מכללת עזריאלי להנדסה ירושלים

המחלקה להנדסת חשמל ואלקטרוניקה

**שעון -**

**עיצוב ותפקידים**

עבודה סמינריונית

מאת:

אור ולורט

עשהאל הדר

ירושלים אב תשפ"א

מכללת עזריאלי להנדסה ירושלים

המחלקה להנדסת חשמל ואלקטרוניקה

**שעון -**

**עיצוב ותפקידים**

עבודה סמינריונית

מאת:

אור ולורט

עשהאל הדר

ירושלים אב תשפ"א

תוכן העניינים

תקציר............................................................................א

פרק א' – יעוד, מבנה ואופן הפעולה................................... 1

יעוד...........................................................1

מבנה השעון................................................2

אופן הפעולה...............................................4

פרק ב' - בעיות סנכרון ופתרונן..........................................6

בעיות סנכרון פנימיות............................................6

Clock Jitter...............................................6

Clock Skew..............................................7

בעיות סנכרון חיצוניות...........................................9

Data Hazard.............................................9

פתרון הבעיות...........................................10

רשימת מקורות..............................................................11

Abstract.........................................................................I

רשימת הפניות לתרשימים, טבלאות ומפות

איור 1 - מונה אסינכרוני.

איור 2 - מערכת סינכרונית.

איור 3 - מבנה מכני, סימול חשמלי ומעגל שקול.

איור 4 - רכיב שעון בתצורה סופית.

איור 5 - תצוגת גלים של VCO .

איור 6 - תרשים סכמטי של PLL.

איור 7 - Data Rates Type.

איור 8 - Clock Jitter.

איור 9 - Violation types of clock skew.

איור 10 - עיקרון ה-Gray Fifo.

רשימת קיצורים וראשי תיבות

AC - Alternating Current

RC - resistor-capacitor

LC - inductor-capacitor

RL - resistor-inductor

CPU - Central Processing Unit

LSB - Least significant bit

MSB - Most significant bit

PLL - Phase Lock Loop

LPF - Low Pass Filter

VCO - Voltage Controlled Oscillators

SDR - Single Data Rate

DDR - Double Data Rate

QDR - Quad Data Rate

FPGA - Field-Programmable Gate Array

CDC - Cross Domain Clocking

WAW - Write After Write

WAR - Write After Read

RAW - Read After Write

תקציר

המעגלים שבהם אנו משתמשים כדי להרכיב מכשירים אלקטרוניים בכלל ומחשבים בפרט מתחלקים באופן גס לשניים: המעגלים הסינכרוניים והמעגלים הא-סינכרוניים.

המעגלים הא-סינכרוניים הם מעגלים בעלי רכיבים אשר פועלים באופן עצמאי ללא רכיב שעון המתזמן את פעולותיהם. לרוב, רכיבים אלו יעבדו לאט יותר כך שכדי שכל רכיב יתחיל את פעולתו הוא חייב שהרכיב שלפניו יסיים את פעולתו הוא, גם אם תיאורטית הם יכולים לעבוד במקביל. מסיבה זו הרכיב העוקב תלוי ברכיב המוביל (אין pipelining).

המעגלים הסינכרוניים, לעומתם, הם מעגלים אשר מקיימים סנכרון בין פעולות בתוך המעגל ובעצם מאפשרים עבודה במקביל (pipelining). כדי לאפשר עבודה במקביל בתוך המעגל, דרוש שעון שיתזמן את תחילת הפעולות של כל רכיב במעגל.

שעון זה יוציא אות שעון (Clock rate) אשר יגיע לכל רכיב ויתנה את פעולתו בעת עליית שעון, ירידת שעון או שניהם. אות השעון יצויר באופן גרפי כגל ריבועי בעל duty cycle של 50%, אשר בעת עליית שעון  
(Rising edge) ייתן ערך של '1' לוגי ובעת ירידת שעון (Falling edge) ייתן ערך של '0' לוגי.

מחזור השעון נמדד ביחידות של זמן (בדרך כלל ns) ותדירות השעון נמדדת ביחידות של הרץ (Hz) בדרך כלל kHz, MHz וכן GHz.

רכיב השעון הוא בעצם רכיב המסוגל לפלוט מתח חשמלי AC, המתבטא באופן גרפי כגל סינוסי המומר לגל ריבועי. רכיב זה מורכב לרוב מגביש קריסטל (לרוב קוורץ - Quartz) המצוי תחת מתח חשמלי. לגבישים תכונה ייחודית להפיק מתח חשמלי כאשר הם מצויים תחת מתח חשמלי בעצמם וזהו הבסיס לכל השעונים מבוססי הקריסטל. חוץ משעונים אלו קיימים שעונים נוספים המבוססים על מעגלי RC או LC (מעגלים המורכבים מנגד וקבל או סליל וקבל).

מכיוון שהשעון הוא רכיב המסתמך בסופו של דבר על תכונות של חומר אורגני, הניסיון לייצר ממנו מוצר מדויק הוא מוצלח באופן **כמעט** אבסולוטי. תכונות החומר גורמות לסטיות קלות המייצרות בעיות סנכרון בין רכיבי המעגל, אשר מתבטאות בסופו של דבר בבעיות קריאה וכתיבה של רכיבים המסונכרנים ע"י השעון. בעיות קטנות יוצרות בעיות גדולות.

בעיות אלו מתחלקות לשניים: בעיות פנימיות למעגל השעון ובעיות הנוצרות בסנכרון בין שעונים. הבעיות הפנימיות למעגל השעון נובעות משני אלמנטים עיקריים: הטיית שעון (Clock skew) וריצוד שעון (Clock jitter), כאשר הטיית השעון היא סטיה הנובעת מקצב התפשטות אות השעון לכלל הרכיבים המחכים לאות השעון, כדי להתחיל פעולה ואילו ריצוד השעון הוא סטיה הנובעת מתכונה פנימית של התקני שעון המייצרת סטיה קלה בעליה או בירידה (או בשניהם).

הבעיות הנוצרות בעקבות סנכרון בין שני מעגלי שעון הן בעיות מעט יותר מורכבות. כאשר שני רכיבים המתקשרים אחד עם השני מתוקף תפקידם (לדוגמא CPU מתקשר עם DDR) וכל אחד מהם רץ בקצב שעון שונה, יש הכרח לסנכרן את הפעולות ההדדיות ביניהם, שאם לא כן יכולות להיווצר בעיות של Overflow או Underflow. כלומר בעיות בקריאה וכתיבה של נתונים מרכיב אחד לשני וכך בעצם על ידי דריסה של נתונים חדשים או שימוש בנתונים ישנים, לשבש את קצב הפעולה של המכשיר כולו.

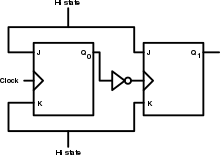
מכיוון שקצב אות השעון הוא שקובע את תחילתה של פעולת כל אחד מהרכיבים במעגל, משתמשי הקצה הבינו כי ניתן לאלץ את המכשיר לפעול בתדר שעון שונה מתדר הגדרת היצרן ובכך להשיג תוצאות מהירות ואפקטיביות יותר ממה שהיצרן הצהיר כי המכשיר מסוגל. תהליך זה נקרא Overclocking ואם עושים בו שימוש נכון אפשר ליעל את פעולת המכשיר, ולעומת זאת אם עושים בו שימוש לא נכון (שימוש בתדרים גבוהים במיוחד) המכשיר יכול להתחמם יתר על המידה ואריזתו יכולה שלא לעמוד בחום הרב הנפלט ולעלות באש.

פרק א': יעוד, מבנה ואופן הפעולה

יעוד:

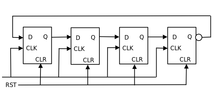
ניתן לחלק את המעגלים החשמליים לשני סוגים של מעגלים: מעגלים סינכרוניים[[1]](#footnote-1) ומעגלים א-סינכרוניים. במעגלים א-סינכרוניים פעולות המעגל יהיו תלויות אחת בשניה ויקרו לרוב אחת אחרי השניה ולא יוכלו לקרות בו זמנית. לעומתם, המעגלים הסינכרוניים מסוגלים לבצע מספר פעולות במקביל (pipelining)[[2]](#footnote-2), כאשר רכיב הנקרא שעון  
 (Computer Clock) מתזמן את תחילת הפעולות בקצב מסוים. חשיבותו של השעון במעגלים הסינכרוניים עצומה - המעגל לא יכול לתפקד ללא שעון שיסנכרן בין כל הרכיבים ויקבע מתי כל רכיב מתחיל את פעולתו ומכאן שמם של המעגלים - סינכרוניים. לעומתם, המעגלים  
הא-סינכרוניים אינם דורשים שעון שיתזמן את פעולותיהם משום שאינם יכולים לבצע פעולות באופן מקבילי.

דוגמא טובה ניתן לראות במונים א-סינכרוניים. במונים אלו הבנויים מדלגלגים [[3]](#footnote-3)(flipflops) המספר ישתנה בהדרגה דלגלג אחרי דלגלג, מהסיבית הנמוכה ביותר (lsb) ועד הסיבית הגבוהה ביותר (msb). בעצם, בכדי לקדם את הספירה לוקח לאות זמן לעבור דרך כל הדלגלגים. לעומתם, במונים סינכרוניים הפועלים לפי קצב שעון ישתנו כל ספרותיהם בו זמנית. מי שיכתיב את השינוי יהיה השעון וכך בעצם לא נצטרך לחכות לאות שיעבור דרך כל הדלגלגים.



בתמונה זו ניתן לראות תרשים של מונה א-סנכרוני המבוסס על דלגלגי JK.

איור 1



בתמונה זו ניתן לראות מערכת סינכרונית המורכבת מדלגלגי D.

איור 2

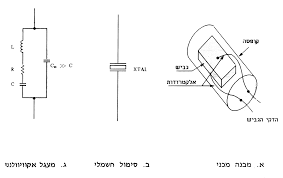
מבנה השעון:

השעון מורכב משני רכיבים עיקריים - מתנד ומגבר.

מתנד - המתנד הוא בעצם מעגל חשמלי שתפקידו ליצור אות חשמלי מחזורי. ברוב המקרים המתנד ייצר אות חשמלי המתבטא באופן גרפי כגל סינוסי או כגל ריבועי (קיימות גם צורות גל נוספות כגון "שן מסור")[[4]](#footnote-4).

קיימות מספר דרכים לבניית מתנדים המפיקים אותות חשמליים אלו:

מתנד קריסטל - מתנד זה מבוסס על גביש קריסטל קטן אשר נתון בשדה חשמלי בעוצמה מסוימת. עצם הימצאותו של גביש קריסטל זה בשדה חשמלי גורמת לגביש להתרחב ולהתכווץ בתדירות מסוימת ובעצם בעת כל התרחבות והתכווצות הגביש פולט אות (מתח חשמלי) סינוסיאדלי. תכונה זו של קריסטלים נקראת פיאזו-אלקטריות והיא התגלתה בשנת 1894 על ידי המדען הצרפתי פייר קירי. תכונה זו גם עובדת בצורה  
הפוכה - על ידי לחיצה פיזית על הגביש ניתן לייצר אות חשמלי. מתנדי קריסטל מורכבים לרוב מגביש קוורץ (Quartz), מכיוון שהוא הקריסטל המפיק את האות היציב והעמיד ביותר ולכן הוא הנוח ביותר לשימוש בתעשייה[[5]](#footnote-5).

להלן תרשים המתאר את מבנה המתנד הגבישי, את סימולו החשמלי וכן את המעגל החשמלי השקול אליו (רלקסציה)

איור 3

מתנד רלקסציה - מתנד זה מבוסס על מעגל חשמלי המשלב נגד עם קבל (RC) או נגד עם סליל (RL) שבו משתמשים ליצירת אות ריבועי או אות "שן מסור". במתנדי רלקסציה משתמשים ליצירת אות שעון הדרוש למעגלים ספרתיים מסוימים, למשל מונים או טיימרים, אך מתנדים גבישיים עדיפים עליהם בגלל יציבותם.

בתעשיית השבבים, כאשר רוצים להשיג מתנד אפקטיבי ניתן לעשות זאת במספר טכנולוגיות: גביש קוורץ, מעגלי RC ומעגלי LC. האופציה הנפוצה ביותר לייצור רכיבי שעון היא גבישי קוורץ.

רכיב שעון (גבישי) כפי שהוא נראה בפועל:

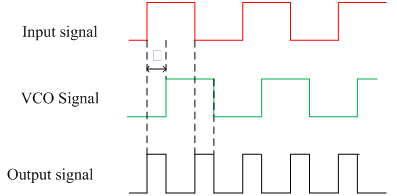
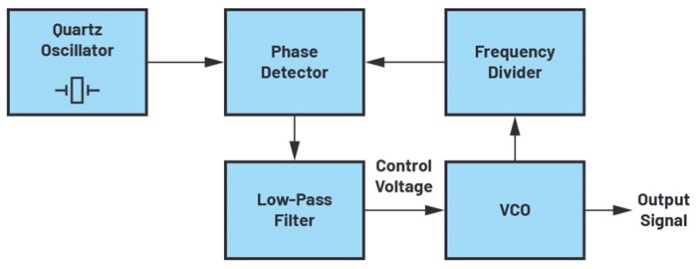
איור 4

מגבר - המגבר מחולק לשני חלקים שונים המגבירים אלמנטים שונים - מגבר מתח ומגבר תדירות. מכיוון שהאות המתקבל מהמתנד על כלל סוגיו הוא אות בעל מחזור שעון ארוך, המאפשר ביצוע של מעט פעולות בשניה יש צורך להגביר את תדירות האות כדי לקבל מתנד אפקטיבי יותר. בנוסף לכך המתנד מאפשר תנודה המוגבלת ע"י ריסון מסוים, כך שהתנודה דועכת אם לא מתחזקים אותה. לטובת משימה זו הוצמדו מגברים אלו למתנד, והם מגבירים את תדירותו ושומרים על מתח קבוע לכל מחזור שעון ובכך משלימים את פעולתו של השעון[[6]](#footnote-6).

מגבר התדרים הוא רכיב א-סינכרוני הנקרא Frequency Synthesizer הפועל על בסיס רכיב הנקרא PLL-Phase Lock Loop - לולאה נעולת שלב. רכיב זה, בהיותו א-סינכרוני מקבל את האות בתדירות נמוכה מצד אחד ומוציא את האות בצד שני בתדירות גבוהה. עקרון הפעולה עובד  
כך - בשלב הראשון האות מתקבל בPhase Detector- שדוגם את  
הפאזה - הסטייה ביחס לתחילת מחזור שעון. בשלב השני האות מועבר דרך מסנן תדרים נמוכים (Low Pass Filter) המסנן את התדרים הגבוהים. המתח המתקבל מה-LPF מפעיל את ה-VCO (מתנד מבוקר מתח) המשווה בין הפאזה הנוכחית לפאזה שהוא מפיק ובכך מגביר את תדירות האות היוצא[[7]](#footnote-7). להלן תרשים סכמתי ותצוגת גלים המתארים את תפקודו של ה-PLL.

איור 6

איור 5



מגבר המתח הוא מגבר שרת המחובר במעגל כמגבר יחידה, כלומר מתח המוצא (Output) שלו חוזר חזרה כמתח מבוא (Input) השלילי. בשל כך המגבר מקבל במבוא החיובי את מתח המקור (המגיע מהגביש) ובשלילי את המתח המתקבל מהמוצא. כך המתח המתקבל מהגביש אינו דועך לפי ריסון ואות המוצא שומר על יציבות בכל פעימה, בדרך לכלל הרכיבים במעגל הסינכרוני.

אופן הפעולה:

אות שעון - Clock Rate

כדי להבין את פעולת אות השעון בצורה מיטבית נסתכל על מחזוריות השעון (clock cycle). בכל מחזור שעון אות השעון מקבל עליית שעון (rising edge) לזמן מסוים ולאחר מכן ירידת שעון (falling edge) לזמן מסוים. כאשר המתח במעגל קרוב ל-5V אות השעון נמצא למעלה ומתקבל ערך לוגי '1' וכאשר המתח קרוב ל-0V אות השעון ירד למטה ויתקבל ערך לוגי '0'. [[8]](#footnote-8)

כדי להתחיל פעולות וכן לבצע תיאום ביניהן קיימות מספר דרכים שבהן השעון שולט בקצב הפעולות (חשוב לציין כי שינוי בין '0' ל-'1' לא מתבצע בהכרח ב-5V, כיום המתחים הרבה יותר נמוכים).

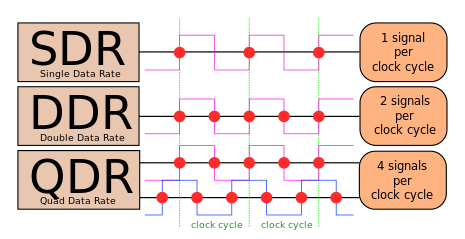
שליטת השעון בקצב הפעולות מתבצעת בשלושה אופנים שונים:

Single Data Rate (SDR) - ביצוע פעולה אחת בכל מחזור שעון - בירידת שעון או בעליית שעון אך לא בשתיהן יחד.

Double Data Rate (DDR) - ביצוע שתי פעולות בכל מחזור שעון - אחת בירידת שעון ואחת בעליית שעון.

Quad Data Rate (QDR) - סנכרון בין שני רכיבי שעון בעלי אותה תדירות בהפרשי פאזה קבועים בנוסף לביצוע כפול של פעולות שעון (DDR) - באופן זה יתאפשר ביצוע של ארבע פעולות בכל מחזור שעון.

שלושת האופנים הנ"ל מאפשרים לנו לסנכרן מספר רכיבי שעון בהפרשי פאזה שונים בין כל הרכיבים ולבצע פעולה כפולה בכל מחזור ובכך לקבל מספר גבוה בהרבה של פעולות שעון בכל מחזור כללי.

ניתן להשתמש בתצוגת גלים זו בכדי להיטיב להבין את הכתוב לעיל:

איור 7

פרק ב': בעיות סנכרון ופתרונות

בעיות הסנכרון בשעונים מתחלקות לשני חלקים עיקריים:

* בעיות סנכרון בשעון עצמו (פנימיות לרכיב).
* בעיות סנכרון בין שני שעונים (בין שני רכיבי שעון).

**בעיות הסנכרון הפנימיות לרכיב השעון**

בעיות אלו מתחלקות לשתיים:

* ריצוד שעון (Clock Jitter).
* הטיית שעון (Clock Skew).

ה-Clock Jitter הוא תכונה מולדת בכל שעון. ריצוד השעון הוא בעצם סטיה ממחזוריות השעון המחושבת (בעת תכנון המעגל), אשר חייבת להילקח בחשבון מכיוון שהיא קיימת **תמיד[[9]](#footnote-9)**. סטייה זו פוגעת ביכולתו של השעון לדייק בזמנים בהם הוא עובד ולתזמן באופן מיטבי את תחילתן של פעולות המעגל. הריצוד יכול להתרחש בעליית שעון (Rising edge) או בירידת שעון (Falling edge) כך שקיימים שני ריצודים -  
Transition Too early ו -Transition Too late .

הסיבות העיקריות להתרחשות ריצוד השעון אלו הפרעות אלקטרומגנטיות והעברת אותות לא רצויה בין ערוצי תקשורת (Crosstalk).

חשוב מאוד להתייחס ולהתחשב בריצודים בעת תכנון המעגל ובנייתו. אם נתעלם מקיום הריצוד במהלך התכנון יכול להיווצר מצב שהמעגל לא יפעל כראוי בגלל הריצוד, בשל כך יש צורך ביכולת מדידת ריצוד השעון, כלומר - יש צורך בקבלת ערך כמותי שנוכל לצמצם אותו ולהצהיר עליו עבור משתמשי הקצה. לכן באופן כללי נשאף להקטין את הריצוד למינימום, לקבלת שעון מדויק ביותר (יש יוצאים מן הכלל).

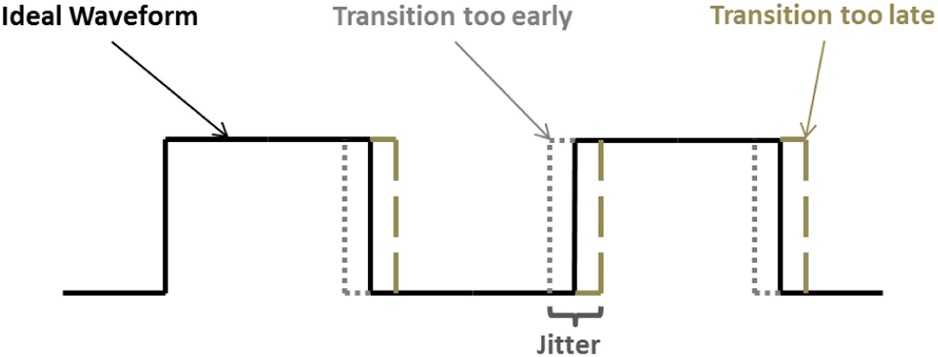
בריצוד שעון יש שלושה מדדים נפוצים:

ריצוד תקופתי (period jitter) - הוא הזמן בין כל תקופת שעון אחת לתקופת השעון האידיאלית או לתקופת השעון הממוצעת. ריצוד מחזור נמדד על ידי מדידת 10,000 מחזורי שעון ובדיקה כמה זמן הוא זמן מחזור מדוד אחד לעומת זמן המחזור האידיאלי.

ריצוד מחזור למחזור (cycle to cycle jitter) - זהו הפרש הזמן בין כל שני זמני מחזור סמוכים. ריצוד מחזור למחזור נמדד בעזרת מדידה של 1,000 מחזורי שעון ובדיקה מה ההפרש בין שני מחזורי שעון הסמוכים אחד לשני.

ריצוד שלב (phase jitter) - זה יחס של אות לרעש המחושב על ידי שילוב של רעש מופע השעון היחיד על פני מגוון רחב של תדרים. ריצוד זה קריטי ברכיבי FPGA כאשר ניתן להוריד את שיעור השגיאה בעזרת ריצוד מוגזם.

להלן תצוגת גלים הממחישה את הסטייה המתקבלת כתוצאה מריצוד:



איור 8

ה Clock Skew-או הטיית שעון - זוהי תופעה בה לוקח לאות משעון אחד זמנים שונים להגיע לרכיבים שונים במעגל האלקטרוני. ההבדל בין קריאות השעון נקרא ההטיה שלו[[10]](#footnote-10).

הטיית שעון נוצרת בעיקר ברכיבי זיכרון, כאשר אות השעון מגיע עם הטיה מסוימת לאוגרים (registers) ולדלגלגים (flip-flops). באופן אידיאלי, הקלט לכל רכיב זיכרון יגיע לערכו הסופי בזמן המתאים לעליית/ירידת השעון הבאה.

הטיית שעון יכולה להיגרם על ידי משתנים שונים, העיקריים שבהם: אורך החוטים המחברים בין השעון לרכיב הזיכרון, שינויי טמפרטורה, שינויים בהתקני ביניים, פגמים בחומר, והבדלים בקיבול הקלט בכניסות השעון של הרכיבים. ככל שתדר השעון של מעגל אלקטרוני גדל, כך התזמון בין רכיבי המעגל קריטי יותר והטיות השעון נהיות בעיתיות יותר אם רוצים שהמעגל יתפקד כראוי.

כאשר אנחנו מדברים על הטיית שעון, אנחנו יכולים לחלק את ההטיות לשני סוגים עיקריים: הטיה חיובית והטיה שלילית. ההטיה החיובית מתבצעת כאשר האוגר המקבל מידע מקבל את אות השעון מוקדם יותר מהאוגר הפולט את המידע. לעומת ההטיה החיובית יש לנו את ההטיה השלילית שהיא הפוכה בדיוק להטיה החיובית, בהטיה השלילית האוגר הפולט מידע מקבל את אות השעון לפני שהאוגר הקולט את המידע מקבל את אות השעון.

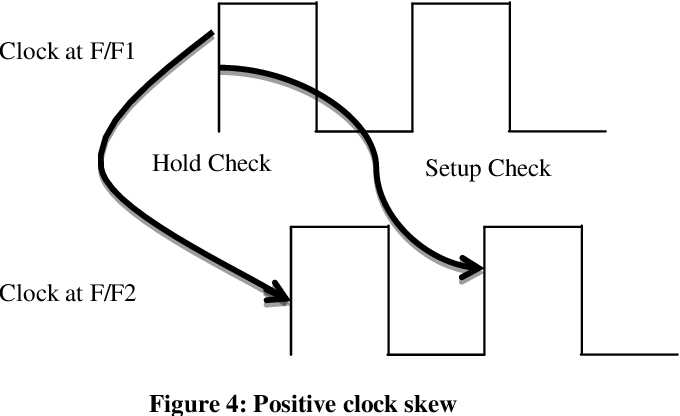
קיימות שתי הפרעות שיכולות להתרחש בעקבות הטיית השעון:

* הפרת זמן מינימלית (hold violation).
* הפרת זמן מקסימלית (setup violation).

ההפרעה הראשונה נגרמת כאשר אות השעון המגיע לאוגר הראשון מגיע לאט יותר לאוגר השני מאשר המידע שמוציא האוגר ראשון, או במילים אחרות, כאשר המידע הנפלט מהאוגר הראשון מגיע לאוגר השני, האוגר השני עדיין לא קיבל את אות השעון. הבעיה המשתמעת מכך היא שהנתונים יידרסו לפני שהאוגר השני ידע בכלל שהוא היה אמור לקרוא אותם על ידי הנתונים הבאים מהאוגר הראשון. הפרעה זו נקראת הפרת זמן מינימלית (hold violation) ובתרגום חופשי מאנגלית "הפרעת החזקה" והיא נקראת כך מכיוון שהמידע הקודם לא מוחזק מספיק זמן בכדי שאוגר היעד יקבל את אות השעון כראוי. הבעיה השנייה היא בדיוק הבעיה ההפוכה. הפרעה זו נגרמת כאשר האוגר השני מקבל את אות השעון לפני שאות השעון מגיע לאוגר הראשון שממנו מגיע המידע. למידע יש הרבה פחות זמן להגיע לאוגר היעד. הפרעה זו נקראת הפרת זמן מקסימלית (setup violation) ובתרגום חופשי מאנגלית "הפרעת איפוס" והיא נקראת כך מכיוון שהמידע החדש עדיין לא התאפס והתייצב לפני שעליית/ירידת השעון הבאה הגיעה.

חשוב לציין שמורכב יותר לפתור הפרת זמן מינימלית מאשר לפתור הפרת זמן מקסימלית מכיוון שאי אפשר לפתור אותה רק בעזרת הגדלת זמן המחזור.

להלן תרשים תצוגת גלים המתאר את סוגי ההטיות הנובעות מה-Skew:



איור 9

תפיסה מוטעית נפוצה בהקשר של הטיית שעון גבוהה מכוונת, היא שהטיה זו נתפסת בהכרח כפחות טובה ממצב שבו ההטייה היא קטנה יותר או שהיא דורשת שליטה מדויקת יותר בעיכובים בחלוקת השעונים. אך כאשר בודקים את העניין לעומק ניתן לראות שבמקרים מסוימים הטיית שעון זעירה תגרום להפרת זמן מינימלית, וכמו שכבר ציינו - קשה לפתור אותה. לעומת מקרים אחרים שבהם נגרום להטיה מכוונת והמעגל יהיה סבלני יותר לעיכובים לא מכוונים.

ניתן להתבלבל בין ריצוד שעון להטיית שעון, אך ריצוד שעון הוא בעצם אי-המחזוריות של השעון לעומת הטיה של השעון שבעצם משתנה מזמן מחזור אחד לשני, בגלל שינויים מקומיים שתלויים בזמן באספקת החשמל, בטמפרטורת הסביבה וברעש.

לכן במקרה רגיל של שליחת וקבלת רישומים במיקומים שונים, אין דרך ברורה להפריד בין אי-הוודאות הכוללת בתזמון השעון להטיה וריצוד. לכן לפעמים ישתמשו במושג הטיית שעון בכדי לתאר גם הטיית השעון וגם ריצוד השעון באופן משולב.

**בעיות סנכרון בין שעונים (CDC)**

מכיוון שלכל רכיב במעגל קיים שעון פנימי משלו, אשר פועל בקצב המתאים לו, יכולות להיווצר בעיות סנכרון בין רכיבים אשר פועלים בקצבי שעון שונים. לדוגמא - נניח ונרצה שה-CPU יתקשר עם ה-DDR יטען ממנו נתונים, יפענח, יחשב, ויכתוב בחזרה את הנתונים החדשים  
ל-DDR, אך מכיוון שלכל אחד מהרכיבים הנ"ל יש שעון שרץ בקצב שונה, יתכן ויתרחשו בעיות מידע (Data Hazards) ושלא נקבל את המידע שרצינו במקום הנכון בזמן הנכון[[11]](#footnote-11).

קיימות שלוש בעיות סנכרון עיקריות:

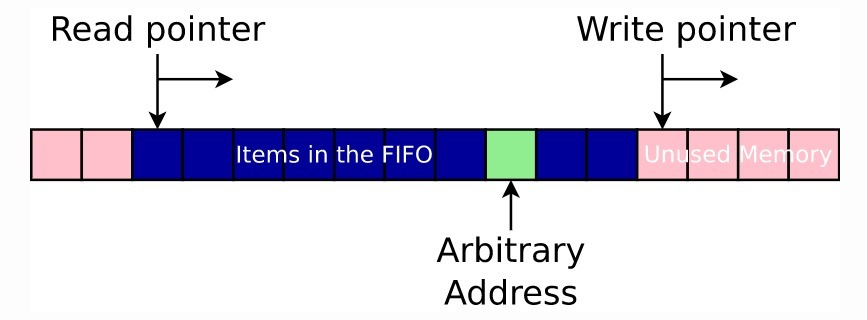
* Write After Write (WAW) - הרכיב שכותב את הנתונים רץ בקצב שעון מהיר יותר מהרכיב הקורא, ועקב כך יכול לכתוב נתונים חדשים אשר ידרסו את הכתיבה הקודמת, לפני שהרכיב הקורא הספיק לקרוא.
* Write After Read (WAR) - במקרה זה עבור שתי פקודות עוקבות, הרכיב ינסה להשתמש בנתונים שאמורים להיות מחושבים בפקודה העוקבת עבור הפקודה הנוכחית - כמובן שזה לא יעבוד כראוי ויתכן כי יתבצע שימוש בנתוני זבל.
* Read After Write (RAW)- מקרה זה דומה למקרה הקודם אך שונה מכיוון שבמקרה זה תהליך החישוב התחיל אך לא הסתיים לעומת WAR שבו תהליך החישוב עדיין לא התחיל.

קיימות מספר שיטות שיכולות לפתור בעיות אלו - נתמקד בשתיים מתוכן:

Bidirectional Handshake - שיטה זו מתאימה את הסטיות בין השעונים כך שתמיד יהיה אות שלא מפספס את הקריאה/כתיבה. היתרון שלו הוא שהוא תמיד עובד לכל יחס שעונים, לעומת זאת החיסרון שלו הוא שאין לו [[12]](#footnote-12)Pipelining.

Gray Fifo- שיטה זו דואגת שמיד לאחר כתיבת הנתונים תתבצע קריאה ע"י Read/Write Pointer שנכנס ויוצא ממחסנית המידע לפי הסדר ובכך אין דריסת נתונים או קריאת נתונים ישנים. את ה-Pointers האלו כותבים ע"י [[13]](#footnote-13)Gray Code .

להלן תמונה הממחישה את עקרון ה- Gray Fifo:



איור 10

רשימת מקורות

1. [מונה (מיתוג) – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%9E%D7%95%D7%A0%D7%94_(%D7%9E%D7%99%D7%AA%D7%95%D7%92))
2. [Pipeline (computing) - Wikipedia](https://en.wikipedia.org/wiki/Pipeline_(computing))
3. [פליפ פלופ – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%A4%D7%9C%D7%99%D7%A4_%D7%A4%D7%9C%D7%95%D7%A4)
4. [מתנד (אלקטרוניקה) – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%9E%D7%AA%D7%A0%D7%93_(%D7%90%D7%9C%D7%A7%D7%98%D7%A8%D7%95%D7%A0%D7%99%D7%A7%D7%94))
5. [מהו מתמר פיאזואלקטרי | מערכות מתקשרות (weizmann.ac.il)](https://stwww1.weizmann.ac.il/communication/?page_id=648)
6. [מגבר – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%9E%D7%92%D7%91%D7%A8)
7. [PLL Phase Locked Loop: How it Works » Electronics Notes (electronics-notes.com)](https://www.electronics-notes.com/articles/radio/pll-phase-locked-loop/tutorial-primer-basics.php)
8. [Clock rate - Wikipedia](https://en.wikipedia.org/wiki/Clock_rate)
9. [Clock Skew - an overview | ScienceDirect Topics](https://www.sciencedirect.com/topics/engineering/clock-skew)
10. [What is Clock Jitter, Why Does It Matter, and What Can You Do to Minimize It? (silabs.com)](https://www.silabs.com/community/blog.entry.html/2015/03/19/jitter_what_it_is-eVzF)
11. [What is Clock Jitter, Why Does It Matter, and What Can You Do to Minimize It? (silabs.com)](https://www.silabs.com/community/blog.entry.html/2015/03/19/jitter_what_it_is-eVzF)
12. [Clock domain crossing - Wikipedia](https://en.wikipedia.org/wiki/Clock_domain_crossing)
13. [cdc - gray code clock domain crossing FIFO fast to slow - Electrical Engineering Stack Exchange](https://electronics.stackexchange.com/questions/432094/gray-code-clock-domain-crossing-fifo-fast-to-slow)
14. [Mode 2—bi-directional I/O (tutorialspoint.com)](https://www.tutorialspoint.com/mode-2-bi-directional-i-o)

Abstract

The circuits we use to assemble electronic devices in general and computers in particular can be roughly divided into two groups: synchronous circuits and asynchronous circuits.

Asynchronous circuits are circuits with components that operate independently without a clock component scheduling their operations. Usually, these components work slower than synchronous circuits: each component starts operating only when the previous component has finished it’s task, even if theoretically, they can work in parallel. For this reason, the subsequent component depends on the leading component (No pipelining).

Synchronous circuits, on the other hand, are circuits that synchronize operations within the circuit which allows for parallel work (pipelining).

This kind of clock will emit a clock signal that reaches each component and begin its operation when the clock rises or falls. The clock signal will be graphically plotted as a Duty Cycle which when rising (Rising edge) will give a logical value of '1' and when falling (Falling edge) will give a logical '0' value.

The clock cycle is measured in units of time (usually ns) and the frequency of the clock is measured in units of hertz (Hz) usually kHz, MHz and GHz.

The clock component is basically a component capable of emitting an AC electric voltage that is graphically expressed as a Sin wave converted to a square wave. This component is usually composed of a crystal (Generally quartz) which is under electrical voltage. Crystals have a unique feature of being able to generate electrical voltage when they are under electrical voltage themselves, this is the basis for all crystal-based computer clocks. In addition to these clocks there are other clocks based on RC or LC circuits (circuits consisting of resistor and capacitor or inductor and capacitor).

Since the clocks are components that ultimately rely on the properties of an organic substance, the attempt to use them in order to produce an accurate product is **almost** absolutely successful. Nonetheless the properties of the material cause slight deviations that produce synchronization problems between the components of the circuit which manifest in difficulties reading and writing the components that are synchronized by the clock. Small problems create big problems.

These problems are divided into two: internal problems within the clock circuit and problems that arise in synchronization between clocks. The internal problems of the clock circuit are due to two main elements: clock skew and clock jitter. Clock skew is a deviation from the clock signal propagation rate to all the components that are dependent on the clock signal to commence their operation, whereas clock jitter is an internal deviation which produces a slight deviation in ascending or descending (or both).

The problems that result from synchronizing between two clock circuits are even more complex. When two components communicate with each other by virtue of their function (e.g., CPU communicates with DDR) and each of them runs at a different clock speed, it is necessary to synchronize the interactions between them or otherwise Overflow or Underflow problems can occur. I.e., problems in reading and writing data from one component to another, basically by overriding new data or using old data that disrupts the rate of operation of the whole device.

Because the clock signal rate determines the start of operation of each of the components in the circuit, end users have realized that the device can be forced to operate at a different clock frequency from the factory setting frequency and thus achieve faster and more effective results than the device states. This process is called Overclocking and if used correctly it can optimize the operation of the device. However, if used incorrectly (under extremely high frequencies) the device can overheat. It’s packaging cannot withstand the high heat emitted and it will probably catch fire.

Table of Contents

Abstract (Hebrew)א.......................................................

Chapter A - Designation, structure and mode of operation .....................................................................1

Designation .......................................................1

Clock structure ................................................ 2

Mode of operation ........................................... 4

Chapter B - Synchronization problems and their solution ...................................................................... 6

Internal Sync Problems ................................... 6

Clock Jitter ............................................ 6

Clock Skew ........................................... 7

External synchronization problems ................. 9

Data Hazard ........................................... 9

Troubleshooting .................................. 10

Source list................................................................. 11

Abstract (English) ....................................................... I

Jerusalem Collage of Engineering

Department of Electrical Engineering

**Computer Clock – Design & Role**

From:

Or Velvart

Asael Hadar

Seminar report

Jerusalem, Israel July, 2021

Jerusalem Collage of Engineering

Department of Electrical Engineering

**Computer Clock – Design & Role**

From:

Or Velvart

Asael Hadar

Seminar report

Jerusalem, Israel July, 2021

1. [מונה (מיתוג) – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%9E%D7%95%D7%A0%D7%94_(%D7%9E%D7%99%D7%AA%D7%95%D7%92)) [↑](#footnote-ref-1)
2. [Pipeline (computing) - Wikipedia](https://en.wikipedia.org/wiki/Pipeline_(computing)) [↑](#footnote-ref-2)
3. [פליפ פלופ – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%A4%D7%9C%D7%99%D7%A4_%D7%A4%D7%9C%D7%95%D7%A4) [↑](#footnote-ref-3)
4. [מתנד (אלקטרוניקה) – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%9E%D7%AA%D7%A0%D7%93_(%D7%90%D7%9C%D7%A7%D7%98%D7%A8%D7%95%D7%A0%D7%99%D7%A7%D7%94)) [↑](#footnote-ref-4)
5. [מהו מתמר פיאזואלקטרי | מערכות מתקשרות (weizmann.ac.il)](https://stwww1.weizmann.ac.il/communication/?page_id=648) [↑](#footnote-ref-5)
6. [מגבר – ויקיפדיה (wikipedia.org)](https://he.wikipedia.org/wiki/%D7%9E%D7%92%D7%91%D7%A8) [↑](#footnote-ref-6)
7. [PLL Phase Locked Loop: How it Works » Electronics Notes (electronics-notes.com)](https://www.electronics-notes.com/articles/radio/pll-phase-locked-loop/tutorial-primer-basics.php) [↑](#footnote-ref-7)
8. [Clock rate - Wikipedia](https://en.wikipedia.org/wiki/Clock_rate) [↑](#footnote-ref-8)
9. <https://www.silabs.com/community/blog.entry.html/2015/03/19/jitter_what_it_is-eVzF> [↑](#footnote-ref-9)
10. [Clock Skew - an overview | ScienceDirect Topics](https://www.sciencedirect.com/topics/engineering/clock-skew) [↑](#footnote-ref-10)
11. [Clock domain crossing - Wikipedia](https://en.wikipedia.org/wiki/Clock_domain_crossing) [↑](#footnote-ref-11)
12. [Mode 2—bi-directional I/O (tutorialspoint.com)](https://www.tutorialspoint.com/mode-2-bi-directional-i-o) [↑](#footnote-ref-12)
13. [cdc - gray code clock domain crossing FIFO fast to slow - Electrical Engineering Stack Exchange](https://electronics.stackexchange.com/questions/432094/gray-code-clock-domain-crossing-fifo-fast-to-slow) [↑](#footnote-ref-13)