הפקולטה להנדסה ע"ש איבי ואלדר פליישמן
אוניברסיטת תל אביב

בי"ס להנדסת חשמל

פרויקט מס': 22-1-1-2494

תכנית עבודה

שם הפרויקט: OCB

מבצעים:

שם: באסל מנצור ת.ז.: 208204859

שם: עדן חיאלד ת.ז.: 206525396

מקום ביצוע הפרויקט: TAU University

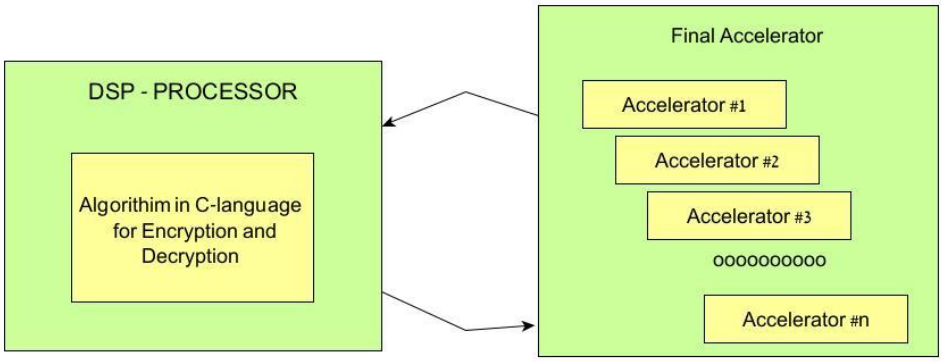
לשימוש המנחה:

הנני מאשר את תכנית העבודה המצורפת

שם: Oren Ganon חתימה: Oren Ganon

1. **תקציר:**

פרויקט זה הינו בתחום החומרה למערכות משובצות מחשב, בו נממש מצפין חומרה על מעבד DSP מתקדם של חברת CADENCE, כפי שמתואר בדיאגרמה הבאה:



כך שהמאיץ בנוי בשפת TIE שבעזרתה נוכל לבנות רכיבים חדשים שיכולים לבצע כמה פעולות באותו מחזור שעון למשל רכיב AddShift שיכול לבצע גם הזזה וגם הוספה במחזור שעון אחד במקום 2,בניית רכיבים כאלה שיכולים לבצע יותר מפעולה אחת באותו מחזור שעון יעזור לנו בלקצר משמעותית את זמן ריצת האלגוריתם, ולבסוף נבדוק את הנכונות שלו בסימולציות, הסינתזה וכמובן נבדוק האם מה שבנינו עובד חלק ולפי הדרישות על שבב Kintex-7 של חברת Xilinx.

2. **מוטיבציה:**

למה בכלל להריץ אלגוריתם הצפנה? ולמה כדאי להאיץ?

שאלות חשובות שהתשובה שלהן פשוטה ... ביטחון.

הצפנה הינה השיטה הטובה ביותר לשמור על אבטחת הנתונים כך שהיא מגינה על תוכן הקבצים ואף אחד שאין לו את ה"כלי" לבצע את הפיענוח המתאים לא יוכל להשתמש בהם, ותמיד עדיף להשתמש במאיצי חומרה לסיבה שאפילו יותר פשוטה ... מהירות, הגענו ל-2022 ולהשתמש בהצפנה של קבצים שתיקח שבוע , כמה ימים או אפילו כמה שעות לפעמים תפגע במיקום של החברה בשוק ותפסיד מול מתחרים.

להשתמש בתוכנה או חומרה לבד כדי לבצע את ההצפנה נותנת לנו או הצפנה "טובה" אבל בזמן מאוד ארוך שיכול להגיע לשבוע, או הצפנה גרועה אבל בזמן הצפנה מצוין, ומכאן הגיע הצורך לפתרון ביניים שמשלב פתרון חומרתי עם פתרון תוכנתי.

עוד שאלה מעניינת, האם יש מי שכבר עשה את זה?

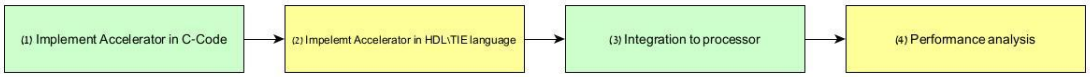
כן יש כמה חלופות למשל CBC and EBC methods .

אז למה OCB?

ECB הינה שיטת הצפנה על ידי חלוקה לבלוקים נפרדים אבל על ידי מפתח הצפנה אחד משותף לכל הבלוקים ,דבר שפוגע ברמת ההצפנה למשל נוכל להצפין תמונה כלשהיא והתמונה החדשה נבחין בצורה המקורית של התמונה למרות שלא רצינו, מצד שני יש לנו את הCBC שגם הוא משתמש בחלוקה לבלוקים אבל לכל בלוק יש לו מפתח הצפנה אחר ,החיסרון הוא שווקטור ההתחלה משודר יחד עם הקובץ המוצפן מה שיכול לגרום בסיבה העיקרית להצפנה... בטיחות.

3. תכולת עבודה:

לפני שנתחיל במימוש המאיץ נצרך להיעזר בכמה מאמרים, ספרים ואתרים שבעזרן נצבור את הידע הנדרש כדי לכתוב אלגוריתם OCB בשפת C שיבצע ההצפנה בצורה הכי יעילה שאפשר ,אחר כך נבדוק בעזרת הכלים של CADENCE את זמן ריצה האלגוריתם ונחקור את ה"תרגום" של האלגוריתם לשפת מכונה ונבנה בלוקים חדשים בשפת TIE שיבצעו את הפעולות שלוקחות הכי הרבה מחזורי שעון בכמה שפחות מחזורי שעון אחר כך נבדוק DRC ("כתיבה" נכונה לרכיבים החדשים), LVS (בתאמה בין קוד TIE לבין המבנה הרצוי), Routing ובדיקת סינתזה על ידי תוכנת Vivado של חברת Xilinx ,ולבסוף נצרוב את הbit file על שבב Kintex-7 של חברת Xilinx, ונבדוק אם הכל רץ חלק ולפי הציפיות שלנו, הדיאגרמה הבאה מסכמת את התוכנית :



- בחלק הראשון נממש את הידע שצברנו על אלגוריתם OCB ונכתוב את האלגוריתם הכי יעיל שאפשר נתחיל בכתיבה על visual studio ואז נעבור לכתוב בסביבת לינוקס.
- בחלק השני ניעזר בכלי XTENSA של חברת CADENCE כדי לחקור את הקוד שכתבנו ולבדוק כמה מחזורי שעון לוקח לתוכנה לרוץ וכמה פעמים מבצעים כל שורה ושורה באלגוריתם ולכמה שורות ממורות השורות שלוקחות הכי הרבה מחזורי שעון ואז לשורות האלו ננסה לבנות רכיב חדש בשפת TIE שבעזרתו נקבל אותה פונקציונאליות אבל בכמה שפחות מחזורי שעון.
- בשלב השלישי נחבר את המאיץ שכתבנו למעבד DSP של חברת CADENCE ונמיר אותו ל net list נבדוק חוקיות וסינתזה ,נעבור על כל הדוחות שנקבל (timing report, power report, utilization report) נבדוק גם את הclock tree ואחרי שנוודא שהכל תקין נעבור לimplementation ואז נבצע routing ונייצר bit file שאותו נוכל לצרוב על השבב (במילים אחרות profiling).
- בשלב הרביעי נצרוב את הקובץ שקיבלנו לשבב ונבדוק את הperformance שלו בזמן אמת.

רשימת מקורות שנשתמש בהן (סביר להניח שיהיו שינויים בהמשך):

- Embedded systems design and verification, edited by Richard Zurawski.
- Processor description languages .
- https://ip.cadence.com/uploads/980/TIP_WP_TIE_FINAL-pdf
- https://en.wikipedia.org/wiki/Profiling_(computer_programming)

4. תוצרי הפרויקט:

אחרי שנעבור את השלב השני בדיאגרמה 2 ונבדוק שהקוד נכתב בצורה נכונה ו"חוקית", נעבור לשלב היותר משמעותי, כפי שכבר תיארנו בחלק השלישי בדיאגרמה 2 נבדוק את הסינתזה ונעבור על כל הדוחות ונסתכל במיוחד על 2 דוחות הכי משמעותיות בעולם של VLSI ונדרוש עבור כל אחר דרישה שבסוף נשאף להגיע אליה והן :

- Timing report- בדוח הזה נבדוק כמה זמן (במחזורי שעון) לקח לרכיב שלנו לרוץ ולבצע את הכל ונדרוש שזמן הריצה שלנו יהיה כמה שפחות, ובמקרה הכי גרוע נקבל שיפור של 10% בין הרצה עם מאיץ להרצה בלי.
- Power report- בדוח הזה נבדוק כמה הספק צורך הרכיב וכמה הספק מבזבז ונדרוש שההספק שהתבזבז יהיה מינימלי כמה שאפשר , ובמקרה הכי גרוע לא תעבור את ה25% מההספק שמתבזבז על ידי המעבד עצמו.

דרישה מאוד משמעותית ומאוד חשובה הינה ה- utilization , פה מילת המפתח הינה "כסף", ובעולם הVLSI שטח שווה כסף, ולכן נשאף לממש את הרכיב שלנו בשטח הכי קטן שאפשר ,ובמקרה הכי גרוע שלא יעבור את 40% משטח המעבד עצמו.

כל הבדיקות האלו ועוד יבוצעו בהתחלה בעזרת סימולציות של תוכנת Vivado של חברת Xilinx ואז בסוף נוודא אם אכן כל הדרישות הנ"ל מתקיימות בזמן אמת על השבב.

5. לוח הזמנים:

אבן דרך	פירוט	תאריך יעד לביצוע
למידת האלגוריתם להצפנה	בשלב זה נחקור את החומר שאספנו ונלמד איך עובד אלגוריתם הצפנה OCB לעומת אלגוריתמים אחרים	25/11/2022
למידת מעבד INORDER - איך עובדת הצפנה שנעשית בתוכנה	בשלב זה נלמד איך עובד מעבד INORDER , ואיך עובד אלגוריתם ההצפנה על מעבד שבנוי בטכנולוגיה זו .	10/12/2022
למידת כלי פיתוח חומרה ולמידת FPGA	בשלב זה, נעבור על הdatasheet של השבב שבסופו של דבר נבדוק את המימוש שלנו דרכו, ונבחר בכלי הכי יעיל בכדי שנוכל לעשות זאת.	24/12/2022
ניתוח האלגוריתם וביצוע PROFILING לקוד	בשלב זה , אחרי שכתבנו את הקוד ,נשתמש בכלי העבודה של חברת CADENCE כדי להבחין בפקודות המשמעותיות ביותר לשיפור.	07/01/2023
כתיבת הצעה לשיפור החומרה על מנת להאיץ את הביצועים	אחרי שביצענו PROFILING , נכתוב הצעות לקוד חומרתי שבעזרתם נוכל לבנות את המאיץ הנדרש או החלקים שמרכיבים אותו.	14.01/2023
הגשת מצגת האמצע		22/01/2023
כתיבת תוכן חומרתי של המאיץ	אחרי שבדקנו כמה אופציות לכתיבת הקוד החומרתי נבחר את האופציה הכי טובה ונממש אותה.	15/02/2023
שילוב של המאיץ(מבחינה חומרתית) עם הקוד של האלגוריתם	אחרי שמימשנו את המאיץ ובדקנו אותו נשלב אותו לתוך האלגוריתם של ההצפנה.	01/03/2023
שילוב של הרכיב שמתקבל לתוך אינטגרציה של המעבד	בשלב זה, נשלב את המאיץ שכתבנו למעבד DSP של חברת CADENCE .	15/03/2023
הרצת ניתוח ביצועים (PROFILING) אחרי הוספת המאיץ	שוב נבצע בדיקות וPROFILING ל"בלוק" הסופי שכולל את המעבד ואת המאיץ.	19/04/2023
ניתוח הביצועים והשוואה בין קוד שרץ בלי מאיץ לקוד שרץ עם מאיץ	בשלב זה, נבדוק אם המאיץ שבנינו באמת מבצע את הדרישות על ידי בדיקת הביצועים של קוד שרץ של מעבד כולל מאיץ ובמעבד בלי.	01/05/2023
מעבר ל-FPGA ובדיקת ביצועים על חומרה אמיתית והשוואת ריצה על FPGA לסימולציה	לבסוף נצרוב את ה bit file שקיבלנו על שבב ה-FPGA של חברת Xilinx ונשווה בין תוצאות אמת לתוצאות הסימולציות שקיבלנו קודם.	19/05/2023
הגשת הפוסטר וסיום העבודה בפרויקט		28/05/2023
הגשת ספר הפרויקט ומצגת הסיום		29/06/2023