בחומרה על גבי פלטפורמות אדפטיביות של AMD, תוך שימוש בספריות מואצות חומרה בקוד פתוח וסביבות פיתוח ספציפיות לתחום על פי AMD. Vitis היא פלטפורמה מאוחדת המיועדת למגוון רחב של מפתחים, כולל מהנדסי תוכנה ומדעי AI.

הפלטפורמה מאפשרת האצת יישומים באמצעות חומרה אדפטיבית, תוך שימוש בכלים מותאמים לפיתוח חומרה ותוכנה. Vitis כוללת ערכת פיתוח ליבה מקיפה לבניית יישומים מואצים, ספריות קוד פתוח מואצות חומרה המותאמות לפלטפורמות AMD, וסביבות פיתוח ספציפיות לתחום המאפשרות פיתוח ישירות במסגרת כלים מוכרים וברמה גבוהה יותר על פי CMC Microsystems. Vitis Embedded היא חבילת פיתוח תוכנה עצמאית לפיתוח וקומפילציה של תוכנת ++C/C++ עבור תתי-מערכות עיבוד משובצות AMD (מבוססות ARM ו-MicroBlaze) במערכות SoC ו-FPGA אדפטיביות של AMD, על פי AMD.

Vitis פועלת בשילוב עם חבילת העיצוב של AMD Vivado כדי לספק רמת הפשטה גבוהה יותר לפיתוח העיצוב.

איך לתפעל את Vitis

- הורדת התוכנות Vitis ו-Vivado

VIVADO:

- לחבר את לוח ה-FPGA
- יצירת פרויקט VIVADO
- בניית ארכיטקטורה
- יצירת export hardware bitstream
- שמירת קובץ ה-XSA
- קובץ XSA הוא קובץ שיוצא מ-VIVADO וכולל את כל המידע הדרוש ל-Vitis על החומרה שיוצרנו לדוגמא מפת זכרון רכיבי IP חיבורים בין רכיבים וקובץ BITSTREAM לצריבה.

VITIS:

- לפתוח את Vitis וליצור פרויקט חדש
- לבחור את קובץ ה-XSA
- לבחור פלטפורמת יעד
- לבחור סוג פרויקט (C)
- כתיבת הקוד באמצעות C
- להשתמש ב-Vitis Debugger או XSCT Console כדי לצרוב את הקוד

סוגי זכרון

ROM – Read Only Memory

תיאור: זיכרון לקריאה בלבד – התוכן שלו נכתב בזמן הייצור ואינו משתנה.

שימושים: קוד אתחול (Bootloader), קושחה (Firmware) קבועה.

תכונות:

- לא נדיף (לא נמחק בעת כיבוי).
- תוכן קבוע ולא ניתן לשנותו.

RAM (Random Access Memory)

תיאור: זיכרון לקריאה וכתיבה בגישה אקראית, לשימוש זמני במהלך פעולת המערכת.

שימושים: אחסון זמני של נתונים ותוכניות בזמן ריצה.

תכונות:

- נדיף (נמחק בעת כיבוי).
- מהיר בהרבה מאחסון קבוע.

SRAM – Static RAM

תיאור: זיכרון מהיר מאוד שלא מצריך רענון מתמיד.

שימושים: זיכרון מטמון (Cache), לוגיקה פנימית ב-FPGA, רשומות מהירות.

תכונות:

- נדיף.
- מהיר ויקר יותר מ-DRAM.
- תופס יותר שטח סיליקון.

DRAM – Dynamic RAM

תיאור: זיכרון איטי יותר שדורש רענון מתמיד, אך חסכוני בנפח.

שימושים: זיכרון עבודה עיקרי במחשב (כמו DDR).

תכונות:

- נדיף.
- איטי מ-SRAM אך זול יותר.
- דורש בקר לרענון.

DDR – Double Data Rate

תיאור: גרסה מתקדמת של DRAM עם יכולת העברת מידע פעמיים בכל מחזור שעון.

שימושים: זיכרון ראשי מהיר במחשבים, במערכות SoC ו-FPGA עם בקר DRAM.

תכונות:

- נדיף.
- מהיר בהרבה מ-DRAM רגיל.
- פועל בתיאום עם בקר זיכרון.

Flash Memory

תיאור: זיכרון קבוע הניתן למחיקה וכתיבה מחדש.

שימושים: אחסון קוד קבוע (firmware), שמירת bitstream ב-FPGA, כונני SSD.

תכונות:

- לא נדיף.
- איטי מ-RAM אך שומר מידע.
- ניתן לכתוב ולמחוק מספר רב של פעמים (אך לא אינסופי).

BRAM

Block RAM הוא רכיב זיכרון פנימי המצוי ב-FPGA, הבנוי מבלוקים נפרדים של זיכרון RAM בגודל קבוע (לרוב 18Kb או 36Kb), שניתן להשתמש בהם לצורכי אחסון נתונים מהיר.

BRAM נבנה כחלק אינטגרלי מה-FPGA.

בניגוד לזיכרון חיצוני, הוא מהיר מאוד ונגיש מהלוגיקה של ה-PL (Programmable Logic).

BRAM הוא זיכרון גישה אקראית (RAM), כלומר מאפשר קריאה וכתיבה לפי כתובת.

ניתן לגשת אליו בצורה סינכרונית – קריאה/כתיבה מתוזמנת לשעון.

פועל לפי כניסות של:

- Address (כתובת)
- Data In (נתונים להכניס)
- Data Out (תוצאה בקריאה)
- WEA – Write Enable (סימון אם יש כתיבה)
- ENA – Enable (מאפשר את פעולת הקריאה או הכתיבה)
- CLK – שעון

ברוב ה-FPGAs של Xilinx (כמו סדרת Zynq/Artix/Kintex):

- כל בלוק BRAM הוא בגודל של 18Kb (2048 כתובות של 9 סיביות למשל).
- ניתן לשלב 2 בלוקים ל-36Kb.
- אפשר לקבוע את רוחב הנתונים: 1, 2, 4, 8, 16, 32, 64... סיביות.
- ניתן להגדיר Single-Port (פורט אחד לקריאה/כתיבה) או Dual-Port (שני פורטים נפרדים – לקריאה וכתיבה במקביל או שתי קריאות במקביל).

שימושים נפוצים

- אחסון טבלאות Lookup (כמו LUTs לדחיסה/פענוח).
- FIFO בין רכיבים שונים.
- frame buffers לתמונות ווידאו.
- זיכרון ביניים לפרויקטי DSP.
- אימפלמנטציה של ROM או RAM ב-VHDL/Verilog.
- אחסון נתונים עבור state machines או מעבדים מוטמעים.

אותות חשובים:

- CLK שעון סינכרוני
- ENA הפעלת הגישה לזיכרון
- WEA איתות כתיבה – 1 = כתוב, 0 = קרא
- ADDR כתובת הזיכרון (בד"כ וקטור של 9–14 ביט)
- DIN נתונים לכתיבה
- DOUT נתונים ביציאה (בתום פעולת קריאה)
- RSTA reset אסינכרוני (אופציונלי)

BRAM IN PS

שלב 1: VIVADO:

- הוסף Block Memory Generator ב- Block Design.
- קבע את הגודל (18Kb, 36Kb וכו') וחבר אותו ל- Processing System.
- הגדר את החיבורים הנדרשים (כתובת, נתונים, WEA, ENA).

שלב 2: VITIS:

- ב- Vitis, השתמש ב-API לגישה ל-BRAM.
- השתמש בכתובת של ה-BRAM שהגדרת ב-Vivado והגדר את פעולות הקריאה/כתיבה בתוכנה.
- השתמש בפקודות כמו Xil_Out32 ו-Xil_In32 לגישה לכתובות הזיכרון.

```
#define BRAM_BASE_ADDR 0x40000000 // כתובת ה-BRAM
void write_bram(uint32_t addr, uint32_t data) {
    Xil_Out32(addr, data); // כתיבה ל-BRAM
}
uint32_t read_bram(uint32_t addr) {
    return Xil_In32(addr); // קריאה מ-BRAM
}
int main() {
    uint32_t data_to_write = 0x12345678;
    write_bram(BRAM_BASE_ADDR, data_to_write); // כתיבה
    uint32_t data_read = read_bram(BRAM_BASE_ADDR); // קריאה
    return 0;
}
```

UART Lite

UART Lite (ראשי תיבות של *Universal Asynchronous Receiver/Transmitter Lite*) הוא רכיב תקשורת פשוט יחסית (IP Core) של Xilinx, המאפשר תקשורת אסינכרונית בין רכיבים במערכת, לדוגמה:

בין FPGA למחשב או מיקרו-בקר.
הוא גרסה "קלילה" של UART רגיל, שמתאימה למערכות שלא צריכות את כל הפיצ'רים המתקדמים.

תכונות עיקריות של UART Lite:

- תמיכה בשידור (Tx) וקבלה (Rx) של נתונים.
- FIFO (תור נתונים) קטן בגודל 16 בתים לשידור ולקבלה.
- מהירות קבועה מראש – קצב הבאוד נקבע בזמן הסינתזה (לא משתנה בזמן ריצה).
- אין תמיכה בהפרעות (Interrupts) מתקדמות או בבקרת זרימה (flow control).
- ממשק פשוט מאוד לתכנות, לרוב דרך רגיסטרים.

רגיסטרים חשובים:

- Tx FIFO – כתיבה לנתונים שישודרו.
- Rx FIFO – קריאה מהנתונים שנתקבלו.
- Status Register – מציג האם יש נתונים בקלט/פלט, האם הפיפו מלא או ריק.
- Control Register – מאפשר לבצע פעולות פשוטות כמו איפוס הפיפו.

```
#include "xuartlite.h"
#include "xparameters.h"

XUartLite uart;

int main() {
    XUartLite_Initialize(&uart, XPAR_UARTLITE_0_DEVICE_ID);

    u8 msg[] = "Hello UART!\n";
    XUartLite_Send(&uart, msg, sizeof(msg) - 1);

    return 0;
}
```

שימושים נפוצים:

- שליחת פלט טקסטואלי למחשב לצורך דיבוג.
- תקשורת בין שני רכיבים ללא צורך בפרוטוקול מורכב.
- קלט ממשתמש (למשל מקלדת דרך טרמינל סריאלי).

UARTPS

UARTPS הוא רכיב תקשורת סידורית אסינכרונית המובנה בתוך המעבד ARM Cortex-A9 שב-Zynq Processing System. הוא שונה מ-UARTLite בכך שהוא יותר מתקדם, כולל יותר פיצ'רים, תומך באינטרפטים, FIFO, קצב באוד משתנה, ועוד.

מאפיינים עיקריים:

- תקשורת אסינכרונית (ללא אות שעון משותף)
- מהירויות באוד שונות (baud rate)
- תמיכה ב-FIFO (לשולח ולמקבל) של 64 בתים
- תמיכה ב-Interrupts (לקריאה או כתיבה)
- תומך בפרמטרים כמו: Stop Bits, Parity, Data Bits

רגיסטרים:

- XUARTPS_SR_OFFSET Status Register - מכיל דגלים על מצב המודול (ריק, מלא, שגיאות וכו')
- XUARTPS_CR_OFFSET Control Register - שולט על הפעלת המודול, Reset לפיפו וכו'
- XUARTPS_ISR_OFFSET Interrupt Status Register
- XUARTPS_FIFO_OFFSET כתובת הפיפו לקריאה/כתיבה של נתונים

קריאה מ-UART:

בדוק אם FIFO של קלט לא ריק (Rx FIFO) באמצעות Status Register.
אם יש נתון - קרא אותו מה-FIFO.

```
u32 status = XUartPs_ReadReg(BaseAddress, XUARTPS_SR_OFFSET);
if (!(status & XUARTPS_SR_RXEMPTY)) {
    u8 received = XUartPs_ReadReg(BaseAddress, XUARTPS_FIFO_OFFSET);
}
```

שליחה ל-UART:

בדוק אם FIFO של שידור לא מלא.
כתוב תו ל-FIFO לשליחה.

```
u32 status = XUartPs_ReadReg(BaseAddress, XUARTPS_SR_OFFSET);
if (!(status & XUARTPS_SR_TXFULL)) {
    XUartPs_WriteReg(BaseAddress, XUARTPS_FIFO_OFFSET, data);
}
```

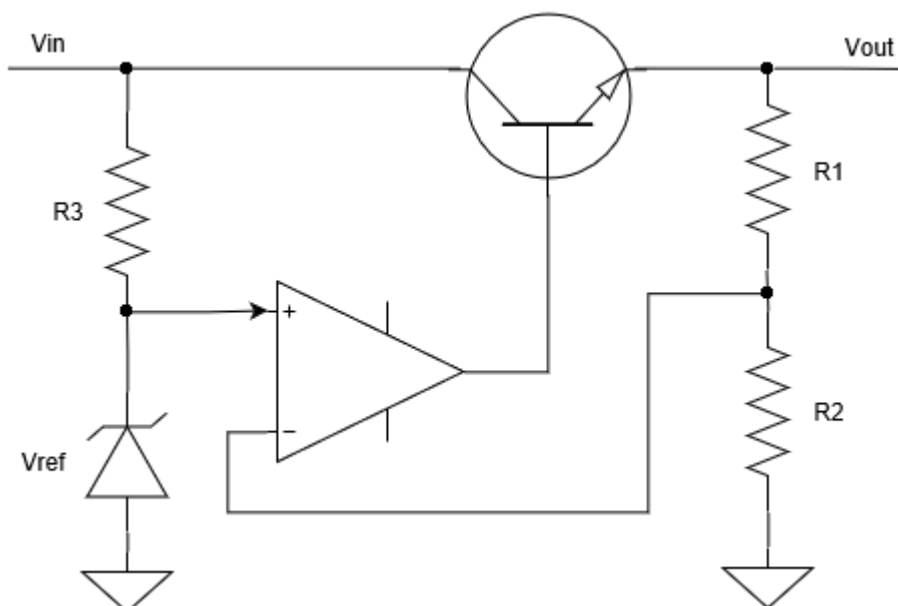
שימושים עיקריים:

- שליחה/קבלה של מידע בין ה-FPGA למחשב (דרך USB-UART)
- תקשורת עם רכיבי חומרה חיצוניים (מודול GPS, חיישנים)
- Debugging (הדפסה סידורית של מידע)

פונקציות שימושיות בספריית Xilinx

- XUartPs_Recv() - קבלת תווים
- XUartPs_Send() - שליחת תווים
- XUartPs_CfgInitialize() - אתחול
- XUartPs_SetBaudRate() - הגדרת קצב באוד

מייצבי מתח ליניארים



מבוססים על טרנזיסטור שפועל באזור הפעיל (Active Region), כלומר משמש כנגזרת משתנה של נגד. לולאת משוב שלילית (Feedback Loop) משווה את מתח היציאה למתח ייחוס פנימי (לרוב 1.25V או 5V), ומווסתת את הולכת הטרנזיסטור בהתאם. עורך המתח ($V_{in} - V_{out}$) מתבזז כחום:

$$P_{loss} = (V_{in} - V_{out}) \times I_{load}$$

טרנזיסטור (NPN / PNP או MOSFET) – מווסת את המתח ע"י שינוי ההתנגדות הפנימית. Reference Voltage – מקור מתח ייחוס מדויק (למשל דיודת זנר או bandgap reference). מגבר שגיאה (Operational Amplifier) – משווה בין מתח היציאה לרפרנס ושולט בטרנזיסטור.

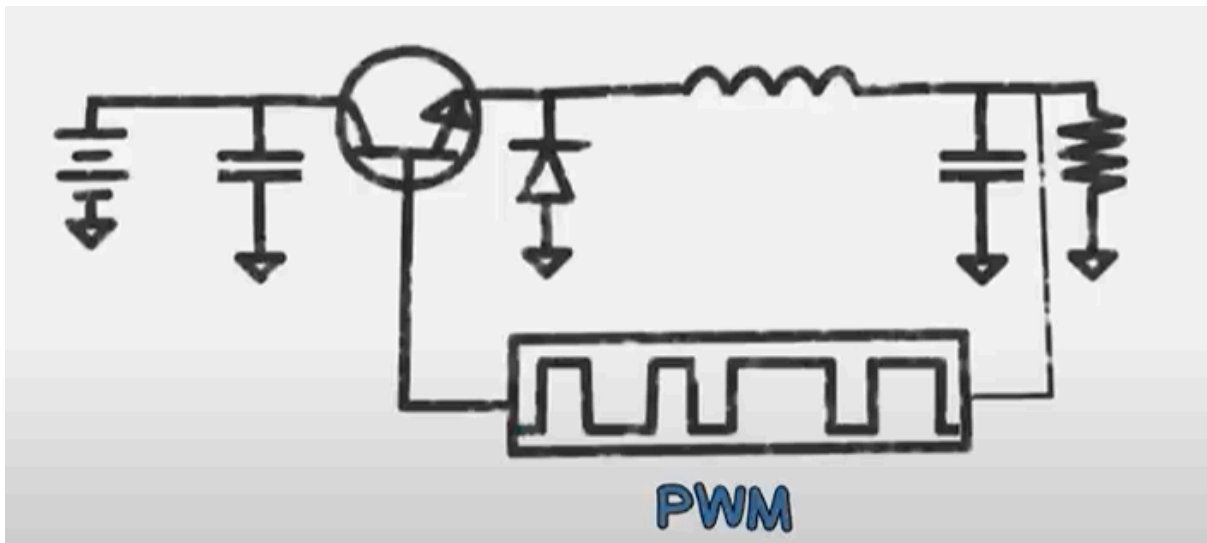
יתרונות:

- מעגל פשוט מאוד – קל לתכנן וליישם.
- עלות נמוכה.
- רעש נמוך מאוד – מתאים למעגלים רגישים (כמו אודיו ו-RF).
- תגובה מהירה לשינויים בזרם העומס.

חסרונות:

- יעילות נמוכה – ההפרש בין V_{in} ל- V_{out} מתבזז כחום.
- נדרש גוף קירור בהספקים גבוהים.
- לא יעילים כשיש הפרש גדול בין מתח הכניסה ליציאה.

מייצבי מתח ממותגים



מבוססים על מיתוג מהיר של טרנזיסטור (לרוב MOSFET) בתדר גבוה (100kHz–2MHz). האנרגיה מועברת במחזוריות דרך רכיבים אגראי-אנרגיה – סליל (Inductor) וקבל (Capacitor) – תוך שימוש בטופולוגיות כמו:

- Buck – להורדת מתח
 - Boost – להגברת מתח
 - Buck-Boost – להורדה והגברה לפי הצורך
- לאחר המיתוג, האות (לרוב גל ריבוע) עובר פילטר באמצעות סליל וקבל לקבלת מתח DC חלק ויציב.

בקר מיתוג (PWM Controller) – יוצר את PWM מותאם.
MOSFET מהיר – משמש כמפסק (ON/OFF) עם זמן מעבר קצר.
דיודה או MOSFET שני – מספקים מסלול לזרם בזמן שהמפסק סגור.
סליל (Inductor) – אוגר אנרגיה מגנטית ומאפשר זרם רציף לעומס.
קבל פילטר (Capacitor) – מייצב את מתח היציאה ומפחית תנודות.
מעגל משוב – משווה את מתח היציאה למתח הייחוס ומכוון את מחזור העבודה (Duty Cycle).

שלב 1 – מיתוג ON:

- ה-MOSFET נפתח.
- זרם זורם דרך הסליל לעומס – הסליל אוגר אנרגיה והזרם עולה בהדרגה.
- מתח היציאה עולה.

שלב 2 – מיתוג OFF:

- ה-MOSFET נסגר.
- הדיודה או המפסק השני מאפשרים לזרם להמשיך לזרום דרך הסליל לעומס (בזכות האנרגיה שנאגרה).
- הסליל משחרר אנרגיה והקבל מייצב את המתח.
- הבקר מתאם את מחזור העבודה כדי לשמור על מתח יציב.

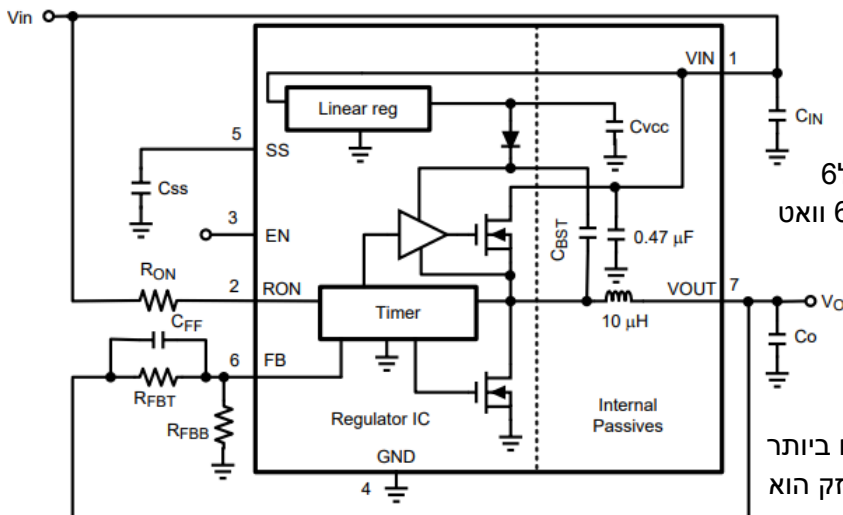
יתרונות:

- יעילות גבוהה מאוד (לעיתים מעל 90%).
- מסוגלים להעלות או להוריד מתח.
- התחממות נמוכה – פחות צורך בגופי קירור.
- אידיאלי לצריכת הספק גבוהה או עבודה מסוללות.

חסרונות:

- מעגל מורכב יותר.
- עלות גבוהה יותר.
- רגישות לרעש – יוצרים הפרעות אלקטרומגנטיות (EMI).
- תגובה איטית יותר לשינויים חדים בזרם עומס, יחסית לליניאריים.
- הסליל חיוני לאגירת אנרגיה ולהחלקת זרם – הזרם דרכו לא משתנה מיידית, מה שנותן יציבות.
- הקבל מסייע ביצירת מתח חלק ותגובה מהירה לשינויים מידיים.
- נדרש תכנון מוקדם של מיקום סלילים, מסלולים, קבלי דה-קפלינג ושיכוך רעש (EMI filtering).

רכיב LMZ14201



הLMZ הוא ממיר מתח ממוטג, רכיב זה מסוגל לספק זרם של עד 1 אמפר ביציאה, יש לו יעילות של 90%, הוא יכול לקבל מתח קלט בין 6 ל 42 וולט ולספק מתח יציאה מתכוון ומדויק בין 0.8 ל 6 וולט, הספק היציאה הכולל של הרכיב הוא 6 וואט הרכיב מופעל במלואו עבור WEBENCH® Power Designer (כלי מקוון של שמאפשר לתכנן במהירות ובקלות מעגלי ספק כוח מותאמים אישית עם סימולציות והמלצות רכיבים).

טווח הטמפרטורות שבו האזור הפנימי החם ביותר ברכיב אלקטרוני יכול לפעול בבטחה וללא נזק הוא בין 40 ל 125 מעלות צלזיוס. הרכיב יכול להתמודד עם פריקה של עד ± 2000 וולט לפי מודל גוף האדם (HBM).

שימושים של LMZ

- המרת מתח ממסילות כניסה של 12 וולט ו-24 וולט למתחים המתאימים לצריכת הרכיבים המקומית.
- פרויקטים שתלויים בזמן
- יישומים עם מגבלות מקום ודרישות תרמיות גבוהות
- יישומים עם מתח יציאה שלילי

יתרונות שימוש בLMZ

- פועל בטמפרטורות סביבה גבוהות ללא הפחתת הספק תרמית
- יעילות גבוהה המפחיתה את יצירת החום במערכת
- פליטת קרינה אלקטרומגנטית (EMI) נמוכה, נבדקה לפי תקן EN55022 Class B
- כמות רכיבים חיצוניים נמוכה

תיאור פעולת הרכיב:

מעגל הFEEDBACK

כדי לנחית ולייצב את המתח ביציאה, הרכיב עושה מעין בדיקה כלשהי, נגיד והרכיב רק הופעל, מתח היציאה שלו יהיה 0 וולט, בגלל מחלק המתח עם הנגדים המתח בFB יהיה גם 0 וולט ה0 וולט נכנס למשווה (מגבר שגיאה) שבו יש את הערכים הבאים:

$$v_o = v_{ref} (1 + r_{fbt} / r_{fbb})$$

$$0 = v_{ref} (x)$$

$$v_{ref} = 0$$

$$FBV - REFV = Error$$

מתח הרפרנס +V : המתח מהLMZ 0.8 מתח המדידה -V : המתח מהFB אם השגיאה חיובית צריך לעלות את VOUT אם השגיאה שלילית צריך להוריד את VOUT

הפלט של מגבר השגיאה נכנס לטיימר, הוא מייצר אות פולסים עם תדירות קבועה 1MHz (נדלק ונסגר מליון פעמים בשנייה) אבל הפולסים בעלי D.C. שונה.
 ביציאה של המערכת (לפני VOUT) יש סליל וקבל CO, כאשר פולס מגיע: הסליל נטען וVOUT עולה
 כאשר הפולס מפסיק: הסליל ממשיך לספק זרם, הקבל אוגר מטען בזמן ON ופורק בזמן OFF
 בגלל שהסליל מתנגד לשינויים פתאומיים בזרם והקבל מתנגד לשינויים פתאומיים במתח השילוב שלהם גורם לכך שהיציאה לא קופצת בפולסים אלא יוצאת כמתח ממוצע חלק
 מטח הפידבק תמיד גם מושווה למתח פנימי של 0.92 וולט
 במקרה והוא עבר את מתח זה המערכת כמעין מנגנון הגנה מפסיקה את פעולתה.

אדמות

למערכת יש 2 סוגים של אדמות GND וEP, לשניהם יש ערך של 0V
 האדמה EP היא אדמה תרמית, GND היא אדמה אלקטרונית
 לעומת GND שמשמשת להארקה, EP משמשת לפיזור חום.

הסבר על פין ה-ENABLE:

כמו שאפשר לראות בשרטוט למטה, פין ה-ENABLE מחובר למחלק מתחבין VIN לאדמה יש 2 נגדים ובאמצע מתח שהולך ל-EN, המטרה של המחלק הזה זה שהחילוק של הנגדים יחליט איזה ערך של VIN יפעיל את המערכת, אם הערך של EN הוא 1.18 ומעלה המערכת תפעל.

$$V_{EN} = V_{in} \left[\frac{R_{ENB}}{R_{ENB} + R_{ENT}} \right]$$

$$1.18 = V_{in} \left[\frac{R_{ENB}}{R_{ENB} + R_{ENT}} \right]$$

$$V_{in} = 1.18 / \left[\frac{R_{ENB}}{R_{ENB} + R_{ENT}} \right]$$

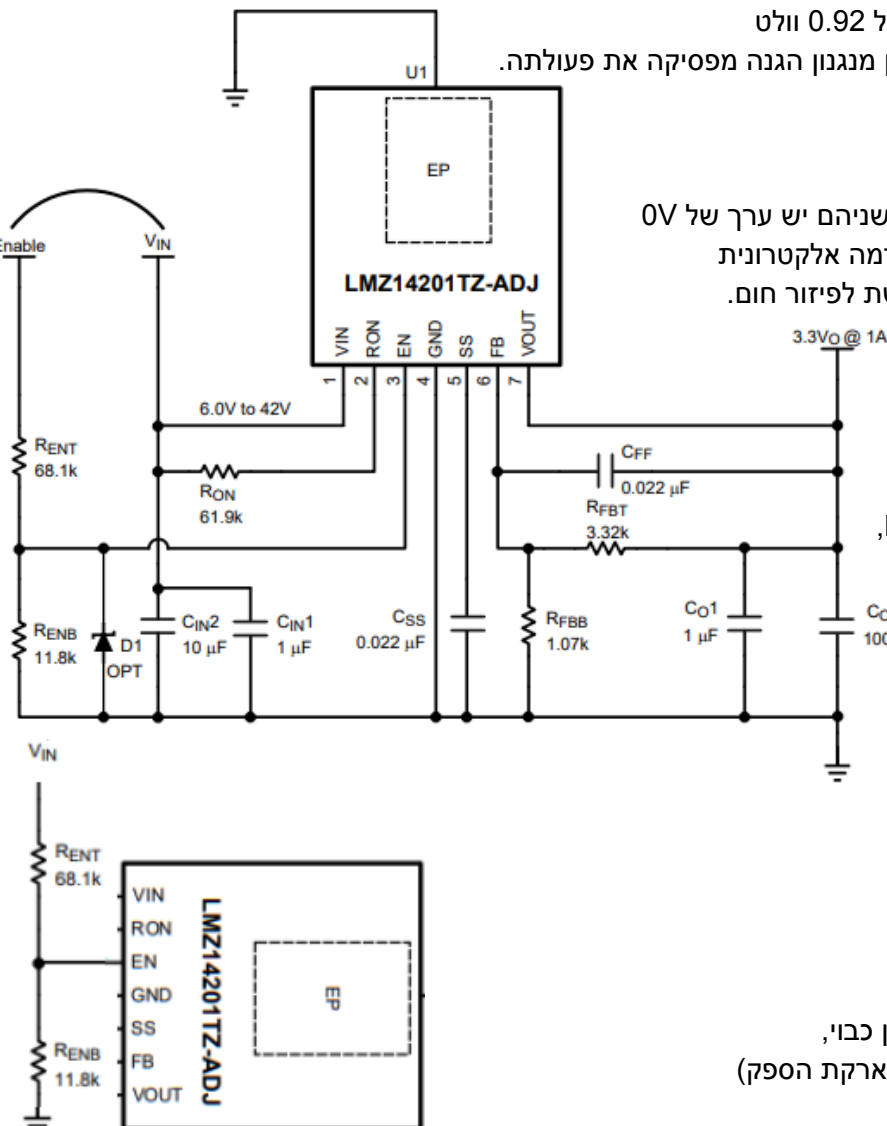
$$V_{in} = 1.18 * \left[1 + \left(R_{ENT} / R_{ENB} \right) \right]$$

כלומר הערך Vin שחישבתי הוא ערך המינימלי שיפעיל את המערכת בהתאם לנגדים שנבחר כמובן.
 במהלך זמן ה-OFF, כאשר ה-MOSFET העליון כבוי, זרם הסליל זורם דרך העומס, פין ה-PGND (הארקת הספק) וה-MOSFET הסנכרוני הפנימי.

אם הזרם הזה חורג מ-2 אמפר במוצע המשווה של מגבלת הזרם מונע את תחילת תקופת ה-ON TIME הבאה, מחזור ההחלפה הבא יתקיים רק אם פין ה-FEEDBACK יהיה פחות מ-0.8 וולט וזרם הסליל ירד מתחת ל-2 אמפר.

איך ה-DC משתנה?

בכל מחזור סוויטצ'ינג יש גל משולש בתדר של 1MHz מתח הגל עולה מ-0 ל- V_{MAX} בקצב ליניארי, ברגע שהמתח של הגל המשולש עולה מעל לאות השגיאה השבב מכבה את ה-MOSFET.



נגד RON:

נגד זה קובע את תדר הסוויצ'ינג וגם את סף ה-VIN להפעלה, בתוך LMZ יש זרם ייחוס קבוע של 1 מיקרו אמפר, RON מחובר חיצונית ל-VIN ובעזרת זרם הייחוס הקבל הפנימי של הטיימר נטען, עם מתח שנקרא VRON שהוא תואם לזרם הייחוס, ככל ש-VIN עולה, הקבל נטען מהר יותר ולכן זמן ה-ON יהיה קצר יותר, כשהמתח הזה עובר סף פנימי (1.18) הוא מאפשר ל-LMZ להמשיך לפעול הנגד מחובר פנימית לטיימר, בתוך הטיימר יש את הקבל שהזכרנו ומשווה שמחובר לקבל, המשווה בודק אם הקבל עבר מתח מסויים וקוטע אותו (כדי שיהיה ON TIME מסויים בלי תלות במשתנים חיצוניים) המשווה לוקח את הגל שיוצא עם קטיעה ומוציא ממנו גל ריבועי, 1 כשהוא עולה 0 כשהוא יורד. הטיימר מייצר גם גל בדיוק הפוך אליו שבעליה 0 בירידה 1, ושולח את 2 הגלים אל הטרנזיסטורים.

קבל Soft Start:

הוא קבל חיצוני שתפקידו למנוע מתחים גבוהים מדי בכניסת LMZ, ברגע שמתח הכניסה עובר את סף הערך הנדרש להפעלת LMZ הקיבול מתחיל להטען בהדרגה כך שמתח הפלט לא יעלה באופן מיידי אלא בצורה חלקה לאורך זמן.

הגנה תרמית:

טמפרטורת ה-Junction לא צריכה לחרוג מהמגבלות המרביות שלה, הגנה תרמית מיושמת באמצעות מעגל כיבוי תרמי פנימי שמופעל ב-165 מעלות צלזיוס בממוצע, מה שגורם למכשיר להכנס למצב המתנה במצב צריכת חשמל נמוכה; במצב זה ה-MOSFET הראשי נשאר כבוי מה שגורם ל-VO לרדת, בנוסף קבל ה-CSS (קבל soft start) משתחרר לאדמה, כאשר טמפרטורת ה-JUNCTION יורדת חזרה מתחת ל-145 מעלות פין ה-SS משוחרר ו-VO עולה בצורה חלקה.

זיהוי זרם אפס בסליל:

הזרם ב-MOSFET התחתון (הסנכרוני) מנוטר על ידי מעגל זיהוי זרם אפס בסליל שמונע את פעולתו של ה-MOSFET הסנכרוני כאשר הזרם שלו מגיע לאפס עד למועד ה-ON TIME הבא. מעגל זה מאפשר את מצב הפעולה DCM (מצב זרם לא רציף) שמשפר את היעילות בעומסים קלים.

מצבי פעולה של זרימה לא רציפה וזרימה רציפה:

בעומס קל המווסת פועל במצב DCM כאשר זרם העומס גבוה מנקודת הזרימה המקסימלית הוא פועל במצב CCM (מצב זרימה רציפה). כאשר המכשיר פועל ב-DCM מחזור ההחלפה מתחיל בזרם סליל של אפס אמפר עולה עד לערך שיא ולאחר מכן יורד חזרה לאפס לפני סוף ה-OFF TIME, במהלך התקופה בה זרם הסליל שווה לאפס כל זרם העומס מסופק על ידי קבל הפלט. תקופת ה-ON TIME הבאה מתחילה כאשר המתח על פין ה-FEEDBACK יורד מתחת לרפרנס הפנימי. תדר ההחלפה נמוך יותר במצב DCM והוא משתנה יותר עם זרם העומס בהשוואה ל-CCM. יעילות ההמרה במצב DCM נשמרת מכיוון שהפסדי הזרימה וההחלפה מצטמצמים עם העומס הקטן ותדר ההחלפה הנמוך ביותר.

הסבר על חלקים נוספים במעגל:

קבל CSS: מטרתו להתמלא בהדרגה בהפעלת הממיר כדי שמתח הפלט יעלה גם הוא בצורה הדרגתית.
נגדי RENT ו RENTB: נגדים אלו משמשים כמחלק מתח שמטרתו לקבוע את המתח VIN שבו פין EN יקבל מתח שהוא יוכל לעבוד בו

דיודת D1: במקרה של הפסקת מתח או היפוך זרימת המתח, הדיודה מונעת זרימת זרם הפוך ל EN PIN, טווח פעולת ה EN הוא בין 1.18 ל 6.5, 6.5 מ דיודת הזנר נפרצת.

הבדל בין ה MOSFETs: העליון משמש להעברת מתח מהכניסה ליציאה, לעומת זאת התחתון משמש להעברת זרם בין הכניסה לאדמה כדי לוודא שהזרם לא יעבור את 1A, כאשר טרנזיסטור אחד פתוח השני סגור.

קבל Cin: מטרתו לוודא שמתח הכניסה יהיה יציב ולא יהיו בו שינויים מיידיים מדי שיכולים לגרום נזק לפעולות המערכת

קבלים Cin1 ו Cin2: תפקידם לסנן רעשים ותנודות במתח הפלט, כשלמעגל יש רעשים בתדר גבוה, הקבלים מעבירים את הרעש דרך עצמם לאדמה.

קבלים CO1 ו CO2: תפקידם לייצב את המתח ביציאה, ולסנן רעשים.

מגבר שגיאה: למגבר יש משוב שלילי עם 2 הטרנזיסטורים, הוא מקבל בכניסה את המתח של FB וביציאה מוציא VOUT חדש

נגד RON: מחובר ל VIN ובעזרת זרם פנימי של 1uA נוצר מתח שנקרא VRON, מתח זה הולך אל קבל פנימי בטיימר, הקבל נטען במהירות מסויימת ומחובר למשווה שקוטע אותו במתח מסויים כדי לשלוט על ה TON, המשווה מוציא ביציאה 2 אותות ריבועיים הפוכים כאשר הקבל עולה ויורד.
רגיסטור ליניארי: תפקידו לווסת את המתח שמוזן למעגליים הפנימיים כך שיהיה יציב ללא תלות במתח כניסה

דיודת הרגיסטור: מטרתה להגן על המערכת ממתח כניסה הפוך או ממתח יתר

קבל CVCC: מטרתו לייצב את המתח שיוצא מהרגיסטור, להתמודד עם שינויים מעומס ולהפחית רעשים.

קבל הטרנזיסטור: הטרנזיסטור העליון דורש מתח גבוה יותר מ VIN ולכן קבל זה נמצא, תפקידו להטען ולשלוט מספיק מתח אל הטרנזיסטור לצורך הפעלתו

קבל CBST: הקבל מסנן תנודות מה PWM ומספק מתח יציב יותר ב-VOUT.

הסליל ביציאה: הסליל פועל כמסנן זרם ומפחית את התנודות בזרם, ובכך משפר את היציבות של זרם הפלט, כאשר הטרנזיסטור העליון סגור והתחתון פתוח, הסליל מפסיק לקבל מתח אז השדה אלקטרומגנטי שלו מתפרק, כתוצאה מכך (חוק לנץ) נוצר מתח הפוך בקצוות שלו אך הזרם נשאר באותו כיוון, הזרם ממשיך לזרום לעומס ולקבל.

קבל CFF: לשפר את תגובת המתח לשינויים פתאומיים מ VOUT כדי לייצב את המתח שנכנס לפין FB

נגדים RFBT ו RFBB: תפקידם לבצע מחלק מתח על פין FB כדי לקבוע מה ערכו של VOUT יהיה כיוון ש FB הוא מתח קבוע 0.8.

<u>טווחי הפעלה ופרמטרים בסיסיים:</u>	<u>מינימום</u>	<u>מקסימום</u>
VIN, RON to GND	-0.3V	43.5V
EN, FB, SS to GND	-0.3V	7V
טמפרטורת JUNCTION		150°C
טמפרטורת איחסון	-65°C	150°C

תנאי הפעלה מומלצים	מינימום	מקסימום
Vin	6v	42v
EN	0v	6.5v
טמפרטורת JUNCTION	-40°C	125°C
מידע תרמי	ערך	
התנגדות תרמית לאוויר (4 שכבות, 100 חורים, ללא זרימת אוויר)	19.3°C/W	
התנגדות תרמית לאוויר (2 שכבות, ללא זרימת אוויר)	21.5°C/W	
התנגדות תרמית למארז	1.9°C/W	

מספר פין	שם הפין	סוג	טווחים ועוד מידע
1	VIN	אספקת מתח	6 וולט עד 42 וולט. נדרש להוסיף קבל קלט חיצוני בין פין זה לבין הEP
2	RON	אנלוגי	(On Time Resistor) קובע את זמן ההפעלה (ON-time) של המערכת. בין 25 kΩ ל-124 kΩ.
3	EN	אנלוגי	סף ההפעלה בעלייה הוא 1.18 וולט (בלי התחשבות בסטיות) ההפרש בין הערכים שבהם המערכת משנה את מצבה הוא 90mV. רמת הקלט המרבית המומלצת היא 6.5 וולט.
4	GND	אדמה	יש לחבר אותה חיצונית ל-EP.
5	SS	אנלוגי	מקור זרם פנימי של 8μA טוען קבל חיצוני כדי לייצר את פונקציית ההתחלה הרכה. הצומת הזה מתפרק בזרם של 200μA במהלך תנאים של disable, זרם יתר (overcurrent), כיבוי תרמי ותנאי UVLO פנימיים (Undervoltage Lockout).
6	FB	אנלוגי	מחובר פנימית למשווים של regulation, overvoltage, ומעגלים קצרים. נקודת ההתייחסות 0.8 וולט בפין זה. יש לחבר את מחלק הנגדים של הפידבק בין הפלט לאדמה על מנת לקבוע את מתח הפלט.
7	VOUT	אספקת מתח	יש לחבר את קבל הפלט בין פין זה לבין הEP
-	EP	אדמה	יש לחבר אותו חשמלית לפין 4 מחוץ לחבילה.

תכנון המעגל ובחירת רכיבים:

בחירת סף הפעלה (UVLO) עם מחלק מתח ל-EN:

החלוקה של נגדי EN מאפשרת לבחור מתח קלט מתחתיו המערכת תכנס למצב UVLO מנגנון שמטרתו להגן על הממיר במקרה של מתח כניסה נמוך מדי. כלומר, אם VIN נופל מתחת לערך מסוים הממיר יפסיק לפלט מתח ויכנס למצב "כיבוי" כדי להימנע מפגיעות במעגלים הפנימיים. ובכך למנוע פריקה עמוקה של סוללה במערכות מונעות סוללה. היחס בין התנגדויות החלוקה נקבע לפי נוסחה שמתאימה למתח UVLO הרצוי

$$R_{ENT} / R_{ENB} = (V_{IN\ UVLO} / 1.18\ V) - 1$$

תכנון מתח הפלט עם מחלק נגדים ל-FB:

מתח היציאה מוגדר על ידי מחלק מתח של 2 נגדים המחוברים בין VO לאדמה. נקודת האמצע של המחלק מחוברת לקלט FB. המתח ב-FB מושווה למתח ייחוס פנימי של 0.8V. במהלך פעולה רגילה מחזור ON TIME מתחיל כאשר המתח על רגל FB יורד מתחת ל-0.8V מחזור ההפעלה של טרנזיסטור ההספק גורם לעליית מתח היציאה והמתח ב-FB עולה מעל 0.8V כל עוד המתח ב-FB גבוה מ-0.8V לא יופעלו מחזורי ON TIME. מתח היציאה המיוצב מחושב כך:

$$V_O = 0.8V \times (1 + R_{FBT} / R_{FBB}) \Leftrightarrow R_{FBT} / R_{FBB} = (V_O / 0.8V) - 1$$

יש לבחור ערכים לנגדים בטווח של 1.0 קילואוהם עד 10.0 קילואוהם. כאשר $V_O = 0.8V$ ניתן לחבר את רגל FB ישירות למתח היציאה כל עוד מחובר נגד עומס שמספק זרם גדול 20uA פעולת הממיר דורשת עומס מינימלי זה כדי ליצור זרם גלים קטן בסליל ולשמור על ויסות נכון גם כשאין עומס. קבל Feed-Forward ממוקם במקביל ל-RFBT כדי לשפר את תגובת המתח לשינויים פתאומיים בעומס. ערכו נקבע לרוב בניסוי, תוך כדי החלפת עומס בין מצבי DCM ל-CCM וכולל לתגובה מהירה ולריפוד מינימלי.

תכנון זמן התחלה (Soft-Start) עם קבל CSS:

התחלה רכה מתוכננת מאפשרת לרגולטור לעלות בהדרגה לנקודת הפעולה היציבה שלו לאחר ההפעלה, ובכך להפחית את זרם ההתחלה מהכניסה ולהאט את קצב העלייה של מתח היציאה כדי למנוע חריגה זמנית מעל מתח היעד משך זמן ההתחלה הרכה ניתן לפי הנוסחה:

$$t_{SS} = V_{REF} \times C_{SS} / I_{SS} = 0.8V \times C_{SS} / 8\ \mu A \Leftrightarrow C_{SS} = t_{SS} \times 8\ \mu A / 0.8\ V$$

שימוש בקבל של 0.022uF ייתן זמן התחלה רכה של כ-2.2ms – זהו הערך המינימלי המומלץ. כאשר מתח קלט ה-SS עובר את 0.8V, מתחילה פעולת הוויסות. הקבל ממשיך להיטען עד לכ-3.8V, אך מתחים בטווח שבין 0.8V ל-3.8V אינם משפיעים על הפעולה.

בחירת קבל יציאה (CO):

אף אחד מערכי הקיבול הדרושים ביציאה (CO) אינו כלול בתוך המודול. לכל הפחות, קבל היציאה חייב לעמוד בדרישת זרם הריפל המינימלית הגרועה ביותר של $ILR\ P-P \times 0.5$, (זרם הריפל הוא רכיב משתנה בזרם הכולל שעובר דרך רכיב, לרוב קבל או סליל, במעגל הספק). מעבר לכך, הוספת קיבול תפחית את הריפל ביציאה, כל עוד ההתנגדות הפנימית האופיינית (ESR) נמוכה מספיק כדי לאפשר זאת. ערך מינימלי של $10\mu F$ נדרש בדרך כלל. אם מנסים לפעול עם ערך מינימלי זה, יש צורך בניסוי. המשוואה הבאה מספקת קירוב ראשוני טוב עבור חישוב קבל היציאה (CO) לצורך דרישות של שינויים בתאומיים בעומס:

$$CO \geq I_{STEP} \times V_{FB} \times L \times V_{IN} / [4 \times V_O \times (V_{IN} - V_O) \times V_{OUT-TRAN}]$$

פתרון:

$$CO \geq 1\ A \times 0.8\ V \times 10\ \mu H \times 24\ V / [4 \times 3.3\ V \times (24\ V - 3.3\ V) \times 33\ mV] \geq 21.3\ \mu F$$

לוחות הדגמה והערכה של LMZ14201 כוללים קבל יציאה של $100\mu F$ מסוג X5R בקיבול של $6.3V$. קיימים מיקומים נוספים עבור קבלי יציאה נוספים.

בחירת קבל כניסה (CIN):

מודול ה-LMZ14201 כולל קבל קרמיקה פנימי בקיבול של $0.47\mu F$ בכניסה. עם זאת, דרוש קיבול כניסה נוסף מחוץ למודול כדי להתמודד עם זרם הריפל בכניסה של היישום. קיבול הכניסה הזה צריך להיות קרוב מאוד למודול. הבחירה של קבל הכניסה מתבצעת בדרך כלל לפי דרישות זרם הריפל בכניסה ולא לפי ערך הקיבול בלבד. דרישת זרם הריפל המקסימלית נקבעת על פי משוואה זו:

$$I(CIN(RMS)) \approx 1 / 2 \times I_O \times \sqrt{(D / 1-D)} \quad \text{תחום הצבה: } D \approx V_O / V_{IN}$$

(לשם השוואה, זרם הריפל המקסימלי יתרחש כאשר המודול מופעל בעומס מלא ו- $V_{IN} = 2 \times V_O$). הקיבול המינימלי המומלץ לכניסה הוא $10\mu F$ מסוג קרמיקה X7R עם דירוג מתח לפחות 25% גבוה יותר מהמתח המקסימלי בכניסה של היישום. מומלץ גם לשים לב לירידות מתח ולירידות טמפרטורה של הקבל הנבחר. יש לשים לב כי דירוג זרם הריפל של קבלי קרמיקה עשוי לא להיכלל בגליון הנתונים של הקבל, ולכן ייתכן שיהיה צורך לפנות ליצרן הקבלים לקבלת דירוג זה. אם עיצוב המערכת דורש לשמור על ערך מינימלי מסוים של מתח ריפל בכניסה (ΔV_{IN}), ניתן להשתמש במשוואה זו:

$$CIN \geq I_O \times D \times (1-D) / f_{SW-CCM} \times \Delta V_{IN}$$

אם ΔV_{IN} הוא 1% מ- V_{IN} עבור יישום עם קלט 24 וולט ופלט 3.3 וולט, זה שווה ל-240mV או $f_{SW} = 400$ קילו-הרץ חישוב:

$$CIN \geq 1\ A \times (3.3\ V / 24\ V) \times (1 - 3.3\ V / 24\ V) / (400000 \times 0.240\ V) \geq 0.9\ \mu F$$

יתכן ויהיה צורך בקיבול נוסף עם ESR גבוה יותר כדי לדכא אפקטים תהודתיים שנוצרים מהקיבול בכניסה ומהאינדוקטיביות הטורית הטבעית של קווי האספקה.

בחירת התנגדות RON לקביעת תדר ההפעלה:

במרבית התכנונים מתחילים עם תדר מיתוג רצוי. לצורך זה ניתן להשתמש במשוואה הבאה:

$$f_{SW}(CCM) \approx VO / (1.3 \times 10^{-10} \times RON) \Leftrightarrow RON \approx VO / (1.3 \times 10^{-10} \times f_{SW}(CCM))$$

יש לבחור את RON ואת $f_{SW}(CCM)$ תוך התחשבות במגבלות זמני ההפעלה והכיבוי (ON/OFF) של מעגל הבקרה מסוג COT/

זמן ההפעלה של טיימר ה-LMZ14201 נקבע לפי ערך הנגד RON ומתח הכניסה VIN, והוא מחושב כך:

$$t_{ON} = (1.3 \times 10^{-10} \times RON) / VIN$$

הקשר ההפוך בין t_{ON} לבין VIN מביא לכך שהתדר נשאר כמעט קבוע גם כאשר VIN משתנה.

יש לבחור את RON כך שזמן ההפעלה (t_{ON}) במתח הכניסה המרבי $VIN(MAX)$ יהיה גדול מ-150

ננו-שניות. לטיימר ההפעלה יש מגביל שמבטיח זמן מינימלי של 150ns. מגבלה זו קובעת את תדר העבודה המרבי, כפי שמתואר במשוואה זו:

$$f_{SW}(MAX) = VO / (VIN(MAX) \times 150 \text{ ns})$$

משוואה זו יכולה לשמש לבחירת ערך RON כאשר רוצים תדר עבודה מסוים, כל עוד מכבדים את מגבלת זמן ההפעלה המינימלי של 150ns.

המגבלה על ערך RON מחושבת כך:

$$RON \geq VIN(MAX) \times 150 \text{ nsec} / (1.3 \times 10^{-10})$$

בחירת מצב DCM ומצב CCM:

תדר פעולה במצב הולכה לא רציפה (DCM) ניתן לחשב כך:

$$f_{SW}(DCM) \approx VO \times (VIN-1) \times 10 \mu H \times 1.18 \times 10^{20} \times IO / (VIN-VO) \times RON \times 2$$

במצב הולכה רציפה (CCM), הזרם עובר דרך הסליל לאורך כל מחזור המיתוג ואף פעם לא יורד לאפס במהלך ה-OFF-time. תדר המיתוג נשאר כמעט קבוע למרות שינויים בזרם העומס ומתח הקו.

הנוסחה המקורבת לקביעת גבול המעבר בין DCM ל-CCM היא:

$$IDCB \approx VO \times (VIN-VO) / (2 \times 10 \mu H \times f_{SW}(CCM) \times VIN)$$

הסליל הפנימי למודול הוא 10 μH . ערך זה נבחר כפשרה טובה בין יישומים במתח כניסה נמוך לגבוה. הפרמטר העיקרי שהסליל משפיע עליו הוא משרעת זרם הגל (ripple current) בסליל (ILR).

את ILR ניתן לחשב כך:

$$ILR \text{ P-P} = VO \times (VIN-VO) / (10 \mu H \times f_{SW} \times VIN)$$

כאשר:

- VIN הוא המתח המקסימלי בכניסה
- f_{SW} נקבע כבר לפני

אם זרם היציאה IO נקבע בהנחה ש-IL = IO (זרם הסליל), ניתן לחשב את השיאים הגבוהים והנמוכים של ILR.

יש לשים לב כי השיא הנמוך של ILR חייב להיות חיובי אם נדרשת פעולה במצב CCM.

המלצות לספק כוח

התקן LMZ14201 מיועד לפעול במתח כניסה בטווח של 4.5V עד 42V. ספק הכוח לכניסה צריך להיות מוסדר היטב, ולהיות מסוגל להתמודד עם זרם כניסה מקסימלי תוך שמירה על מתח יציב. ההתנגדות בנתיב ספק הכוח לכניסה צריכה להיות נמוכה מספיק כך שטרנזיאנטים של זרם כניסה לא יגרמו לירידת מתח משמעותית במתח ההזנה של ה-LMZ14201, אשר עלולה לגרום להפעלת שגיאת UVLO שגויה ואיפוס המערכת. אם ספק הכוח נמצא במרחק של יותר מכמה סנטימטרים מה-LMZ14201, יתכן ויש להוסיף קבל נפח נוסף בנוסף לקבלים הקרמיים לעקיפת רעש. כמות קיבול הנפח (bulk capacitance) אינה קריטית, אך קבל אלקטרוליטי של $47\mu F$ או $100\mu F$ הוא בחירה טיפוסית.

שיקולי פריסת מעגל (Layout)

הנחיות לפריסת מעגל:

פריסת מעגל מודפס (PCB) היא חלק חשוב בתכנון ממיר DC-DC. פריסה לא טובה עלולה לפגוע בביצועי הממיר ובמעגלים הסמוכים, כתוצאה מהפרעות אלקטרומגנטיות (EMI), קפיצות מתח בהארקה (ground bounce) ונפילות מתח בתוואי המוליכים (traces). תופעות אלו עלולות לשלוח אותות שגויים אל הממיר, ולגרום לאיבוד ויסות או לאי-יציבות. ניתן להשיג פריסה טובה על ידי יישום מספר כללים פשוטים:

מזעור שטח לולאות זרם ממותגות

מבחינת הפחתת EMI, חשוב מאוד למזער את המסלולים בעלי קצב שינוי זרם גבוה (di/dt) בפריסת המעגל. לולאות זרם גבוה שאינן חופפות מכילות תוכן di/dt גבוה, אשר יגרום לרעש בתדר גבוה על פין היציאה אם קבל הכניסה (CIN1) ממוקם רחוק מה-LMZ14201. לכן, יש למקם את CIN1 קרוב ככל האפשר לפין VIN ולפד ההארקה (GND exposed pad) של ה-LMZ14201. כך ניתן למזער את שטח ה- di/dt הגבוה ולהפחית קרינה אלקטרומגנטית (EMI). בנוסף, הארקה של קבלי הכניסה והיציאה צריכה להיעשות על-ידי מישור מקומי בשכבה העליונה שמתחבר לפד הארקה (EP).

נקודת הארקה אחת (Single Point Ground)

חיבורי ההארקה של מעגל המשוב, ההפעלה הרכה (soft-start) והרכיב enable צריכים להיות מנותבים לפין ההארקה (GND) של הרכיב. זה מונע זרמי מיתוג או זרמי עומס מלעבור במסלולי ההארקה האנלוגיים. אם לא מטפלים בכך כראוי, תיגרם ירידה באיכות ויסות העומס או רעש לא יציב ביציאת המתח. יש ליצור חיבור הארקה מנקודה אחת – מפין 4 אל הפד החשוף (EP).

מזעור אורך התוואי אל פין המשוב (FB)

שני נגדי המשוב, RFBT ו-RFBB, וכן הקבל העוקף CFF, צריכים להיות ממוקמים קרוב לפין FB. כיוון שפין FB הוא צומת בעל עכבה גבוהה, יש לשמור על שטח נחושת קטן ככל האפשר. התוואים של RFBT, RFBB ו-CFF צריכים להיות מנותבים הרחק מגוף רכיב ה-LMZ14201 כדי להפחית רעש.

חיבורי כניסה ויציאה רחבים ככל האפשר

יש להרחיב את תוואי הכניסה והיציאה של הממיר כדי להפחית נפילות מתח ולשפר את היעילות. לצורך דיוק מתח גבוה יותר בעומס, יש לבצע מדידת מתח משוב (sense) נפרדת מהעומס, כדי לתקן נפילות מתח לאורך התוואים ולשמור על דיוק מרבי ביציאה.

פיזור חום נאות לרכיב

כדי להבטיח פיזור חום יעיל מהרכיב, יש לחבר את הפד החשוף (EP) למישור ההארקה בשכבתו התחתונה של לוח המעגל (PCB) באמצעות מערך של מעברי חום (thermal vias). במידה והלוח כולל מספר שכבות נחושת, ניתן לחבר את המעברים גם למישורי הארקה פנימיים נוספים כדי לשפר את פיזור החום. לתוצאה מיטבית, מומלץ להשתמש במערך של 6 על 6 מעברים (סה"כ 36), בקוטר מינימלי של 8 mils, וברוח של 59 mils (1.5mm) ביניהם. יש להקצות שטח נחושת מספק לצורך פיזור חום, על מנת לשמור על טמפרטורת צומת מתחת ל-125°C.

הנחיות להרכבת מודול הספק בהלחמת SMT

- ההמלצות הבאות מיועדות להרכבת מודול סטנדרטי בטכנולוגיית הלחמת פני שטח (SMT):
 - **תבנית נחיתה (Land Pattern)** – יש לעקוב אחר תבנית הנחיתה של ה-PCB, עם פדים מוגדרים ע"י מסיכת הלחמה או ללא הגדרה כזו.
 - **פתחים בתבנית השבלונה (Stencil Aperture):**
 - עבור הפד החשוף של הדיי (DAP – Die Attach Pad), יש להתאים את פתיחת השבלונה כך שתכסה כ-80% מתבנית הנחיתה שעל הלוח.
 - עבור שאר פני הקלט/פלט (I/O), יש לשמור על יחס של 1:1 בין פתיחת השבלונה לבין תבנית הנחיתה.
 - **משחת הלחמה** – יש להשתמש בסגסוגת סטנדרטית מסוג SAC, למשל SAC305, מסוג 3 או גבוה יותר.
 - **עובי השבלונה (Stencil Thickness)** – בין 0.125 מ"מ ל-0.15 מ"מ.
 - **תהליך Reflow (המסה והלחמה בתנור)** – יש לפעול לפי המלצות ספק משחת ההלחמה, ולבצע אופטימיזציה בהתאם לגודל הלוח ולצפיפות הרכיבים.
 - למידע נוסף בנוגע ל-Reflow יש לעיין במסמך **AN SNA214**.
 - **מספר מרבי של תהליכי Reflow מותר: אחד בלבד.**

פיזור הספק ודרישות תרמיות של לוח המעגל

במקרה תכנוני של:

$$V_{IN} = 24V, V_O = 3.3V, I_O = 1A$$

טמפרטורת סביבה מרבית $T_{AMB}(MAX) = 85^{\circ}C$, וטמפרטורת צומת $T_{JUNCTION} = 125^{\circ}C$,

יש לוודא כי ההתנגדות התרמית מהמארז (case) אל הסביבה (ambient) תהיה קטנה מ:

$$R_{\theta CA} < (T_{J-MAX} - T_{AMB}(MAX)) / P_{IC-LOSS} - R_{\theta JC}$$

בהינתן שההתנגדות התרמית הטיפוסית מהצומת למארז היא $1.9^{\circ}C/W$,

יש להשתמש בעקומות פיזור ההספק בטמפרטורת כדי להעריך את אובדן ההספק של הרכיב ($P_{IC-LOSS}$) עבור היישום.

במקרה זה, ההפסד הוא $0.52W$.

החישוב:

$$R_{\theta CA} = 125 - 85 - 0.52W - 1.9 = 75^{\circ}C/W$$

כדי להגיע ל- $R_{\theta CA} = 75^{\circ}C/W$, לוח המעגל חייב לפזר חום בצורה יעילה.

כאשר אין זרימת אוויר ואין קירור חיצוני, הערכה טובה לשטח הלוח הנדרש (שיכוסה בנחושת בעובי 1oz)

בשכבות העליונה והתחתונה היא:

$$Board Area_{cm^2} = 500^{\circ}C \times cm^2 / W / R_{\theta JC}$$

לכן, נדרש שטח של כ-6 ס"מ² של נחושת בעובי 1oz בשכבה העליונה והתחתונה של הלוח.

הגדלת שטח זה תסייע בהפחתת טמפרטורת הצומת בהתאם.

יש לחבר את גוף קירור הנחושת של ה-PCB לפד החשוף של הרכיב.

כ-36 מעברי חום (thermal vias) בקוטר 8 mils וברוח של 1.5 מ"מ (59 mils) נדרשים לחיבור בין שכבות הנחושת העליונה והתחתונה.

תגובה לשינוי במתח הכניסה:

ה-LMZ כולל לולאת בקרה שלילית (feedback loop) שמשווה את מתח היציאה לרפרנס פנימי (0.8V). כשמתח הכניסה משתנה:

- אם מתח הכניסה עולה, הממיר יקטין את זמן ההולכה של הטרנזיסטור כדי לא לעלות את מתח היציאה.
- אם מתח הכניסה יורד, הממיר יאריך את זמן ההולכה של הטרנזיסטור (אם אפשר), כדי לשמור על מתח יציאה קבוע.

לדוגמה: אם $V_{IN} = 6V$ ו- V_{out} מוגדר ל-5V — אין הרבה מקום לויסות. אם V_{IN} יורד

ל-5.5V, המתח ביציאה יתחיל גם הוא לרדת כי פשוט אין מספיק מרווח.

ההבדל בין EP ל-GND:

EP הוא אדמה טרמית, לעומת זאת GND זו אדמה אלקטרונית, לשניהם יש אותו ערך של 0 וולט אבל EP חיוני לפיזור חום לעומת GND שאחראי על הארקה חשמלית

פרוטוקול ETHERNET

פרוטוקול זה הוא פרוטוקול לתקשורת מחשבים שמשמש בעיקר לרשתות מקומיות (LAN (Local Area Network

יתרונות של ETHERNET

- **יעילות ומהירות** - מאפשר תקשורת מהירה מאוד בין מכשירים, עם מהירויות שונות מ-10Mbps ועד ל-400Gbps זה אומר שניתן להעביר כמויות עצומות של מידע ברשת בזמן קצר.
- **אמינות** - Ethernet כולל מנגנונים מובנים לזיהוי שגיאות בתקשורת ולתיקון שלהם כלומר, הפרוטוקול נותן מענה למקרים שבהם יש הפרעות או בעיות בתקשורת, ומוודא שהמידע מגיע ליעדו בשלמותו ובאופן מדויק.
- **קלות התקנה ותחזוקה** - בניגוד לאופציות תקשורת אחרות, Ethernet קל מאוד להתקנה, לחיבור ולניהול. למעשה, רוב מערכות ההפעלה והתקני הרשת כבר תומכים בו "מהקופסה", ללא צורך בהגדרות או התאמות מסובכות. זה מפחית מאוד את העלויות והמורכבות של הקמה ותחזוקה של רשתות.
- **תאימות** - מכיוון ש-Ethernet הוא תקן פתוח ומוסכם, קל מאוד לחבר אליו מגוון עצום של מכשירים והתקנים. כך שאין צורך להתעסק במתאמים או פתרונות ספציפיים לכל מכשיר.
- **גמישות** - Ethernet תומך במגוון רחב של סוגי כבלים, מהירויות ומרחקי תקשורת. הוא יכול לפעול על כבל נחושת פשוט או על סיבים אופטיים מתקדמים, במהירות של 10Mbps או 100Gbps, ועל מרחקים של כמה מטרים עד קילומטרים רבים. כך שהוא מתאים לכל צורך ותרחיש, מרשת ביתית פשוטה ועד לרשתות עצומות של חברות וארגונים.

מסגרות (FRAMES)

Frame הוא למעשה "חבילת" המידע הבסיסית שעוברת ברשת Ethernet. זוהי היחידה הקטנה ביותר של נתונים שיכולה להישלח בפני עצמה. מסגרת מכילה את הפרטים הבאים:

- **מבוא (preamble):** זהו רצף של 8 בייטים שמטרתו לסנכרן את כל ההתקנים ברשת לקראת השידור של הנתונים עצמם.
- **כתובת MAC יעדית:** זהו שדה של 6 בייטים (48 ביטים) שמכיל את כתובת ה-MAC של ההתקן שאליו מיועדת המסגרת. זה מאפשר לרשת לדעת לאן בדיוק צריך להעביר את החבילה.
- **כתובת MAC מקור:** גם זה שדה של 6 בייטים, אבל הוא מכיל את כתובת ה-MAC של ההתקן ששלח את המסגרת. זה חשוב כדי שההתקן המקבל ידע למי לשלוח תשובה במקרה הצורך.
- **סוג / אורך:** שדה של 2 בייטים שמציין את סוג הפרוטוקול של השכבות העליונות ש"עוטף" את הנתונים במסגרת (כמו IP), או לחילופין, את אורך שדה הנתונים (במקרה של Ethernet II).
- **נתונים:** זהו החלק העיקרי של המסגרת, שמכיל את המידע שאנחנו רוצים להעביר. שדה הנתונים יכול להכיל עד 1500 בייטים של מידע (אם כי יש מקרים מיוחדים עם מסגרות גדולות יותר).
- **רצף בדיקת מסגרת:** שדה של 4 בייטים שמכיל ערך מתמטי (נקרא CRC – Cyclic Redundancy Check) שמחושב על בסיס כל שאר תוכן המסגרת. מטרתו לאפשר לצד המקבל לזהות אם הייתה שגיאה כלשהי בהעברת הנתונים, ולבקש שידור מחדש במקרה הצורך.

כל מסגרת כזו נשלחת על גבי הכבל ברשת ה-Ethernet. ההתקנים המחוברים לרשת בודקים כל מסגרת שעוברת ומחליטים אם היא רלוונטית להם, על סמך כתובת ה-MAC היעד. אם הכתובת תואמת לשלהם (או שהיא כתובת Broadcast, שמיועדת לכולם), הם יקבלו את המסגרת ויעבדו אותה. אחרת, הם פשוט יתעלמו ממנה ויתנו לה לעבור הלאה.

שיטת CSMA/CD למניעת שגיאות

בגרסאות ישנות של ETHERNET, השתמשו בשיטת מניעת שגיאות בשם CSMA/CD, אומנם היום ב-ETHERNET מבוססות על SWITCHES במקום HUBS, כלומר כל מכשיר מחובר ל-Switch בקו ייעודי. אין "התנגשויות" כי כל חיבור הוא full-duplex – כלומר, אפשר לשדר ולקבל מידע בו-זמנית. לא קיים "shared medium", אז אין צורך לגלות התנגשויות.

איך CSMA/CD עובד?

- **האזנה לפני שידור** - לפני שההתקן שולח משהו, הוא מקשיב לרשת כדי לראות אם מתבצע שידור אחר באותו הזמן. אם הערוץ "שקט", הוא יתחיל לשדר. אם מישו אחר משדר, הוא ימתין.
- **זיהוי התנגשויות** - בזמן השידור, ההתקן ממשיך "להקשיב" לרשת. אם הוא מזהה שידור אחר בו-זמנית (מצב של "התנגשות"), הוא מפסיק מיד את השידור שלו.
- **ניסיון שידור חוזר** - לאחר התנגשות, ההתקן מחכה פרק זמן אקראי (כדי להימנע מהתנגשות חוזרת), ואז מנסה לשדר שוב מההתחלה.

כתובות MAC

MAC היא כתובת פיזית, ייחודית לכל התקן ברשת. היא צרובה על כרטיס הרשת (או ממשק הרשת) של ההתקן, ולרוב לא ניתן לשנות אותה. כתובת MAC מורכבת מ-6 בייטים ונראית בדרך כלל כסדרה של 12 ספרות הקסדצימליות, מופרדות בקווים או נקודותיים. למשל: 00:11:22:33:44:55. חשוב להבין שכתובת ה-MAC היא כתובת קבועה שמזהה את ההתקן עצמו, להבדיל מכתובת ה-IP שהיא הכתובת הלוגית של ההתקן ברשת, ויכולה להשתנות. כתובת ה-MAC משמשת לתקשורת ברמת השכבה הפיזית והקו, בעוד IP משמש ברמת שכבת הרשת. מכיוון שכל יצרן מקבל טווח כתובות MAC משלו (כאשר 3 הבייטים הראשונים נקבעים על ידי ארגון התקנים IEEE), אפשר תיאורטית להבטיח שלא יהיו שתי כתובות זהות בעולם (אף שבפועל זה לא תמיד המצב, עקב זיופים או שגיאות אנוש).

סוגי כבלים ומחברים

כבל קואקסיאלי - זהו הכבל שהיה נפוץ מאוד בימים הראשונים של Ethernet. הוא עבה יחסית, עם ליבה מוליכה במרכז וציפוי מתכתי מסביב. היתרון שלו הוא עמידות רבה להפרעות ולנזקים. החיסרון – שהוא קשיח, יקר ומסורבל יותר לעבודה. נעשה בו שימוש בעבר במה שנקרא רשתות "אסטי" (Bus topology), שבהן כל ההתקנים מחוברים לאותו כבל יחיד. כיום הוא פחות נפוץ בשימוש ב-Ethernet.

זוגות שזורים twisted pairs - סוג הכבלים הנפוץ ביותר בימינו ל-Ethernet. הם מכילים ארבעה זוגות של כבלי נחושת, שכל אחד מהם שזור סביב עצמו כדי להפחית הפרעות. יש שני סוגים עיקריים UTP Unshielded Twisted Pair, ללא מיגון מתכתי חיצוני, ו-STP Shielded Twisted Pair, עם מיגון כזה. כבלים אלו דקים וגמישים יותר, וקלים יותר לניתוב והתקנה. הם משולבים במחברי RJ45 בקצוות.

סוגי זוגות שזורים:

CAT5: תומך ב-100Mbps עד 100 מטר. משמש בעיקר ברשתות ישנות יותר.
 CAT5e: משופר ל-1Gbps עד 100 מטר. הכי נפוץ בבתיים ובעסקים קטנים.
 CAT6: ממוטב עוד יותר, תומך ב-10Gbps עד 55 מטר וב-1Gbps עד 100 מטר.
 CAT6A, CAT7, CAT7A: תומכים במהירויות גבוהות אף יותר, אך פחות נפוצים כרגע.

סיבים אופטיים - אלה כבלים המורכבים ממספר סיבים זכוכית דקים ביותר, שמעבירים אותות אור במקום חשמל. הם מציעים מהירויות גבוהות מאוד (עד מאות ג'יגה-ביט לשנייה) על פני מרחקים גדולים (עד קילומטרים). בנוסף, הם חסינים מאוד מהפרעות אלקטרומגנטיות. העלות שלהם גבוהה יותר, והם דורשים ציוד קצה ייעודי.

ETHERNET במודל 7 השכבות OSI

רוב הפרוטוקולים והטכנולוגיות ברשת פועלים בשכבה אחת או יותר במודל הזה. Ethernet, בפרט, פועל

בעיקר בשתי השכבות התחתונות: הפיזית ושכבת הקו:

בשכבה הפיזית: ETHERNET מגדיר את סוגי הכבלים

והמחברים, את השיטות לקידוד הסיביות לסימני מתח או

אור, ואת כל החוקים הפיזיים של השידור.

בשכבת הקו: Ethernet מטפל בכל מה שקשור למסגרות

איך הן מובנות, איך ההתקנים ניגשים לערוץ המשותף

(CSMA/CD בעבר, מתגים כיום), איך מוודאים שהמסגרות

הגיעו בשלמותן ובלי טעויות, וכו'.

7 layers of OSI model

• Application	End User Layer
• Presentation	Syntax layer
• Session	Synch and send to Port
• Transport	End to End connection
• Network	Packets
• Data Link	Frames
• Physical	Physical structure

IoT Dunia .com

מתאם רשת NIC Network Interface Card - זהו כרטיס או שבב שקיים בכל מחשב או התקן קצה (כמו מדפסת רשת) ומשמש לחיבור פיזי לרשת. הוא מבצע את כל הפעולות של קידוד ופענוח האותות, מימוש CSMA/CD (בעבר), טיפול במסגרות ועוד. כל מתאם כולל כתובת MAC ייחודית משלו, שמזהה את ההתקן הזה ברשת.

מרכזת Hub - זהו התקן פשוט יחסית, שפועל כמעין "מפצל" לאותות. כל מה שהוא עושה הוא לקחת את מה שהוא מקבל בכניסה אחת ולהעביר את זה לכל שאר ההתקנים המחוברים אליו. הוא לא מנתח את המסגרות או מנתב אותן – הוא פשוט מעתיק ומשכפל. זה אומר שמבחינת מרכזת, כל הרשת היא למעשה כבל אחד גדול, עם כל החסרונות הנובעים מכך (התנגשויות, עומס מיותר וכו'). כיום מרכזות פחות נפוצות, והן הוחלפו ברובן במתגים.

מתג Switch - זהו כבר התקן חכם יותר, שיודע להבין את משמעות המסגרות שהוא מקבל ולנתב אותן רק ליעד הרלוונטי. הוא עושה את זה על ידי למידה של טבלת כתובות MAC – מי מחובר לכל כניסה שלו. כשמגיעה מסגרת, הוא בודק את כתובת ה-MAC של היעד, מסתכל בטבלה, ומשגר את המסגרת רק לכניסה הספציפית שבה נמצא אותו התקן. זה מייצר הפרדה לוגית של הרשת ופותר חלק ניכר מבעיות העומס וההתנגשויות של העבר.

נתב Router - התקן זה פועל כבר בשכבה הגבוהה יותר (שכבת הרשת במודל OSI). הוא מחבר בין רשתות Ethernet שונות, וקובע את הנתיבים להעברת חבילות נתונים בין רשתות אלו, בהסתמך על פרוטוקולי ניתוב כמו IP. נתבים יודעים להכריע לאן לשלוח נתונים שמיועדים ליעדים מחוץ לרשת המקומית, ויוצרים בפועל את האינטרנט.

שער Gateway - שער הוא התקן שיכול לחבר בין רשת Ethernet לבין רשת אחרת לגמרי, מסוג שונה. הוא מתרגם בין הפרוטוקולים של הרשתות השונות ויודע לארוז ולפרוק מחדש את הנתונים לפורמט המתאים. למשל, שער יכול לחבר בין רשת Ethernet במשרד לבין רשת הסלולר של הטלפונים הניידים.

אבטחה ב-Ethernet

טכנולוגיות שמספקות שירותי אבטחה ל-Ethernet:

- **SSL/TLS** - פרוטוקולי הצפנה שמאפשרים תקשורת מאובטחת בין לקוח לשרת, עם זיהוי הגורמים המעורבים. משמשים באתרי אינטרנט מאובטחים (HTTPS), דוא"ל מאובטח ועוד.
- **IPSec** - פרוטוקול להצפנה ואימות ברמת חבילות ה-IP. משמש בעיקר ליצירת רשתות פרטיות וירטואליות (VPN) מאובטחות, על גבי תשתיות ציבוריות כמו האינטרנט.
- **802.1X** - תקן אבטחה ברמת הגישה לרשת, המאפשר אימות משתמשים או התקנים לפני שהם מורשים להתחבר. זה עוזר להתמודד עם איומים כמו התקנים לא מורשים שמתחברים לרשת הארגונית.

תקנים של Ethernet

IEEE 802.3 - זהו התקן הבסיסי של Ethernet, שפורסם לראשונה ב-1983 על ידי ארגון התקינה IEEE. הוא מגדיר את העקרונות של CSMA/CD, את מבנה המסגרות, ואת הדרישות הפיזיות לכבלים ולממשקים. כל שאר התקנים של Ethernet הם למעשה "וריאציות" או הרחבות של 802.3.

10BaseT - זהו התקן שפורסם ב-1990 ומגדיר Ethernet של 10Mbps על גבי כבלי זוגות שזורים (Twisted Pair) מסוג CAT3 או CAT5, עם מחברי RJ45. זה היה למעשה התקן שהפך את Ethernet לנפוץ ונגיש יותר.

100BaseTX - זהו התקן Fast Ethernet שהוצג ב-1995, ומספק 100Mbps על גבי זוגות שזורים באיכות CAT5 ומעלה. הוא משתמש בשיטת קידוד מתקדמת יותר (4B5B ו-MLT-3) כדי להשיג מהירויות גבוהות יותר על אותם כבלים.

1000BaseT - זהו תקן ה-Gigabit Ethernet משנת 1999, המיועד לזוגות שזורים מסוג CAT5e ומעלה. הוא משתמש בכל ארבעת הזוגות בכבל (במקום שניים ב-100Mbps) ובשיטות איתות מתקדמות כדי להשיג מהירות של 1Gbps.

10GBASE-T, 40GBASE-T, 100GBASE-T - אלו התקנים המגדירים Ethernet של 10, 40 ו-100 ג'יגה-ביט לשנייה, בהתאמה, על גבי כבלי זוגות שזורים. הם מיועדים בדרך כלל לרשתות ליבה, שרתים ומרכזי נתונים.

השכבה הפיזית ב-Ethernet

קידוד - קידוד הוא התהליך שבו הסיביות הבינאריות מומרות לאותות פיזיים שיכולים לעבור בכבל. ב-Ethernet, משתמשים בשיטות קידוד שנקראות Manchester ו-4B5B.

Manchester - בשיטה זו, כל סיבית מיוצגת על ידי מעבר במתח – ממתח נמוך לגבוה (עבור 1) או ממתח גבוה לנמוך (עבור 0). המעבר מתרחש תמיד באמצע הסיבית. זה מאפשר סנכרון קל בין המסדר למקלט.

4B5B - כל קבוצה של 4 סיביות מקודדת ל-5 סימני מתח, כאשר כל סימן יכול להיות חיובי, שלילי או אפס. זה מאפשר לזהות שגיאות ולהשתמש בפחות מעברי מתח, מה שמאפשר להעביר אותות במהירות גבוהה יותר.

אפנון - האפנון הוא השינוי של האות הפיזי כך שישא את המידע הדיגיטלי. ב-Ethernet של 10Mbps משתמשים באפנון בסיסי של מיתוג מתח (voltage switching), אבל ב-Fast Ethernet ומעלה עוברים לשיטות מתקדמות יותר כמו MLT-3 - Multi-Level Transmit-3, שבה יש שלוש רמות של מתח שמשתנות בצורה מחזורית.

סנכרון - כדי שההתקנים יוכלו לתקשר, הם צריכים להיות מסונכרנים – כלומר, לשדר ולקלוט את הסיביות באותו קצב בדיוק. לשם כך, Ethernet משתמש במנגנונים כמו מבוא (Preamble) ומרווח בין מסגרות (Inter-Frame Gap), שהם רצפים ידועים מראש של סיביות שמסמנים את תחילת וסיום השידור.

זיהוי שגיאות - בסוף כל מסגרת Ethernet יש שדה FCS - Frame Check Sequence, שהוא למעשה חתימה מתמטית שנקראת CRC – Cyclic Redundancy Check של כל תוכן המסגרת. כשמקבלים מסגרת, מחשבים מחדש את ה-CRC ומשווים אותו לערך בשדה FCS. אם הם לא זהים, זה אומר שהייתה שגיאה בהעברה והמסגרת נזרקת.

מחברי RJ45 מציעים מגוון של יתרונות, כגון:

- עלות נמוכה
- הרכבה ללא-הלחמה של המחבר והחיווט
- ייצור מהיר של כבלים בהתאמה-מיוחדת
- הרכבה קלה בשטח עם כלי עבודה פשוטים ויכולת התאמה-מיוחדת של כבלים במקום.

מחברי RJ45 יכולים להציע גם מספר תכונות נוספות, כולל:

- קידוד - מבטיח תחילה נכונה
- יכולות הרכבה שונות - מאפשרות שיטות פאנל, לוח, הרכבה-משטחית או חור-עובר
- תצוגות ומחוננים - מציינים את סטטוס החיבור
- מגנטיות משולבת - שנאי מובנה בתוך המחבר, שבעזרתם הרעש נפחת והתקשורת מתייצבת
 - בידוד גלוני - 2 הצדדים מעבירים מידע בלי שיש זרם ישיר ביניהם בעזרת סליל (למקרה ול2 מכשירים יש אדמות שונות אז שלא יהיה זרימה של זרימת איזון ביניהם).
 - סינון של רעשים - העברת האות כהפרש בין 2 חוטים
- Hi-Rel - מעניקה הגנה עבור החומרה והחיבור
 - כלומר כל חלקים הרכיב עשויים מחומרים איכותיים והם עמידים לטמפ' גבוהות נמוכות ללחות להפרעות חשמליות וכולי.

ישנם מספר תקנים ישימים עבור מחברי RJ45, בהתאם ליישום ולשימוש המיועד שלהם.

- ANSI/TIA 1096-A - מכסה את המאפיינים המכניים הבסיסיים, הממדים הפיזיים ודרישות המגעים עבור התקני RJ45.
- T-568A ו-T-568B - תקני החיווט ומערך-הפינים הבסיסיים
- IEEE 802.3af, 802.3bt ו-802.3 - תקני Power over Ethernet (PoE) השונים עבור מחברי RJ45 המסוגלים לספק זרם חשמלי להתקן הקצה.
- IEEE 1394 - מבנה אפיק ממשק נתונים המשמש עם RJ45

תקני T568A ו-T568B

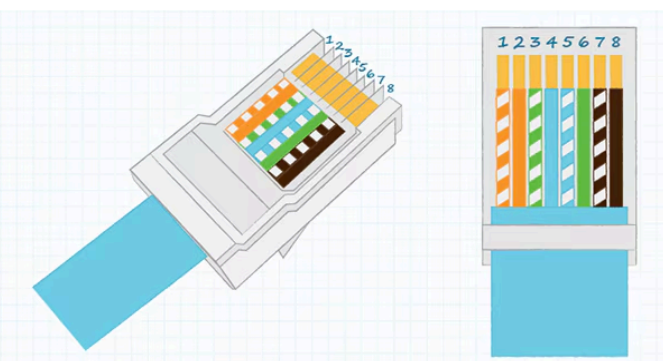
T568A ו-T568B הם שני תקני מערך-פינים RJ45 המציינים את ההקצאה של 8 החוטים שבתוך המחבר. יש להכניס את החוטים המקודדים בצבעים בצורה נכונה למיקומי מערך-הפינים שהוגדרו כדי להבטיח תפקוד תקין של רשת ה-Ethernet.

T568A מציע תאימות לאחר עבור חיווט ישן יותר ו-T568B מספק בידוד אותות טוב יותר והגנה מפני רעשים. מחברי RJ45 Pass-Through מאפשרים להזין את החוטים דרך המחבר ולחתוך אותם במהלך תהליך הלחיצה לטובת יישור קל יותר.

T568A PIN ASSIGNMENT								
Pin	1	2	3	4	5	6	7	8
Color	White/ Green	Green	White/ Orange	Blue	White/ Blue	Orange	White/ Brown	Brown

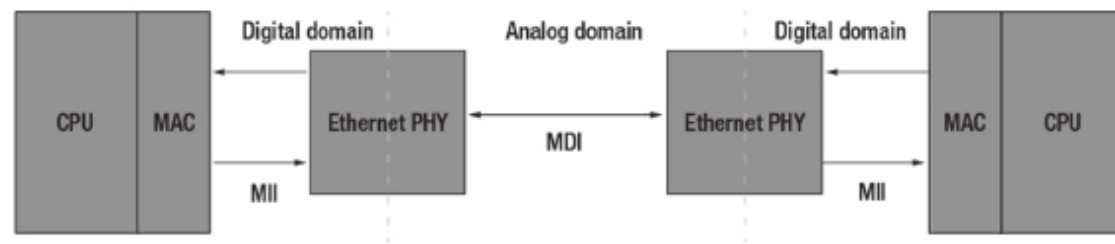
T568B PIN ASSIGNMENT								
Pin	1	2	3	4	5	6	7	8
Color	White/ Orange	Orange	White/ Green	Blue	White/ Blue	Green	White/ Brown	Brown

איור 3: מערכי פינים T568A ו-T568B (מקור התמונה: Same Sky)



ETHERNET PHY

אתרנט פיזי הוא בעצם טרנזמיטר ורסיבר פיזיים שמתחברים פיזית לרכיב אחד או אחר. החיבור הפיזי יכול להיות נחושת כמו CAT5 או סיבים אופטיים



תפקידים עיקריים של הPHY

- המרת אותות – בין אותות חשמליים/אופטיים לבין ביטים דיגיטליים.
- שידור וקליטה – טיפול בשידור הנתונים בקצב מוגדר (למשל 10/100/1000 Mbps).
- זיהוי קו (Link Detection) – קובע האם קיים חיבור לרכיב בקצה השני של הכבל.
- Auto-Negotiation – משא ומתן אוטומטי על מהירות וקידוד התקשורת עם הצד השני.
- Clock Recovery – סנכרון שעון מהאות הנכנס.
- Encoding/Decoding – קידוד ופיענוח האותות (למשל: MLT-3, 4B/5B, 8B/10B וכו').

פונקציות

לPHY יש domain דיגיטלי שמתחבר ישיר לMAC של המכשיר כמו FPGA MCU או CPU. לPHY יהיה במידה משתנה MII - באס מידע באורך 4 ביטים עם קו שליטה וקו שעון בכיוון השליחה והקבלה. MII באים בצורות שונות בהתאם למהירות הMAC והPHY ויש להם כמות פינים שונה בהתאם הטבלה הבאה מציגה את הMII הנפוצים ביותר ומספקת סיכום ברמה גבוהה של היתרונות והחסרונות.

Interface	Pins (pin count)	Speed support (Mbps)	Pros	Cons
MII	RX_D[3:0], RX_CLK, RX_DV, CRS, COL TX_D[3:0], TX_CLK, TX_EN (14)	10, 100	Common pinout, low speed, simple to route, lowest latency	No 1-Gbps support, high pin count
Reduced MII (RMII)	RX_D[1:0], CRS_DV, TX_D[1:0], TX_EN (6)	10, 100	Reduced pin count	Poor deterministic latency (due to first-in, first-out), no 1-Gbps support
Gigabit MII (GMII)	RX_D[7:0], GRX_CLK, RX_CTRL, TX_D[7:0], GTX_CLK, TX_CTRL (20)	10, 100, 1000	1-Gbps support, low latency	High pin count, not commonly supported
Reduced Gigabit MII (RGMII)	RX_D[3:0], RX_CLK, RX_CTRL, TX_D[3:0], TX_CLK, TX_CTRL (12)	10, 100, 1000	1-Gbps support, common pinout	Difficult to route, poor electromagnetic compatibility (EMC)
Serial Gigabit MII (SGMII)	SO_P, SO_M, SI_P, SI_M (4)	10, 100, 1000	1-Gbps support, common pinout, excellent EMC performance, easy to route	More expensive integrated circuits

לPHY יש MDI Medium Dependent Interface שמחבר מכשיר אחד לאחר דרך מדיה פיזית, מתייחסים לזה לרוב כdomains אנלוגי של הPHY כי הוא סיגנל מתמשך תלוי בזמן.

MDI

MDI הוא הממשק הפיזי בין רכיב ה-PHY לבין התווך הפיזי (Physical Medium), כמו כבלי Twisted Pair, סיבים אופטיים או מדיות אחרות להעברת אותות.

תפקידים מרכזיים של MDI:

- המרת אותות דיגיטליים לאותות אנלוגיים:
 - ה-PHY מייצר אותות דיגיטליים בפורמט פנימי (למשל ברמת הלוגיקה).
 - ה-MDI ממיר את האותות הדיגיטליים לאותות חשמליים אנלוגיים, המתואמים לתקן Ethernet, למשל באמצעות קידוד 3B-10B או 5B-PAM ב-100BASE-TX.
- שידור וקליטה דיפרנציאלית:
 - ה-MDI משתמש בקווים דיפרנציאליים (RX+/RX-, TX+/TX-) להפחתת רעשי שורה ורגישות להפרעות אלקטרומגנטיות.
- התאמת עמידות והתנגדות:
 - על ה-MDI לספק התאמת עומסים (impedance matching) בין PHY לבין הכבל, כדי למנוע החזרות אותות (Reflections) ולשפר את איכות השידור.
- בידוד חשמלי:
 - בשלב זה בדרך כלל מוסיפים שנאים המבודדים חשמלית בין המעגלים הפנימיים של ה-PHY לבין הכבל, וכך מונעים זרמים בלתי רצויים, רעשים ושמירה על תקינות האותות.
- טיפול בסיגנלים אנלוגיים בזמן אמת:
 - ה-MDI מטפל באותות רציפים משתנים בזמן (Continuous Time Signals), בניגוד ל-domain הדיגיטלי שבו האותות הם ערכים בינאריים בדידים.
 - הוא מבצע את תהליך השידור בזמן אמת, כולל התאמת רמות מתח, רעשים ורגישות.

בחירת PHY בהתאם ל-MDI

- רוב מייצרי מעגלים משולבים מספקים את המידע הבא
- Data rate לדוגמא 10Mbps 100Mbps 1Gbps
 - Interface support לדוגמא MII RMII GMII RGMII SGMII
 - Media support לדוגמא BASET BASETE BASETX BASET1
- טבלת השוואה של MDI נפוצים:

MDI	IEEE specification (data rate)	Typical systems	Medium	Pros	Cons
10BASE-T/Te	IEEE802.3u (10 Mbps)	Industrial lighting	CAT5	Commonly supported Long reach Low standby power	Low speed
10BASE-T1L	IEEE802.3cg (10 Mbps)	Field transmitters; switches; heating, ventilation and air conditioning controllers; escalators	Unshielded twisted pair (UTP), shielded twisted pair (STP)	Ultra-long reach, bidirectional over single pair, power coupled over the data	Low speed
100BASE-TX	IEEE802.3u (100 Mbps)	PLCs, IP cameras, OBD ports	CAT5	Commonly supported, used by field buses	High emissions, external components
100BASE-T1	IEEE802.3bu (100 Mbps)	Display clusters, head units, gateways, infotainment, avionics communication, robotics, machine vision	UTP, STP	Low emissions, high immunity, bidirectional over single-pair cables	Not as common (no PC connection support), short cable reach
1000BASE-T	IEEE802.3ab (1 Gbps)	IP cameras, test and measurement	CAT6	1-Gbps speed	Expensive cable
1000BASE-T1	IEEE802.3bp (1 Gbps)	Telematics control unit, gateways, avionics communication, robotics, machine vision	UTP, STP	1-Gbps speed, bidirectional over single pair	Not as common (no PC connection support), short cable reach

MII – מממשק תקשורת בין בקר הרשת (MAC) לשכבת הפיזית (PHY), תומך במהירויות של 10 ו-100 מגה-ביט לשנייה, מעביר מידע במקביל על 4 סיביות.

RMII – גרסה מצומצמת של MII, משתמשת בפחות קווים (רק 2 סיביות במקום 4), אך עדיין תומכת ב-100 ו-1000 מגה-ביט לשנייה.

GMII – הרחבה של MII למהירות של 1 גיגה-ביט לשנייה, מעביר מידע במקביל על 8 סיביות.

RGMII – גרסה מצומצמת של GMII, משתמשת בפחות קווים על ידי העברת מידע גם בקצה העולה וגם בקצה היורד של השעון.

SGMII – מממשק סידורי שמעביר את כל המידע על זוג חוטים בלבד (Tx/Rx), חוסך הרבה קווים, ותומך במהירויות של 10, 100 ו-1000 מגה-ביט לשנייה.

MII מכיל:

- ביטים של נתונים – נתוני השכבה הפיזית, מוכן לשידור או התקבל מקו הרשת.
- אותות שליטה – כמו:
 - Transmit Enable – סימון שה-MAC רוצה לשדר.
 - TXD – נתונים לשידור (4 ביטים בכל מחזור שעון).
 - Receive Data Valid – סימון שהתקבל מידע תקף.
 - RXD – נתונים שהתקבלו מהרשת (גם כן 4 ביטים).
 - Collision – זיהוי התנגשות ברשת (בחצי דופלקס).
 - Carrier Sense – זיהוי תעבורה על הקו.
 - קו שעון שמסנכרן את ההעברה – בדרך כלל 25MHz.

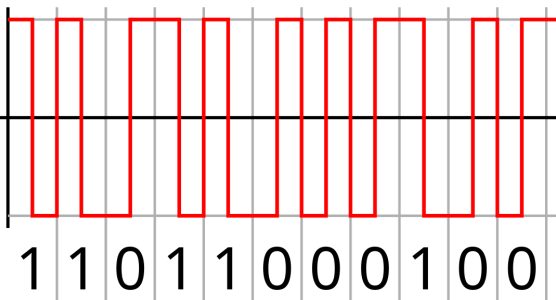
MII הוא תקן שמגדיר איך 2 רכיבים אלקטרוניים (MAC, PHY) מסונכרנים להעברת נתונים.

MII מוגדר לפי תקן LVTTTL או LVCMOS כלומר

בכניסה '1' הוא מעל 2 וולט, '0' הוא מתחת ל-0.8

ביציאה '1' הוא מעל 2.4 וולט, '0' הוא מתחת ל-0.4

קידוד מנצ'סטר



בקידוד מנצ'סטר כל תחילת ביט במידה והביט 1 יתחיל מלמעלה, במידה ו0 יתחיל מלמטה, באמצע הביט הוא ירד/יעלה עד סוף הביט.

PTM

המידע משודר באמצעות שינוי מדויק של מאפיינים הקשורים לזמן של הפולסים. (Pulse-Time Modulation), יש סוגים שונים של PTM:

- **PWM** – שינוי רוחב הפולס (Pulse-Width Modulation)
 - **PPM** - שינוי מיקום הפולס (Pulse-Position Modulation)
- המידע מועבר על ידי שינוי מיקום הזמן של פולס בודד בתוך חלון זמן מוגדר
- **PIM** - שינוי מרווח הזמן בין פולסים (Pulse-Interval Modulation)

Ultra Wideband

UWB הוא פורטוקול תקשורת אל חוטית לטווח קצר שפועל דרך גלי רדיו, UWB פועל בתדרים גבוהים (ספקטרום של תדרי GHz). מגבלת המרחק שלו בפנים עד 20-35 מטרים ובחוץ בדרך כלל עד 200 מטרים (למרות שזה תלוי ברכיב ובחברה, יש חברות שהצליחו להגיע גם ל300 מטרים) וניתן להשתמש בהם ללכידת מידע מרחבי וכיווני בדיוק גבוה מאוד. אפשר לחשוב על UWB כראדאר הסורק ברציפות, שיכול לנעול בדיוק על אובייקט, לאתר את מיקומו ולתקשר איתו.

כשמכשיר המאפשר UWB מתקרב למכשיר אחר המאפשר UWB הם מתחילים לחשב את Tof (הזמן שלוקח למידע לעבור אחר מהשני).

בעזרת ערוץ Bandwidth גדול יותר (500MHz) עם פולסים קצרים (2ns כל אחד) ל-UWB יש דיוק גדול. תהליך זיהוי המיקום של UWB עוקב אחרי מיקום המכשיר בזמן אמת וכך מכשירי UWB יכולים לזהות תזוזה ומיקום יחסי.

טכנולוגיית UWB מספקת דיוק גבוה יותר בקווי ראייה ישירה (LoS) ויכולת מיקום חזקה גם בתרחישים ללא קו ראייה ישיר (nLoS) וגם מסוגלת לתפקד בסביבות שבהן קירות רבים, אנשים ומכשולים אחרים בדרך כלל חוסמים אותות אלו.

LoS יש דיוק של ± 10 ס"מ לעומת זאת ב nLoS הדיוק נפגע מעט (± 30 ס"מ) אך הוא עדיין גבוה באמצעות טכנולוגיית זווית הגעה (AoA), הדיוק בזמן אמת של מדידות UWB מספק שירותי מיקום מדויקים ברמת סנטימטרים.

הסיבה העיקרית ש-UWB מצליח להיות מדויק גם ב-NLOS היא הרזולוציה הגבוהה בזמן שנובעת מהרוחב פס הענק, בגלל רוחב הפס הגדול יש לו פולסים קצרים מאוד בזמן, לכן אפשר להבדיל בין האות הישיר לבין ההחזרות מקירות או חפצים, כך אפשר לבדוד את המסלול הראשון שהגיע גם אם הוא חלש ולהתעלם מהשתקפויות מאוחרות

לא רק זאת, מכשירי UWB יכולים גם לקבוע האם האובייקט עומד במקום, מתקרב או מתרחק. ל-UWB יש דיוק טוב יותר, יכולות מיקום מדויקות יותר וביטחון אלחוטי טוב יותר מאשר WiFi ו-Bluetooth. לעומת זאת כן יש לו חסרונות (במקומות סגורים הוא מצליח להגיע לכמה מטרים לעומת WiFi ו-Bluetooth, יש לו צריכת חשמל גבוהה, קצב העברת נתונים נמוכה יחסית, עלות חומרה גבוהה יותר נכון לעכשיו ותמיכה מוגבלת).

רוחב הפס של UWB גדול מ-500 MHz (או חורג מ-20% רוחב פס חלקי).

כדי לחשב אם הוא חורג מ-20% רוחב פס חלקי צריך את החישוב הבא

$$\frac{F_{\max} + F_{\min}}{2} = F_{\text{avg}}$$

$$F_{\max} - F_{\min} = \text{Bandwidth}$$

$$100 * \frac{\text{Bandwidth}}{F_{\text{avg}}} = \text{Ratio}$$

הבדלים בין שידור רדיו רגיל ל UWB

בשידור רדיו רגיל, משנים את עוצמת האות, התדר או הפאזה של גל סינוס כדי להעביר מידע. לעומת זאת, בשידור UWB משדרים פולסים של אנרגיית רדיו מאוד קצרים שדורשים רוחב פס גדול מאוד. בזכות זה אפשר להעביר מידע בצורה של PPM (אפנון מיקום פולס) או PTM (אפנון זמן פולס). אפשר גם לקודד את המידע באמצעות שינוי הקוטביות של הפולס, העוצמה שלו, או להשתמש בפולסים נפרדים שלא מפריעים אחד לשני. פולסי UWB יכולים להישלח בקצב נמוך כדי להעביר מידע של זמן או מיקום, אבל אפשר גם לשלוח אותם בקצב גבוה מאוד, שמתאים לרוחב הפס הרחב שלהם.

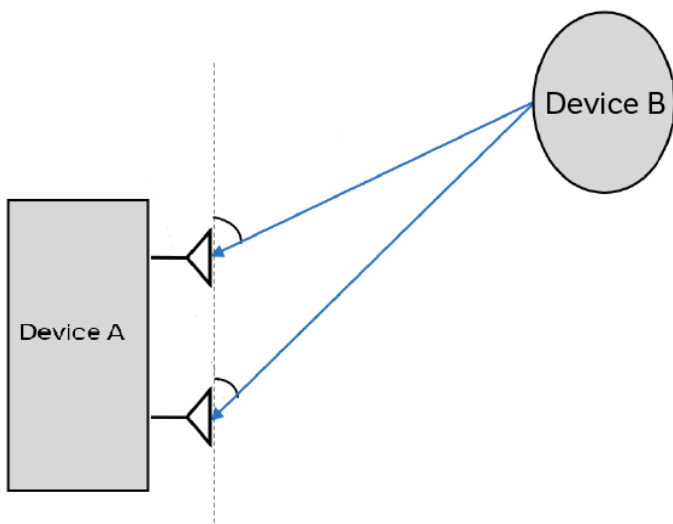
שלבי התקשורת

UWB משתמש ב-2 סוגים של מכשירים: Controller ו- controllee (יכול להיות כמה controllee). המכשיר השולט קובע את הערוץ המורכב מ-2 המכשירים ישתפו בו מידע והוא זה שמתחיל את התהליך והמכשיר הנשלט מגיב לו. כדי להתחיל את תהליך הטווח, המכשירים צריך להטמיע את החלפת המפתחות באמצעות מנגנון מאובטח מחוץ לערוץ (OOB) (בלוטוס עם צריכת אנרגיה נמוכה) OutOfBand - ערוץ נפרד או רשת אחרת שמשמשת לבקרה וניהול של המכשירים פרמטרים של טווח כוללים, בין היתר, כתובת מקומית, ערוץ מורכב ומפתח סשן. חשוב לזכור שהפרמטרים האלה עשויים להשתנות או להסתובב לאחר סיום הטווח, ויש להעביר אותם מחדש כדי להפעיל מחדש את הטווח. טווח התדרים של UWB הוא בדרך כלל בין 3.1 GHz ל 10.6 GHz כשהמכשיר מקבל גלי רדיו בעזרת שימוש בכמה אנטנות, כל אנטנה תקבל פאזה אחרת וכך אפשר לחשב את הזווית והמיקום של המכשיר

שלבי עבודה

זיהוי Controllee:

ה Controller שולח את UWB של 500MHz במידה ויש מכשיר שתומך ב UWB הוא יחזיר לו תשובה, לאחר מכן ה Controller ישלח מפתח ועוד פרמטרים כמו כתובת מקומית ערוץ מורכב וכולי. במידה והנשלט תואם הוא יחזיר לו תשובה



חישוב Tof:

הקונטרולר יחשב את זמן העברת המידע (Time of Flight) בעזרת הנוסחו

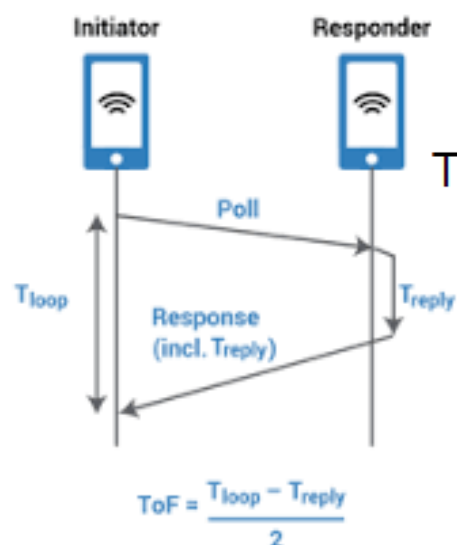
הבאה:

$$T_{of} = \frac{T_{loop} - T_{reply}}{2}$$

לאחר מכן ניתן להכפיל את Tof במהירות

האור כדי לקבל את המרחק בין המכשירים (לפי נוסחת $d=vt$) משתמשים במהירות האור בגלל שגלי הרדיו הם בעצם קרינה אלקטרומגנטית, שנעה במהירות האור בריק.

בנוסף בעזרת מספר אנטנות, הקונטרולר יכול לזהות את זווית הקונטרולר לאחר חישוב Tof, המכשירים יכולים להסתנכרן בקלות ולשלוח אחד לשני מידע (אחד אחרי השני לפי סדר וטווח שנקבע מראש)

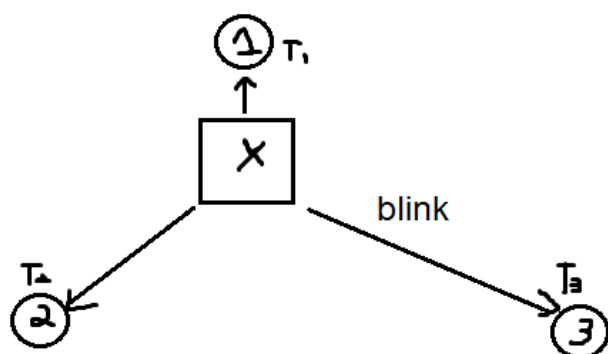


חישוב TDOA

חישוב Time Difference of Arrival זה חישוב שקורה במצב שיש מכשיר ורוצים לחשב את המיקום שלו

ביחס לכמה מכשירים אחרים

המכשיר לדוגמא X שולח לכל שאר המכשירים אות בשם Blink מנקודת הנחה שהמכשירים מסונכרנים ושהמידע על הזמנים שלהם משותף, כל מכשיר מחשב כמה זמן לקח לו לקבל את המידע וכך הם מקבלים בעצם כמה זמנים שונים ובעזרתם אפשר לחשב את המיקום של X ביחס אליהם כמו חישוב מיקום בעזרת לוויינים. צריך 3 מכשירים מינימום כדי למצוא מיקום של מכשיר, 4 מינימום אם רוצים גם גובה.



הגנה מפני Relay Attacks:

מתקפות "הארכה" (Relay) הן מצב שבו תוקף מנסה להוליך שולל את המערכת בכך שהוא "מחדש" את האות בין המכשיר למקור, כאילו הם קרובים זה לזה. ב-UWB, המרחק בין המכשירים מחושב בזמן אמת באמצעות מדידת TOF בדיוק של ננו-שניות. מכיוון שמהירות האור קבועה, כל עיכוב מלאכותי קטן באות יגרום לחישוב מרחק שגוי, מה שמקשה מאוד על תוקף לזייף קרבה פיזית.

הצפנה ואימות:

התקשורת ב-UWB יכולה להיות מוצפנת ברמת הפולסים עצמם, כך שהמידע המקודד לא ניתן לקריאה ללא המפתח המתאים. בנוסף, עוד לפני שמתחיל חישוב המיקום, מתבצעת החלפת מפתחות והגדרות פרמטרים בערוץ נפרד ומאובטח (OOB – Out Of Band), לדוגמה באמצעות Bluetooth Low Energy. כך ניתן לאמת את זהות המכשיר, למנוע התחברות לא מורשית, ולהבטיח שמידע המיקום יגיע אך ורק לצד המורשה.

פרוטוקולים וסטנדרטים רלוונטיים

• 802.15.4a

- הרחבת PHY לתמיכה ב-UWB.
- מוסיף יכולות מיקום מדויק בעזרת TOF ו-TDOA.
- מגדיר איך לשדר פולסים קצרים מאוד (ננו-שניות) ברוחב פס גדול.
- תומך במיקומים גם ב-LoS וגם ב-nLoS.
- קצבי נתונים אופייניים: 110 kbps, 850 kbps, 6.8 Mbps.

• 802.15.4z

- עדכון ושדרוג ל-4a, עם שיפור באבטחה ובדיוק.
- מוסיף Secure Ranging – מנגנון אתגר-תגובה מוצפן כדי למנוע Relay Attacks.
- משפר את עמידות האות להפרעות ולקליטה רעה.
- מאפשר שילוב קל יותר עם BLE, NFC, וערוצים אחרים לערוץ OOB.

מידע נוסף על סוגי אותות קידוד ואפנון

סוגי אותות

Multi-Band Orthogonal Frequency Division Multiplexing Ultra-Wideband MB-OFDM

הוא סוג של שידור UWB שבו:

- תחום התדר הרחב של ה-UWB מחולק למספר תתי-ערוצים כל אחד ברוחב של בערך 500MHz.
- בתוך כל תת-ערוץ, השידור מתבצע בשיטה שנקראת OFDM שבה מחלקים את הערוץ למספר נשאים צרים שמשודרים במקביל בצורה שלא מפריעה אחד לשני.
- השידור לא נשאר תמיד על אותו תת-ערוץ, אלא "קופץ" בין תתי-ערוצים לפי דפוס מוגדר מראש כדי להפחית הפרעות ולשפר עמידות לרעש.
- כדי שהגלים לא יתנגשו זה בזה דואגים שכשאחד מהם נמצא בפסגה כל שאר הגלים מקבלים ערך של אפס.

Impulse Radio Ultra Wide Band IR-UWB

- במקום לשדר אות בעל תדר נשא מסוים IR-UWB משדר פולסים מאוד קצרים של אנרגיה רדיו בדרך כלל מתחת לננו-שנייה.
- פולסים אלו מועברים בטווח תדר רחב מאוד (מספר גיגה-הרץ רוחב פס).
- כל פולס הוא בעצם עליה קצרה וחדה שמשדרת אנרגיה לרגע קצר מאוד, כך שהאנרגיה מתפרסת על כל רוחב הפס של הספקטרום.
- המידע מקודד על ידי שינוי זמני של הפולסים (PPM), או על ידי שינוי קוטביות הפולס (BPSK), וגם משתמשים בטכניקות Time-Hopping (קפיצה בזמנים לפי סדר קבוע או אקראי) כדי לאפשר שיתוף תדר בין מכשירים שונים ולהקטין הפרעות.

מתי משתמשים ב-MBOFDM?

- ריבוי משתמשים ותקשורת מורכבת
- דרישות ליציבות וביצועים גבוהים
- קצב נתונים גבוה
- עמידות טובה יותר להחלקות תדר
- יישומים בתקשורת אלחוטית רחבה

מתי משתמשים ב-IRUWB?

- מדידות מרחק ומיקום מדויק
- צריכת אנרגיה נמוכה
- פשטות יחסית של המערכת
- תקשורת טווח קצר מאוד
- מערכות שיתוף תדר בין כמה מכשירים

שיטות קידוד \ אפנון

:Time hopping

בUWB השידור מבוסס על פולסים מאוד קצרים שמשודרים ברוחב פס מאוד רחב.

- במקום לשדר פולס אחד בקצב קבוע, ב-Time Hopping משנים את מיקום הזמן שבו פולס מסוים משודר, לפי רצף קפיצות זמן (hopping sequence).
- כל פולס משודר בפריים זמן ארוך יותר (כדי שלא יפגע בפולס אחר), והזמן המדויק שבו הפולס יוצא משתנה בהתאם לרצף קפיצות הזמן.
- הקפיצות מתבצעות בתוך מסגרת זמן שחוזרת על עצמה, כשבכל מסגרת הפולס זז לזמן אחר בהתאם לרצף.

:Polarity Coding - Bipolar pulses

שיטה שבה משנים את הקוטביות (הפאזה) של כל פולס — כלומר, האם הפולס הוא חיובי (+) או שלילי (-).
ב-bipolar הפולסים יכולים להיות חיוביים או שליליים לפי קוד מסוים.
לדוגמה, הרצף של הפולסים יכול להיות: פולס ראשון חיובי, פולס שני שלילי, ושלישי חיובי וכן הלאה.

: PPM - Pulse Position Modulation

מחלקים את חלון הזמן ל-Time Slots, בכל חלון כזה יש פולס אחד בלבד.
מיקום הפולס בתוך החלון קובע את המידע.

דוגמה בסיסית:

פולס בהתחלה = 0, פולס בסוף = 1

אפשר גם להגדיר יותר משתי מיקומים:

- 4 מיקומים שונים - 2 ביטים לכל פולס
- 8 מיקומים שונים - 3 ביטים לכל פולס

: BPSK Binary Phase Shift Keying

שיטת אפנון דיגיטלית שבה המידע המקודד מיוצג על ידי שינוי פאזה של גל נושא בין שתי מצבים אפשריים בלבד, לדוגמא:

פאזה 0° = מייצגת ביט 0

פאזה 180° = מייצגת ביט 1

: BPM Bi-Phase Modulation

כל שינוי פאזה שמתרחש לפחות פעם אחת בכל ביט.

שיטת הקידוד	יתרונות	חסרונות
Time hopping	מפחית הפרעות והתנגשויות בין משדרים. משפר אבטחה ועמידות.	לא מקודד מידע בעצמו (משמש כפיזור).
Bipolar pulses	משפר יחס אות לרעש. מוסיף שכבת אבטחה וקידוד.	דורש סינכרון פאזה מדויק.
PPM	עמיד לרעש בעוצמה (כי מידע תלוי בזמן, לא בעוצמה). מתאים לאותות פולס קצרים של UWB.	דורש סנכרון זמן מדויק.
BPSK	עמיד מאוד לרעש קל ליישום קל לפענוח יעיל מבחינת ניצול אנרגיה	קצב נתונים מוגבל (1 ביט לנשא) דורש סינכרון פאזה

DW1000 DecaWave

טרנזמיטר אלחוטי (משתמש בטכנולוגיית UWB) בהתאם לפרוטוקול IEEE802.15.4-2011.

מאפשר מיקום של חפצים במערכות זמן אמת (RTLS) לדיוק של עד 10 ס"מ במקום סגור.

מאפשר קצב שליחת נתונים גבוה (עד 6.8 Mb/s)

טווח תקשורת עד 300 מטרים בעזרת טכניקות coherent receiver.

יכול לתקשר עם עד 11,000 ברדיוס של 20 מטרים.

חסין מאוד למכשולים בסביבה.

צריכת מתח נמוכה - מאפשר שימוש בבטריות לזמנים ארוכים

טווחי ה LOS וה NLOS הגדולים מקטינים את כמות

התשתיות שצריך בשביל להפעיל את המערכות

מעולה לשימוש במערכות RTLS (מיקום בזמן אמת)

ולמערכות חיישני רשת אלחוטי.

מידע טכני

תומך בקצבי העברת נתונים של

6.8mbit/s, 100kbit/s, 850kbit/s

המערכת יכולה לפעול ב-6 ערוצים, ותדרי המרכז שלהם יכולים להיות בין התדרים 3.5GHz ל-6.5GHz.

עוצמת שליחה של -14dBm או -10dBm

צפיפות שידור קטנה מ 41.3dBm / MHz -

אורך אות המבוא הוא בין $64\mu s$ ל $4ms$.

תמיכה בשליחת חבילות מידע עד אורך של 1023 בייטים.

קידוד BPM עם BPSK

(BPM כל שינוי פאזה שמתרחש לפחות פעם אחת בכל ביט

כמו מנצ'סטר)

תמיכה פנימית בשיטות בדיקת ותיקון שגיאות CRC | FEC

ממשק SPI לhost controller (מהירות מקסימלית 20 MHz)

מאפשר אינטגרציה עם מספר רחב של מיקרוקונטרולים

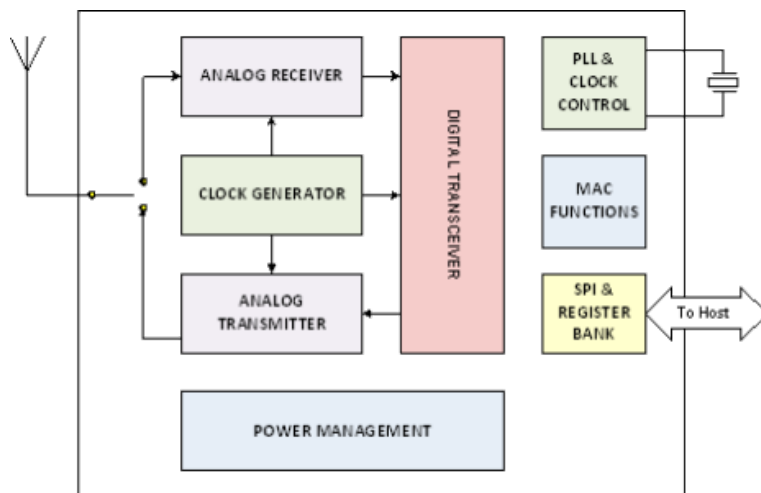
מתח אספקה יחיד של 2.8V או 3.6V

מצב שליחה מ31mA

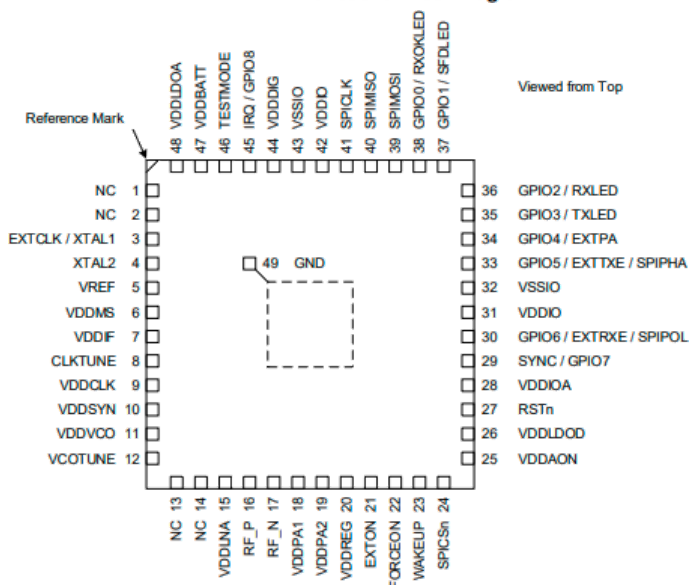
מצב קליטה 64mA

watchdog timer מצב $2 \mu A$

deep sleep מצב 100nA



Functional Block Diagram



DW1000 Pin Diagram

טכניקות media access:

- FDMA עם 6 ערוצים
- CDMA עם 12 קודים לערוצים שונים.

תומך במדידת מרחק חד כיוונית ודו כיוונית

שימוש ב TOF וב TDOA.

טווח טמפרטורות של מינוס 40 עד 85 מעלות צלזיוס.

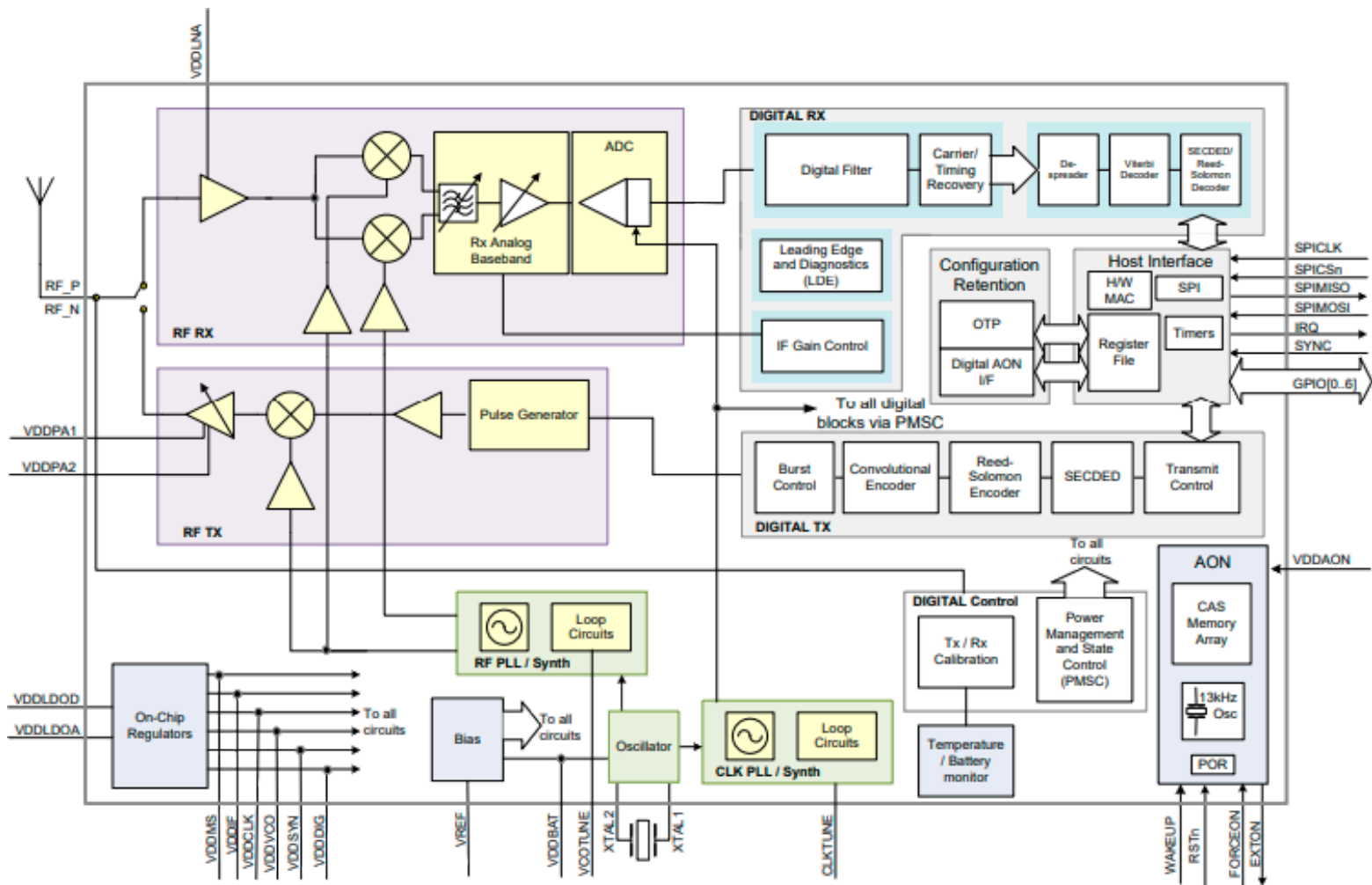


Figure 1: IC Block Diagram

שיטות בדיקת ותיקון שגיאות :

: FEC

מוסיף מידע עודף לנתונים המשודרים, כך שהמקלט יכול לאתר וגם לתקן שגיאות קטנות מבלי לבקש שידור חוזר.

עובד בזמן אמת אין צורך להחזיר (acknowledge) ACK או לבקש מידע עודף.

לפני השידור, הנתונים עוברים קידוד, אם בזמן השידור יש ביטים ש"התהפכו" עקב רעש או הפרעות, המקלט יכול לזהות ולתקן אותם הודות לנתוני העודף.

לפני שמשדרים את המידע, אנחנו לא שולחים רק את הביטים עצמם – אנחנו מוסיפים ביטים נוספים שהם לא "המידע המקורי" אלא חוקים מתמטיים שמתארים את המידע הזה.

יתרונות:

- מאפשר לתקן שגיאות שנגרמות מרעש או החזרות מרובות בלי לבקש שידור חוזר.
- משפר את הטווח האפקטיבי של המודול, ניתן לשדר באותה עוצמה ועדיין לקבל תקשורת יציבה במרחק גדול יותר.
- מפחית דיליי – אין צורך ב-ACK/NACK לשגיאות קטנות.

חסרונות:

- מוסיף עודף ביטים, מעט מוריד את קצב הנתונים האפקטיבי.
- צורך יותר עיבוד בשידור ובקבלה, צריכת חשמל גבוהה יותר.
- תיקון שגיאות חזק מאוד יכול להיות בזבוז משאבים אם הקו נקי.
- .

: CRC

מוסיף קוד בדיקת שלמות שמאפשר למקלט לזהות שגיאות, אבל לא לתקן אותן.

אם יש שגיאה, המידע נפסל והמקלט צריך לבקש שידור מחדש (אם הפרוטוקול תומך בזה).

לפני השידור, המידע עובר חישוב CRC (אלגוריתם מתמטי המבוסס על פולינומים בינאריים).

בקצה השני, מבצעים את אותו חישוב על המידע שהתקבל ומשווים לקוד שנשלח.

אם הקוד לא תואם – יש שגיאה.

יתרונות:

- מאוד קל ומהיר לחישוב, מתאים למערכות עם MCU חלש או צריכת חשמל נמוכה.
- מוסיף רק מעט מאוד מידע עודף (למשל 16 או 32 ביטים פר פריים).
- מובנה ב-DW1000 כחלק מה-MAC, כך שאין כמעט עלות נוספת בשימוש.

חסרונות:

- מזהה שגיאות אבל לא מתקן, אם יש שגיאה חייבים לשדר מחדש.
- אם התקשורת חד-כיוונית (OWR) או שאין אפשרות לשדר שוב, מידע עם שגיאה הולך לאיבוד.
- לא משפר את הטווח כמו FEC, אם יש הרבה רעש, תצטרך שידורים חוזרים.

טכניקות media access:

FDMA – Frequency Division Multiple Access

כל משתמש/תחנה מקבלת תדר ייחודי משלה.
כל ערוץ מוגדר ברצועת תדר מסוימת (למשל 3.1–4 GHz, 4–5 GHz וכו').
כל מכשיר ששולח בערוץ מסוים לא יפריע למכשירים אחרים בערוצים אחרים.

יתרונות FDMA

- הפרעות מינימליות בין ערוצים שונים.
- קליטה פשוטה רק לבחור את הערוץ הרצוי.

חסרונות FDMA

- מספר הערוצים מוגבל (רק 6).
- לא גמיש במיוחד אם מספר המשתמשים גדל.

CDMA – Code Division Multiple Access

כל משתמש מקבל קוד ייחודי שממירה את האות שלו לסדרה דיגיטלית מורכבת.
האות מוכפל בקוד ייחודי לכל משתמש.
המקלט יודע את הקוד ומסנן רק את האות המתאים לו.
כל השאר נשאר "רעש" בלתי מזיק.

יתרונות CDMA

- מאפשר יותר משתמשים על אותו ערוץ (12 במקום 1).
- עמידות גבוהה יותר לריבוי מסלולים והפרעות.

חסרונות CDMA

- מורכב יותר דורש קידוד/פענוח דיגיטלי.
- יעילות האנרגיה קצת פחותה (כי צריך לפענח אותות מורכבים).

מבנה פיזי

מקדימה הרכיב מקבל מידע אנלוגי בעזרת האצ והאז שלו שמחוברים בעזרת מתג לאנטנה (שמחליטה אם קולטים או פולטים מידע)

בתוך הרכיב יש מעקב על מתח ומעקב על טמפ, החלק האחורי מעביר מידע דיגיטלי לhost processor חיצוני.

הרכיב לוקח את האות שנקלט, מגביר אותו בlow noise amplifier, לאחר מכן מנמיכים אותו ישירות לbaseband.

Baseband הוא טווח התדרים המקורי של האות לפני שהוא מונחה על גלי רדיו.

מבצעים הגברה לפני שמנמיכים את האות למיקסר כי הרעש הפנימי של המיקסר יכול להרוס את SNR המקלט מתוכנן במיוחד בשביל להשיג 3 דברים עיקריים בקליטה:

- קליטת טווח רחב של תדרים או אותות מהירים מאוד (קליטת טווח רחב בעזרת אנטנה רחבת פס)
 - שמירה על יחס ישר בין האות שנכנס לאות שיוצא.
 - מדד לכמה רעש המקלט מוסיף לאות בזמן ההגברה.
- זה מאפשר לערוצי UWB להיות מומרים חזרה עם מינימום רעש מוסף.

את סיגנל baseband מעבירים בתהליך שמפרק את הקידוד שלו,

ולאחר מכן האות ניהיה זמין לhost controller בעזרת SPI.

שרשרת פולסי האצ נוצרת על-ידי יישום נתוני שידור המקודדים דיגיטלית על גנרטור פולסים אנלוגים. שרשרת הפולסים מועלת בתדר על-ידי double balanced mixer לנשא המיוצר על-ידי הסינתסיזר וממוקדת על אחד מערוצי ה-UWB המורשים.

סינתסיזר -מעגל אלקטרוני שמייצר תדרים מדויקים ויציבים על בסיס מקור תדר אחד (למשל מתנד קריסטל).

צורת גל הרדיו לאחר המודולציה מוגברת לפני שידורה מהאנטנה החיצונית.

לרכיב יש זיכרון OTP (לא נדיף), אפשר להשתמש בו כדי לאחסן מידע על הקליברציה של עוצמת tx, תיקון שגיאה של תדר הקריסטל וכוונון דיוק טווח.

אפשר להשתמש בזכרון AON (תמיד פעיל) כדי לשמור קונפיגורציות של מידע מDW1000 במצב תפעול הספק הנמוך ביותר כשהרגולטורי מתח ברכיב כבויים.

המידע הזה מועלה ומורד באופן אוטומטי

סכמת השעונים מבוססת על 3 מעגלים עיקריים:

- מתנד הקריסטל
- CLOCK PLL
- RF PLL

המתנד מתוכנן לפעול בתדר של 38.4MHz בעזרת קריסטל חיצוני
אפשר להוסיף במקום הקריסטל שעון סיגנל חיצוני של 38.4MHz, משתמשים בשעון זה כreference inputs
2 מעגלי PLL.
הclock מייצר את הclock שהחלק הדיגיטלי האחורי צריך כדי לעבד את הסיגנל
הRF PLL מייצר את הdown conversion local oscillator
מסופק מתנד פנימי של 13kHz בשביל הSLEEP state
הממשק המארח כולל SPI כדי לתקשר עם מכשירים, יש גם מספר תכונות MAC מיושמות כולל יצירת CRC
בדיקת CRC וסינון מסגרות בקליטה.

פינים

שם הסיגנל	PIN	I/O	תיאור
ממשק הקריסטל			
EXTCLK / XTAL1	3	קלט אנלוגית	קלט קריסטל reference או פין reference חיצוני
XTAL2	4	קלט אנלוגית	קלט קריסטל התייחסות. השאר פתוח אם משתמשים בשעון חיצוני.
ממשק דיגיטלי			
SPICLK	41	קלט דיגיטלי	שעון SPI
SPIMISO	40	פלט דיגיטלי	פלט מידע SPI
SPIMOSI	39	קלט דיגיטלי	כניסת מידע SPI
SPICSn	24	קלט דיגיטלי	פעיל בנמוך, המעבר מגבוה לנמוך מסמל מעבר, SPI בחירת שבב DW שיוציא את wakeup חדשה, יכול גם לשמש כסיגנל SPI לשליחת DEEPSLEEP או מSLEEP
SYNC / GPIO7	29	קלט ופלט דיגיטלי בדרך כלל קלט	משמש כסנכרון חיצוני, כשלא משתמשים בו לסנכרון אפשר להגדיר אותו כIO כללי.
WAKEUP	23	קלט דיגיטלי	כשהוא במצב גבוה, הוא מביא את DW ממצב SLEEP או DEEPSLEEP למצב פעולה, כשהפין לא פועל הוא צריך להיות מחובר vssio
EXTON	21	פלט דיגיטלי	מכשיר איפשר חיצוני, נטען במהלך תהליך הwakeup ומושאר גבוה עד שהמכשיר נכנס למצב שינה אפשר להשתמש בזה כדי לשלוט בממירי DCDC חיצוניים או במעגלים אחרים שלא אכפת להם אם המכשיר במצב שינה כדי למזער את צריכת החשמל אם לא משתמשים בEXTON אפשר להשאיר אותו פתוח.
FORCEON	22	קלט דיגיטלי	לחבר אותו לאדמה.

IRQ / GPIO8	45	קלט ופלט דיגיטלי	פלט בקשת interrupt מה DW אל hostn. פין זה מנותק בשינה ובמצבי שינה עמוק ועלול לגרום להפרעות מזויפות אלא אם כן מושכים אותו לנמוך. אם לא משתמשים בזה אפשר להפוך אותו ל i/o כללי ל- PIN זה יש pulldown פנימי ל- VSSIO וניתן להשאיר אותו ללא חיבור אם לא נעשה בו שימוש.
GPIO6 / EXTRXE / SPIPHA	30	קלט ופלט דיגיטלי בדרך כלל קלט	כללי i/o בהפעלה הוא פועל כפין של SPI phase selection להגדרת קונפיגורצית מצב פעולת ה SPI לאחר הפעלה הוא יהיה i/o כללי זה עשוי להיות מוגדר לשימוש כ (איפשר מקלט חיצוני). הפין עולה ל 1 כש DW במצב RECEIVE ל- PIN זה יש pulldown פנימי ל- VSSIO וניתן להשאיר אותו ללא חיבור אם לא נעשה בו שימוש.
GPIO5 / EXTTXE / SPIPOL	33	קלט ופלט דיגיטלי בדרך כלל קלט	כללי i/o בהפעלה הוא פועל כפין של SPI phase selection להגדרת קונפיגורצית מצב פעולת ה SPI לאחר הפעלה הוא יהיה i/o כללי זה עשוי להיות מוגדר לשימוש כ (איפשר משדר חיצוני). הפין עולה ל 1 כש DW במצב Transmit ל- PIN זה יש pulldown פנימי ל- VSSIO וניתן להשאיר אותו ללא חיבור אם לא נעשה בו שימוש.
GPIO4 / EXTPA	34	קלט ופלט דיגיטלי בדרך כלל קלט	פין i/o כללי זה יכול להיות מוגדר כדי לאפשר מגבר חשמל חיצוני במידת הצורך. לפין זה יש pulldown פנימי ל vssio ואפשר להשאיר אותו לא מחובר אם הוא לא בשימוש
GPIO3 / TXLED	35	קלט ופלט דיגיטלי בדרך כלל קלט	פין i/o כללי אפשר להשתמש בו כלד שנדלק לאחר שידור לפין זה יש pulldown פנימי ל vssio ואפשר להשאיר אותו לא מחובר אם הוא לא בשימוש
GPIO2 / RXLED	36	קלט ופלט דיגיטלי בדרך כלל קלט	פין i/o כללי אפשר להשתמש בו כלד שנדלק בזמן קליטה לפין זה יש pulldown פנימי ל vssio ואפשר להשאיר אותו לא מחובר אם הוא לא בשימוש
GPIO1 / SFDLED	37	קלט ופלט דיגיטלי בדרך כלל קלט	פין i/o כללי אפשר להשתמש בו כ SFDLED כלומר לד שנדלק לאחר זיהוי START FRAME לפין זה יש pulldown פנימי ל vssio ואפשר להשאיר אותו לא מחובר אם הוא לא בשימוש
GPIO0 / RXOKLED	38	קלט ופלט דיגיטלי בדרך כלל קלט	פין i/o כללי אפשר להשתמש כלד שנדלק לאחר קבלת מסגרת טובה לפין זה יש pulldown פנימי ל vssio ואפשר להשאיר אותו לא מחובר אם הוא לא בשימוש
RSTn	27	קלט ופלט דיגיטלי בדרך כלל פלט, גבוה אחרי איפוס	פין RESET עם מוצא פעיל בנמוך ניתן להוריד אותו ל 0 עם open drain driver חיצוני כדי לאפס את ה DW1000 אסור לעלות אותו עם מקור חיצוני.

TESTMODE	46	קלט דיגיטלי	לחבר לאדמה
מתחי Reference			
VREF	5	מוצא וכניסה אנלוגי	משמש לייצור זרם ייחוס בשבב. יש לחבר לנגד של 11 קילו-אווהם ($\pm 1\%$ שגיאה) לאדמה.
מקורות מתח דיגיטלים			
VDDLDO	26	מתח	מתח חיצוני למעגלים דיגיטלים
VDDIOA	28	מתח	מתח חיצוני לטבעת IO דיגיטלית קבוצה של פינים שניתן להגדיר אותם ככניסות או יציאות דיגיטליות
VSSIO	32 43	אדמה	מתח שלילי לטבעת IO לחבר לאדמה.