מכון טכנולוגי חולון
Holon Institute of Technology

הפקולטה להנדסת חשמל ואלקטרוניקה

דוח ביניים לפרויקט גמר – תואר ראשון

מימוש מיקרופרוסור על FPGA הכולל פריפריות DSP

Implementation of Microprocessor Includes DSP Peripherals

מאת:

אריאל אוחיון #206912354

ליאור ידגרוב #315852848

מנחה: שמואל מעודה

מכון טכנולוגי חולון
Holon Institute of Technology

תוכן עניינים

תקציר 4

רשימת איורים 5

רשימת קיצורים 6

פרק א' – מבוא לפרויקט 7

רקע 7

תיאור הבעיה 8

פרק ב' – ניתוח תיאורטי 9

סוגים של ארכיטקטורות מעבדים 9

ארכיטקטורת MIPS 11

- תכונות ומימוש Pipeline 12
- שגיאות Hazards במימוש Pipeline 14
- דרכי פתרון לשגיאות Hazards 14

הקדמה לעיבוד אותות ושימושים שלהם 15

- תכונות של מערכות 15
- מערכות LTI 15
- דגימת אותות אנאלוגיים 17
- ממירים DAC/ADC 20

מסננים 22

- מסננים אנאלוגיים (LPF,HPF) 22
- מסננים דיגיטליים (FIR) 25

תיאור הכלים והעזרים בסביבת הפיתוח 30

רכיב FPGA 30

- מבנה רכיב ה - FPGA 30
- שלבי פיתוח 31

שפות תיאור חומרה HDL 32

תוכנות Quartus ו-Modelsim 32

הכרטיסים שבחרנו לעבוד איתם 33

תוכנית עבודה להמשך 35

סיכונים להמשך הפרויקט 35

ביבליוגרפיה 35

מכון טכנולוגי חולון
Holon Institute of Technology

תקציר

הפרויקט עוסק בתכנון חומרה של מיקרופרוסור בארכיטקטורת MIPS 5 Stage Pipeline.
מימוש החומרה בפרויקט יבוצע באמצעות צריבה לרכיב FPGA.
תחילה נציג רקע כללי על ארכיטקטורות שונות של מיקרופרוסורים ובעיקר נרחיב על ארכיטקטורת MIPS.
נתאר רקע תיאורטי בנושאי העיבוד אותות ועל האופן שבו ניתן לשלב בין יחידות לעיבוד אות לרכיב FPGA.
נדבר על כלי העבודה של סביבת הפיתוח שיעזרו לנו בשלבי הפרויקט כגון : תוכנה לניהול גרסאות, כרטיסים אלקטרוניים, תוכנות וכלים לצרכי Debug וסימולציות.
נתאר את אופן העבודה עם רכיבי FPGA וההתמודדות עם בעיות.
בהמשך נתאר את המבנה הכללי של רכיב ה- FPGA שבחרנו לעבוד אתו בפרויקט ושלבי הפיתוח שביצענו.
מטרת הפרויקט היא לאפשר ממשקי DSP למיקרופרוסור על מנת לפתור בעיות בתחום העיבוד אותות באופן יעיל ופשוט.

Abstract

The project based on hardware design of MIPS 5 Stage Pipeline Microprocessor.
The hardware will be implemented on FPGA component.
At first we will show some background on microprocessor architecture and specially the structure of the MIPS architecture.
We will describe theoretical info of DSP and how to interface the FPGA with the DSP components.
Also we will talk about the development environment tools that will help us to implement the project: version management software, Electronic Cards, software and debug tools and simulations.
We will describe how to work with FPGA components and deal with problems.
Then, we will present the specific FPGA component that we chose for this project, describe the development stages.
Our goal is to allow a DSP interfaces for microprocessor in order to analyze and solve signal processing problems.

מכון טכנולוגי חולון
Holon Institute of Technology

רשימת איורים

איור 1 - הכרטיס האלקטרוני במכשיר 7
איור 2 - המכשיר Speak & Spell 7
איור 3 - דוגמא לשימוש בעיבוד תמונה 8
איור 4 - מבנה ארכיטקטורת Von Neuman 9
איור 5 - ארכיטקטורת Harvard 10
איור 6 - Instruction Type Format 11
איור 7 - MIPS 5Stage Pipeline 12
איור 8 - סדר ביצוע הוראות בארכיטקטורה מסוג Pipeline 13
איור 9 - מבנה למערכת המבצעת דגימה 18
איור 10 - דוגמא לשכפול הספקטרום כתוצאה מדגימה 18
איור 11 –דוגמא לספקטרום של אות 18
איור 12 - דוגמא לספקטרום של אות שנדגם מתחת לתדר Nyquist 19
איור 13 - דוגמה לספקטרום של אות שנדגם בדיוק בתדר Nyquist 19
איור 14 - מעגל חשמלי הממשמש ממיר DAC מסוג נגדים משוקללים בינארית 20
איור 15 - סכמת בלוקים ש מערכת הממשמשת ממיר ADC מסוג Successive Approximation 21
איור 16 - דוגמא לתגובת תדר של מסנן LPF 22
איור 17 - דוגמא לתגובת תדר של מסנן HPF 22
איור 18 - תגובת תדר של מסנן LPF בתדר קטעון 4[Hz] 23
איור 19 - אות הכניסה למסנן המכיל שני הרמוניות במישור התדר 23
איור 20 - אות כניסה למסנן המכיל 2 הרמוניות 23
איור 21 - ספקטרום של אות המוצא מהמסנן 24
איור 22 - אות המוצא כפונקציה של הזמן 24
איור 23 - דוגמה לאות מוצא מה-FIR 25
איור 24 - דוגמה לאות הכניסה של ה-FIR 25
איור 25 - סכמה המתארת את המבנה של מסנן FIR 26
איור 26 - מסנן LPF עם תדר קטעון 4[Hz] 27
איור 27 - פונקציית Sinc 27
איור 28 - פונקציית Sinc לאחר ביצוע הזזה 28
איור 29 - פונקציית Sinc לאחר דגימות 28
איור 30 - המבנה הלוגי של CLB 30
איור 31 - מבנה כללי של רכיב ה-FPGA 31
איור 32 - הכרטיס DE2-115 34

מכון טכנולוגי חולון
Holon Institute of Technology

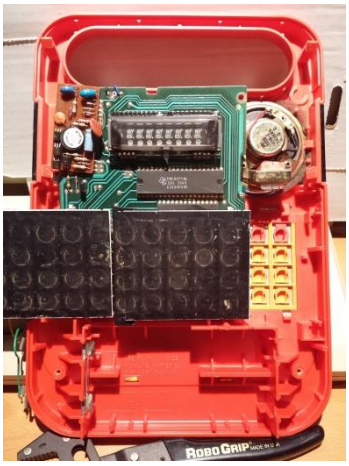
רשימת קיצורים

FPGA	Field Programmable Gate Array
MIPS	Microprocessor without Interlocked Pipeline Stages
RISC	Reduced Instruction Set Computing
R-Type	Register Type Instruction
I-Type	Immediate Type Instruction
J-Type	Jump Type Instruction
IF	Instruction Fetch
ID	Instruction Decode
EX	Execute
MEM	Memory
WB	Write Back
PC	Program Counter
ALU	Arithmetic Logic Unit
LTI	Linear Time Invariant
ADC	Analog Digital Converter
DAC	Digital Analog Converter
SAR	Successive Approximation
EOC	End Of Conversion
LPF	Low Pass Filter
HPF	High Pass Filter
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
LE	Logic Element
CLB	Configurable Logic Block
MUX	Multiplexer
LUT	Look Up Table
FF	Flip Flop
CPLD	Complex Programmable Logic Device
HDL	Hardware Description Language
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
PLL	Phase Locked Loop
DSP	Digital Signal Processing
RTL	Register Transfer Level
EEPROM	Electrically Erase Programmable Read Only Memory
SRAM	Static Random Access Memory
LCD	Liquid Crystal Display
RGB	Red Green Blue
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

פרק א' – מבוא לפרויקט

רקע

בשנת 1976, פותחה מערכת עיבוד אותות הראשונה על ידי חברת Texas Instruments הנקראת Speak & Spell.
מערכת זו פותחה כמשחק ילדים אך שימשה כמהפכה טכנולוגית בעולם עיבוד האותות.
המערכת מימשה משחק איות של מילים ואותיות באנגלית, על ידי ממשק משתמש של מקלדת ורמקול המחוברים לכרטיס אלקטרוני שמכיל בקר (TMS5100).
הבקר בכרטיס הכיל סט פקודות שאפשרו ביצוע עיבוד אותות שמע (פיתוח אות שמע מהרמוניות של אותות סינוסואידליים) .



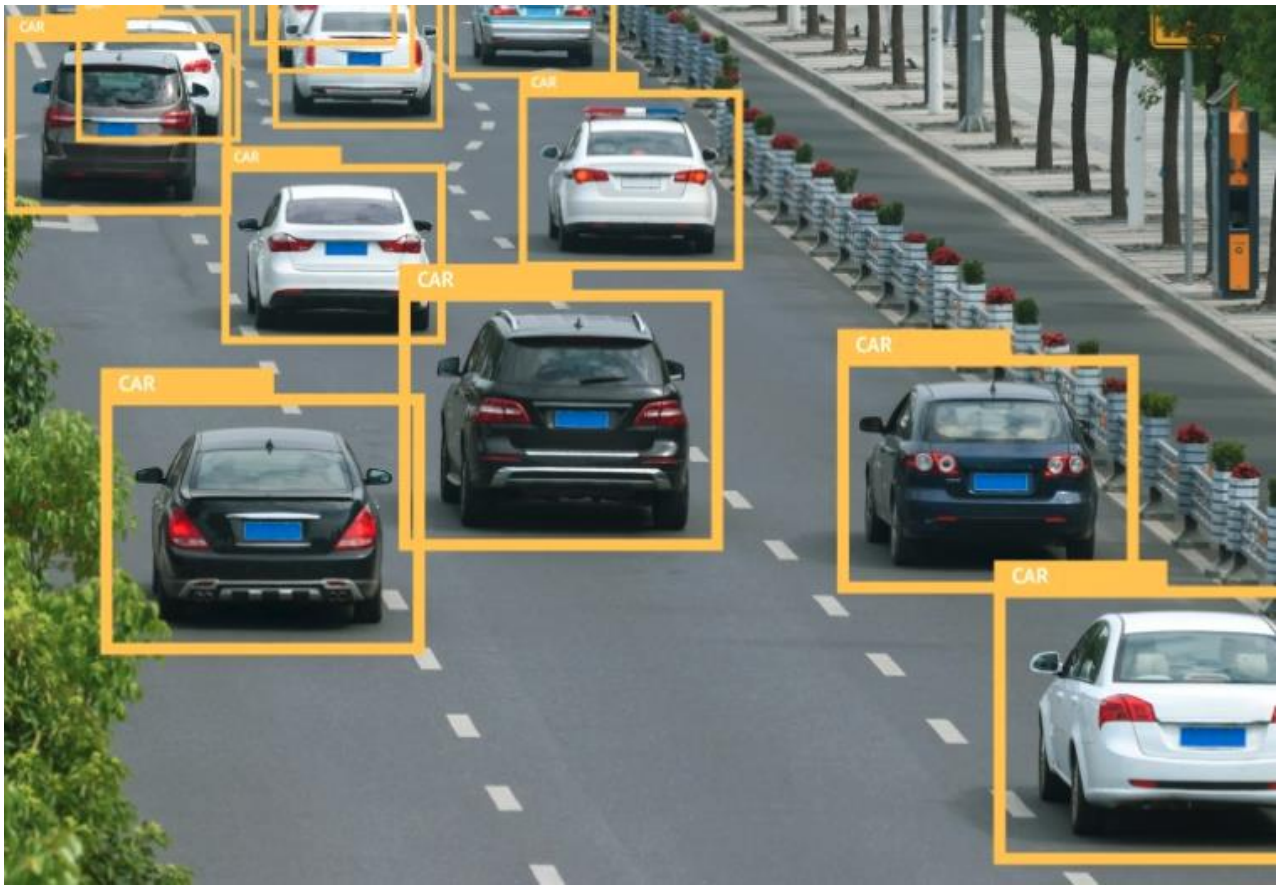
איור 1 - הכרטיס האלקטרוני במכשיר

איור 2 - המכשיר Speak & Spell

מאז, עולם האלקטרוניקה נחשף יותר ויותר למעבדים המאפשרים:

1. פעולות מתמטיות מורכבות.
2. קריאה וכתיבה של נתונים אנאלוגיים על ידי ממירים לתחום האנאלוגי.
3. מימוש מסננים במעבדים.

המטרה העיקרית היא לייעל את החומרה והתוכנה של המעבדים, כדי לעבד את הנתונים המתקבלים מהאותות האנאלוגיים הנקלטים מהסביבה שמחייבים פענוח מהיר (לדוגמא: אותות שמע, אותות וידאו, אותות תקשורת).
פיתוח מערכות בתחום עיבוד האותות מהווה אחד מהגורמים העיקריים בהתקדמות הטכנולוגית.
כיום, בעולם המחשוב ניתן לבצע עיבוד תמונה לזיהוי עצמים, עיבוד אותות קול (סינתזה – יצירה של קולות חדשים, וניתוח ספקטרלי של אותות שעברו דגימה), קידוד ופענוח של אותות תקשורת המועברים בכבל ובצורה אלחוטית, ועוד.



איור 3 - דוגמא לשימוש בעיבוד תמונה

המטרה שלנו בפרויקט היא לייצר ארכיטקטורה לחומרה של מעבד אשר יכלול בתוכו סט פקודות שיאפשרו מימושים בסיסיים ליישומים בתחום עיבוד אותות.

תיאור הבעיה

הטכנולוגיה הנוכחית דורשת ממהנדסים בתחום החשמל והאלקטרוניקה לחשוב על פתרונות במגוון תחומי החיים לצורך שיפור איכות החיים.

הנושא המרכזי סביבו מתמקדים מהנדסים ואנשי פיתוח רבים הוא כיצד ניתן לייצר מערכות המשלבות חומרה ותוכנה על מנת לפתור בעיות בחיי היום יום וכיצד ניתן לקדם תחומים כמו תקשורת, רפואה, תחבורה ועוד לכיוון של אוטומציה והתייעלות באמצעות כלי הפיתוח.

הרכיבים האלקטרוניים המסוגלים לפתור בעיות מסוג זה על ידי עיבוד נתונים מהיר הם רכיבים מתוכנתים.

תחום הרכיבים המתוכנתים מחולק למיקרובקרים ורכיבי FPGA.

היתרון העיקרי של רכיבי FPGA על פני המיקרובקרים הוא שניתן להגדיר חומרה לרכיב FPGA ולצרוב אותה עליו, הדבר מאפשר לעבוד עם סט שעונים מהיר. במיקרובקר החומרה קיימת כבר ברכיב ומוגדרת לתדר שעון מקסימלי (שהוא נמוך יותר מתדר השעון שניתן לעבוד עליו ברכיב FPGA).

לכן, בחרנו לתכנן את המערכת שלנו בפרויקט על רכיב FPGA.

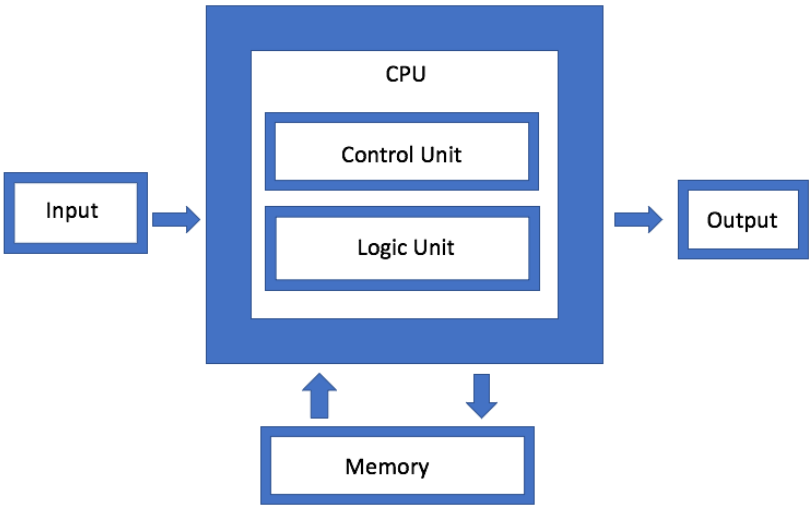
פרק ב' – ניתוח תיאורטי

סוגים של ארכיטקטורות מעבדים

ניתן לסווג את הארכיטקטורות השונות לפי פרמטרים שונים התלויים במימוש כמו : סט ההוראות של המיקרופרוסור, אופן הקידוד של ההוראות, מבנה הזיכרון.
בסיכום זה נסווג את הארכיטקטורות לפי מבנה הזיכרון.

ארכיטקטורת Von Neumann:

הארכיטקטורה בנויה באופן כזה שהזיכרון הראשי שהמעבד עובד אתו כולל פקודות שהתוכנית מריצה (Instruction Memory) ואת הנתונים שהמעבד קורא\כותב לזיכרון במהלך ריצת התוכנה (Data Memory).
בעיה נפוצה של שימוש בשיטת תכנון לפי ארכיטקטורה כזו היא "צוואר בקבוק הפון נוימן".
בעיה זו מתארת התנגשות בין קריאה\כתיבה לזיכרון הנתונים במהלך התוכנית, לבין שליפת הפקודה הבאה לביצוע מהזיכרון.
אלו שתי פעולות שהמעבד מבצע אותן עבור כל הוראה שכתובה בתוכנית ויפורטו בפרק של תכונות ומימוש Pipeline.
בעיה נוספת חמורה עוד יותר בארכיטקטורה זו, היא שהמעבד מסוגל לרשום נתוני מידע שהוא משתמש בהם במהלך התוכנית על כתובות בזיכרון ששמורות עבור הפקודות שהמעבד צריך לבצע.



איור 4 - מבנה ארכיטקטורת Von Neuman

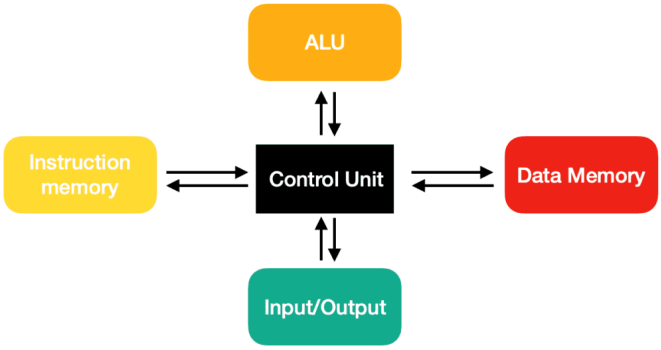
ארכיטקטורת Harvard:

הארכיטקטורה בנויה בצורה בה קיימים שני רכיבי זיכרון

1. מיועד לשמירת ההוראות לביצוע (Instruction Memory).
2. משמש לצורך שמירת הנתונים במהלך התוכנית (Data Memory).

ארכיטקטורה זו פותרת את הבעיות שנוצרו משימוש בזיכרון בודד גם להוראות וגם לנתונים אך דורשת יותר משאבים לצורך תכנון שני מודולים של זיכרון בארכיטקטורה.

שיטת עבודה בארכיטקטורה זו מאפשרת כבר את העבודה הבסיסית עם מיקרופרוסורים שמיושם בהם Pipeline.



איור 5 - ארכיטקטורת Harvard

ארכיטקטורת MIPS

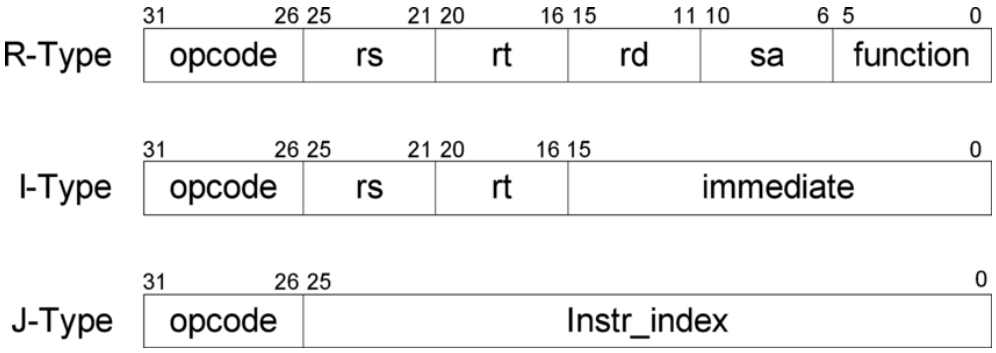
ארכיטקטורת MIPS (Microprocessor without Interlocked Pipeline Stages) היא ארכיטקטורה המממשת סט פקודות למעבדים מסוג RISC.

הנתונים במעבד מורכבים מ-32 סיביות והוא מומש בשני סוגים של ארכיטקטורות שכל אחת מהן עובדת בצורה שונה:

1. Single Cycle Architecture – ארכיטקטורה מסוג זה הייתה מקבלת פקודה ומבצעת רק אותה עד שהייתה מסתיימת ורק אז שולפת את הפקודה הבאה מזיכרון התוכנית.
2. Pipelined Architecture – ארכיטקטורה מסוג זה בנויה באופן שבו ביצוע פקודה מפורק ל-5 שלבים (נקראת גם 5 Stage Pipeline) בזמן שחלק מהחומרה בארכיטקטורה מבצע שלב מסוים בביצוע הפקודה, חלקים אחרים בחומרה מבצעים שלבים אחרים עבור הפקודות הבאות שהמעבד צריך לבצע. ארכיטקטורה זו יעילה יותר מה-Single Cycle ולכן בפרויקט החלטנו לממש אותה.

סט הפקודות המוגדר עבור ארכיטקטורת MIPS מורכב משלושה סוגים של הוראות:

1. R-Type Instruction: הוראה שמתבצעת על שני אוגרים (Registers) ומכניסה את התוצאה לאוגר נוסף.
2. I-Type Instruction: הוראה שמבצעת פעולה על אוגר וערך מידי שקיים כבר בקידוד של הפקודה ומכניסה את התוצאה לאוגר נוסף.
3. J-Type Instruction: הוראה שמבצעת בדיקה לתנאי מסוים, אם התנאי מתקיים התוכנית מבצעת קפיצה להוראה אחרת בתוכנית שלא לפי הסדר.

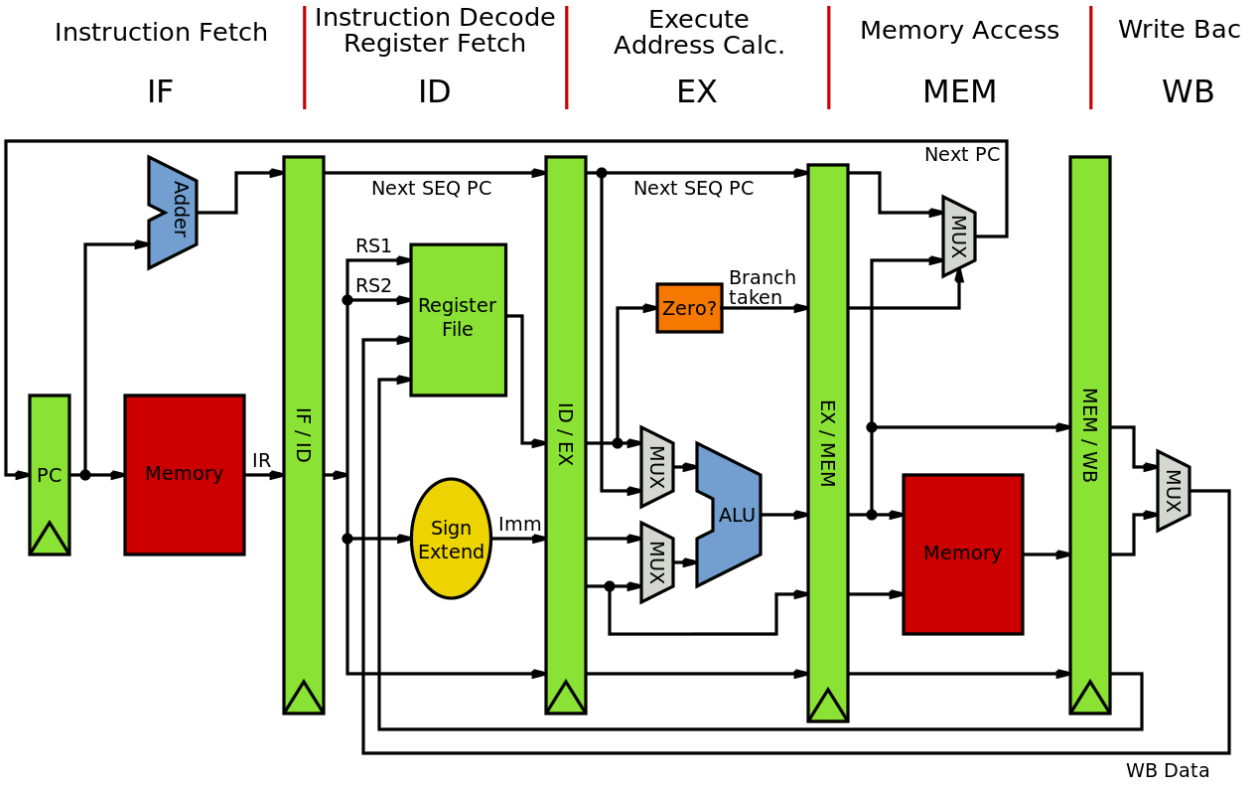


איור 6 - Instruction Type Format

- **תכונות ומימוש Pipeline**

כפי שכבר צוין בפרויקט נרצה לממש ארכיטקטורת 5 Stage Pipeline בה כל הוראה בתוכנית עוברת 5 שלבים.
לכל שלב יש חומרה משלו שמבצע את אותו השלב בביצוע הפקודה, בין השלבים יש חוצץ שאוגר את הנתונים של השלב הקודם ומעביר אותו לשלב הבא.
על ידי החוצצים בין השלבים ניתן לסנכרן את הפעולות שמתבצעות בין כל שלב בביצוע הפקודה.

ניתן לראות באיור מטה את ארכיטקטורת 5 Stage Pipeline -MIPS



איור 7 - MIPS 5Stage Pipeline

שלבי ביצוע של ההוראות:

1. שלב Instruction Fetch (שליפת הוראה) – בשלב זה המיקרופרוססור מוציא הוראה מזיכרון התוכנית לאוגר IF/ID.

   החומרה הדרושה לביצוע פעולה זו היא אוגר שיצביע על הכתובת בזיכרון בו נמצאת הפקודה שרוצים לבצע (PC), זיכרון התוכנית (Instruction Memory) ומחבר שמוצאו יחובר לאוגר PC כדי להעלות את ערכו בערך המתאים לכתובת בה נמצאת ההוראה הבאה בזיכרון התוכנית.

2. שלב Instruction Decode (פענוח הוראה) – בשלב זה המיקרופרוסיסור מחלק את סיביות ההוראה לשתי צורות עיקריות של פענוח: האחת, שידע לאיזה אוגרים ההוראה מופנית, והשנייה שיבדוק אם קיימים בהוראה ערכים מיידים שצריך לבצע איתם פעולה.
   החומרה הדרושה לביצוע שלב זה היא האוגרים של המעבד.
3. שלב Execute (ביצוע) – בשלב זה החומרה הדרושה היא ALU, יודעת לעשות את הפעולות המתמטיות (אריתמטיות ולוגיות) שמוגדרות לה בסט ההוראות, ובנוסף זו יחידה שיודעת לבצע את הבדיקות הדרושות לצורך קפיצות בתוכנית (בדיקת תנאים).
4. שלב Memory (זיכרון נתונים) – בשלב זה המעבד כותב לזיכרון הנתונים (Data Memory) את התוצאה שיצאה מהשלב הקודם (מהיחידה של ה-ALU).
   שלב זה אופציונלי שהרי לא תמיד כל נתון אנו כותבים בזיכרון הנתונים.
5. שלב Write Back (כתיבה בחזרה - משוב) – בשלב זה המעבד מבצע השמה של התוצאה שהתקבלה מהשלב השלישי ב-Pipeline באחד מהאוגרים של המעבד.
   גם שלב זה אופציונלי שהרי לא תמיד כל נתון אנו כותבים בחזרה לאוגרי המעבד.

החוצצים בארכיטקטורה מסוג Pipeline מאפשרים לנו לבצע את הפקודות בתוכנית באופן יעיל כך שכאשר הוראה נכנסת לשלב השני (Instruction Decode) ההוראה הבאה אחריה נכנסת לשלב הראשון (Instruction Fetch), וכשאותה הוראה עוברת לשלב השלישי (Execute) ההוראה הבאה אחריה נכנסת לשלב השני וההוראה שתבוא אחריה תיכנס לשלב הראשון.

איור 8 - סדר ביצוע הוראות בארכיטקטורה מסוג Pipeline

לכן, שיטה זו נחשבת לשיטה יעילה בתכנון מיקרופרוסיסורים אך היא כוללת כמה בעיות שעלינו להתגבר עליהם.

- **שגיאות Hazards במימוש Pipeline**

שגיאה מסוג Hazard מתרחשת כאשר הוראה שנמצאת באחד משלבי הביצוע תלויה בתוצאה של הוראה שקדמה לה ועדיין לא הסתיימה.

קיימים שלושה סוגים של שגיאות מסוג זה :

Data Hazards

כאשר הוראה שנמצאת באמצע אחד השלבים ב-Pipeline צריכה להשתמש בתוצאה של ההוראה שקדמה לה ונמצאת גם כן באחד משלבי הביצוע ועדיין לא הסתיימה.

לדוגמה :

```
addi $t1,$t2,5
addi $t3,$t1,1
```

אם ההוראה הראשונה נמצאת בשלב MEM (כתיבה לזיכרון הנתונים), אז ההוראה שאחריה נמצאת בשלב Ex (ביצוע הפקודה על ידי ה-ALU), והתוצאה לא יכולה להיכנס לאוגר $t3 עד שלא יתבצע השלב WB (כתיבה לאוגר $t1) בהוראה הראשונה.

שגיאות מסוג זה עלולות לבצע פעולות לא רצויות והשמת ערכים שגויים שהמתכנת לא התכוון אליהם.

Control Hazards

שגיאה זו מתרחשת כאשר קיימת בתוכנית הוראה שבודקת תנאי לקפיצה בתוכנית, התנאי נבדק בשלב השלישי בPipeline – שלב ה- Ex (בALU). מכיוון שהבדיקה מתבצעת רק בשלב השלישי, בינתיים נכנסות 2 ההוראות הבאות אחריה לפי הסדר שמוגדר ב-Pipeline ולא מתבצעת קפיצה בתוכנית כפי שתוכנן על ידי המתכנת.

Structural Hazards

שגיאה כזו מתרחשת כאשר יש פנייה לזיכרון הנתונים, תוך כדי שהמעבד מנסה לשלוף את ההוראה הבאה לביצוע, מתוך אותו הזיכרון.

שגיאה זו מתרחשת רק בארכיטקטורות מסוג Von Neumann.

בפרויקט זה אנו מתכננים את המעבד בארכיטקטורת Harvard (עם שני מודולים של זיכרונות) לכן אנו לא צפויים להיתקל בשגיאות מסוג זה.

- **דרכי פתרון לשגיאות Hazards**

1. Forwarding – בהצעת פתרון זו כאשר אנו מזהים שהמערכת נכנסת לשגיאה מסוג זה, אנו מושכים את התוצאה משלב הביצוע (Ex) מה-ALU, לשלב המתאים בה נמצאת הפקודה הבאה שיש לבצע.
2. Stalling – בהצעת פתרון זו החומרה בודקת את התלות בין הוראות ועושה השהייה מתאימה עד לסיום הפקודה כך שנגיע לפקודה הבאה עם התוצאות הרלוונטיות.

הקדמה לעיבוד אותות ושימושים שלהם

תחום העיבוד אותות מתאר מודלים מתמטיים לצורך ניתוח אותות אקראיים/דטרמיניסטיים המועברים בין מערכות חשמל ורכיבים אלקטרוניים. הפרמטרים שמאפיינים אותות אלו הם: משרעת, תדר, פאזה.
המערכת מקבלת אותות חשמליים בכניסה ומוציאה אותות חשמליים במוצאה בהתאם לאופי פעולתה.
גם מערכות חשמליות/אלקטרוניות נהוג למדל על ידי נוסחאות מתמטיות, לצורך פישוט הניתוח של מערכות מורכבות ואותות בין המערכות.

- ## תכונות של מערכות

יציבות: מוצא המערכת לא יקבל ערך אינסופי בתגובה להלם סופית.
סיבתיות: תלות של המערכת רק בהווה או בעבר, תלות בעתיד זה לא פיזיקלי(יוצר תדרים שליליים).
קביעות בזמן: חסין להשהיות, אם יש אות בכניסה, הזזה בזמן לא תשפיע על ערך שאמור להיות במוצא.
לינאריות: מקיים הומוגניות ואדטיביות, ניתן להכפיל בפרמטרים ולהכניס כמה אותות ולקבל במוצא כל אות בנפרד עם הפרמטר שהכפלנו.
זיכרון: בכל מערכת קיים בארגומנט המוצא תלות בזמנים שונים מההווה
הופכית: ניתן למצוא כניסה באמצעות מוצא.
פונקציות תמסורת: הצגה מתמטית (העתקה מציר הזמן לציר התדר) שמאפשרת לחקור פונקציה באופן הרבה יותר ברור ונוח.
פונקציית התמסורת הינה פרמטר שמייצג את היחס בין המוצא לכניסה במישור התדר.
*מערכת LTI מאפשרת להשתמש בפעולת קונבולוציה בין כל רכיב.

- ## מערכות LTI

נרצה לעבוד בDSP עם מערכות LTI לצורך פשטות החישובים והורדת השגיאות.
מערכות LTI מבטיחות את שני התכונות: לינאריות וקביעות בזמן.

<u>קונבולוציה</u>

קונבולוציה הינה פעולה מתמטית שמבצעת סכום של כפל בין אות לבין שיקוף של אות אחר.
פעולה זו מפשטת לנו את מוצא מערכת LTI בהינתן הכניסה ופונקציית התמסורת שלה.

נוסחת הקונבולוציה לאות רציף

$$f(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) \cdot h(t-\tau) d\tau$$

נוסחת הקונבולוציה לאות בדיד

$$f[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$

בטבלה למטה נציג את התכונות של פעולת הקונבולוציה:

תכונה	נוסחה
קומוטטיביות	$f(t) * g(t) = g(t) * f(t)$
אסוציאטיביות	$f(t) * [g(t) * h(t)] = [f(t) * g(t)] * h(t)$
הכפלה בסקלר	$\alpha \cdot [f(t) * g(t)] = \big(\alpha \cdot f(t)\big) * g(t) = f(t) * (\alpha \cdot g(t))$
דיסטריבוטיביות	$f(t) * \big(g(t) + h(t)\big) = f(t) * g(t) + f(t) * h(t)$
קומוטטיביות בגזירה	$\frac{df}{dt} * g(t) = \frac{d}{dt}[f(t) * g(t)] = f(t) * \frac{dg}{dt}$

<u>מעבר בין מימדים</u>

מעבר בין מימדים בתורת המידע והעברתו הינו חלק מאוד קריטי בעיבוד אותות.

באופן אינטואיטיבי קל לנו לחשוב ולנסות לחקור אותות חשמליים במימד הזמן, עם זאת, מימד הזמן בדרך כלל לא נותן לנו מספיק מידע על האות כדי שנוכל להוציא ממנו את כל הנתונים שאנו צריכים לצורכי עיבוד אות.

לכן, ישנו תהליך של מעבר בין מימדים מזמן לתדר, Z ואומגה.

מימד התדר: המימד הפופולרי ביותר בתחום התקשורת ועיבוד האותות בשל הצורך במניפולציה על רכיבי תדר. מימד זה מציג את עוצמת האות כפונקציה של התדר.

הנוסחה הבסיסית שמאפשרת לנו מעבר ממימד הזמן למימד התדר היא התמרת פורייה (היא מהווה מקרה פרטי להתמרת לפלס, אשר מתייחסת רק לרכיב של האות המורכב).

התמרת פורייה:

$$F(j\omega) = \mathcal{F}\{f(t)\} = \int_{t=-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt$$

על ידי קבלת מידע וניתוחים של התדרים, מאפשר לעשות עליהם מניפולציות שונות.

מימד Z: מאפשר להציג מערכות עם אותות סופיים/אינסופיים בעלות יציבות/חוסר יציבות כאחד. זה מאפשר ניתוח של אותות בטווח רחב של משכי זמן.

מימד Z מאפשר לנו לעשות ניתוח של אותות ומערכות בדידות בדומה לניתוח של אותות ומערכות רציפות שנעשה על ידי התמרת פורייה.

התמרת Z:

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n] \cdot z^{-n}$$

בנוסף, יותר קל לבצע ניתוח מתמטי למערכת LTI במישור התדר הודות לתכונות של ההתמרות.

בין התכונות שיכולות לעזור לנו לצורך ניתוח מערכות הם:

- הזזה בתדר: על ידי הכפלת אות מסוים באות $\cos(\omega_0 t)$ ניתן להזיז את התדר של האות בתדר של האות cos

$$\mathcal{F}\{x(t)\cdot\cos(\omega_0 t)\} = \frac{X\big(j(\omega-\omega_0)\big)+X(j(\omega+\omega_0))}{2}$$

- הכפלה בתדר: כאשר מבצעים התמרה על קונבולוציה בין שני אותות מקבלים את המכפלה שלהם במישור התדר.

$$\mathcal{F}\{x(t)*h(t)\} = X(j\omega)\cdot H(j\omega)$$

• דגימת אותות אנאלוגיים

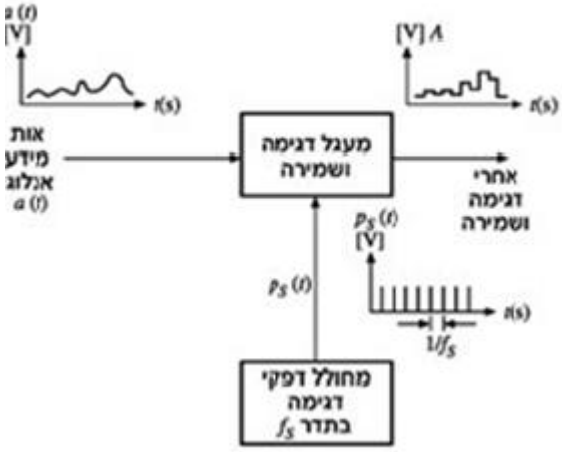
דגימה הינה פעולה בתהליך הפיכת אות אנלוגי לאות בדיד.
דגימת אותות תיעשה באופן מחזורי כך שנקבל אות בדיד שנדגם במרווחי זמן שווים.

<u>תכונות</u>

תדר דגימה: פרמטר הקובע את קצב הדגימה של אות אנאלוגי. פרמטר זה חייב להיות גדול פי 2 לפחות מהתדר המקסימלי שזה רכיב התדר הגבוה ביותר בספקטרום של האות.
מרווח דגימה: הזמן המוגדר בין הדגימות.
קוונטיזציית ערכי האות האנלוגי: ייצוג אמפליטודה במישור הבדיד. ככל שיש יותר ביטים לייצוג כל דגימה, כך האיכות שרואים את האות הבדיד גבוהה יותר (יותר קרוב לתצוגת האות האנלוגי).
אינטרפולציית האות הבדיד לאות רציף: שחזור לאות המקורי.

דוגמא למכשיר המבצע דגימות של אותות חשמליים הוא אוסילוסקופ, בעזרתו ניתן לנתח אותות חשמליים כפונקציה של הזמן. המכשיר מקבל אות אנלוגי, דוגם אותו, מחלץ את כל הרכיבים שמרכיבים את האות ואז משחזר בחזרה את האות האנלוגי שנקלט בכניסה, כאשר הצג של המכשיר מקרין את האות המשוחזר.

אופן ביצוע הדגימה והשפעתה



איור 9 - מבנה למערכת המבצעת דגימה

אות כניסה אנלוגי נכנס למעגל דגימה.

שעון פולסים שנקבע מראש על ערך מרווח הדגימה (חלק חשוב מההמרה לאות ספרתי) קובע את התדר במעגל הדגימה. מעגל דגימה הוא רכיב חומרתי לביצוע דגימה בפועל, שם האות האנלוגי מוכפל בפולסים (מסרק הלמים אינסופי באופן אידאלי) בתהליך, האות עובר ממימד זמן רציף למימד זמן בדיד כאשר הערכים דטרמיניסטיים.

הצורה המתמטית לתאר דגימה היא על ידי הנוסחה הבאה:

$$f(t_0) = \int_{t_0-}^{t_0+} f(t) \cdot \delta(t - t_0)dt$$

האינטגרל בנוסחה מתאר את פעולת הקונבולוציה של האות שרוצים לדגום עם אות הלם הממוקם בנקודת זמן ספציפית (t_0) בה רוצים לבצע את הדגימה (הזמן t_0 יחושב על ידי הנתון מרווח הדגימה).

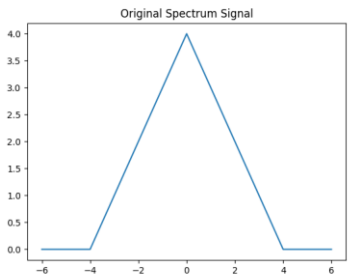
פעולה מתמטית זו יוצרת שכפול של הספקטרום של האות *f(t)* במישור התדר סביב נקודת הדגימה.

לדוגמא:

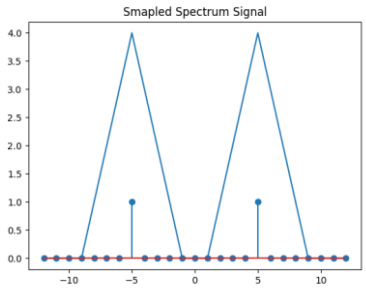
נניח שרוצים לדגום את האות בעל הספקטרום הבא:

ניתן לראות שספקטרום האות בתחום [-4,4].

אם נגדיר שאנו דוגמים את האות במרווח דגימה שערכו 10 נקבל שספקטרום האות שנדגם ישוכפל בספקטרום כל 10 יחידות בתדר.



איור 11 –דוגמא לספקטרום של אות



איור 10 - דוגמא לשכפול הספקטרום כתוצאה מדגימה

הערות לתכנון מערכת לדגימת אותות אנאלוגיים :

- על מנת לבצע שחזור של האות האנאלוגי שנקלט על ידי מערכת ספרתית אנו נדרשים לבצע דגימות. פעולה מתמטית זו גורמת לשינוי בספקטרום של האות האנאלוגי ולכן לאחר הדגימה נדרש לבצע פעולות על מנת לשחזר את האות המקורי.

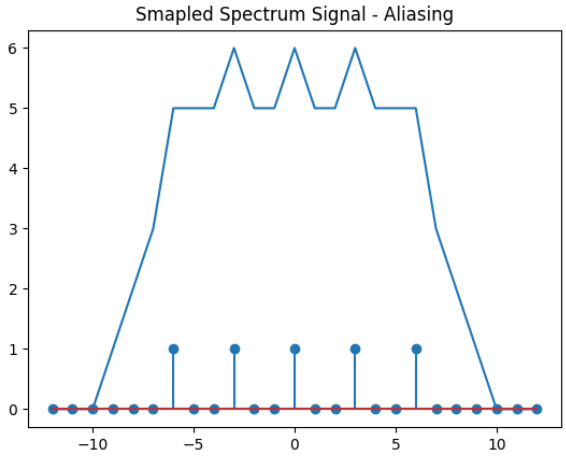
קצב הדגימה הינו פרמטר חשוב, ואם הוא לא מחושב כראוי אנו עלולים להיתקל בשגיאות חמורות בדגימת האות אשר לא ניתן לפתור אותם.

בעיית ה-Aliasing :

נניח שרוצים לדגום את אותו האות עם מרווח דגימה קצר יותר מ-10 שערכו 3.

אנו נקבל שהשכפול בין הגרפים עולה אחד על השני וכתוצאה מכך נוצר עיוותים בשחזור האות.

הצגה גרפית :



איור 12 - דוגמא לספקטרום של אות שנדגם מתחת לתדר Nyquist

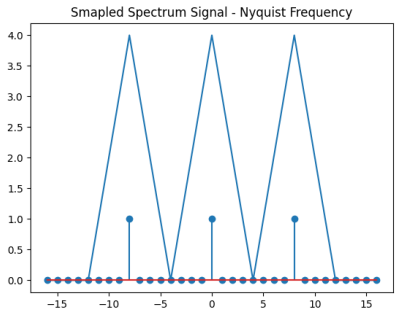
ניתן לראות שכתוצאה מדגימות במרווחי זמן ארוכים יותר התדר קטן ולכן המרווח בין ההלמים שמייצגים את נקודות הדגימה קטן, וכתוצאה מכך נוצר סופרפוזיציה בין הספקטרום של האות המקורי לספקטרום של האות הדגום מה שיוצר ספקטרום של אות חדש עם עיוותים.

- לצורך פתרון של בעיית ה-Aliasing יש לדגום לפי תנאי Nyquist, שאומר שתדר הדגימה צריך להיות לפחות פי 2 יותר גדול מההרמוניה בעלת התדר הגבוהה ביותר בספקטרום.

ניסוח מתמטי לתנאי Nyquist :

$$f_{sample} \geq f_{Ny} = 2 \cdot f_{max} \qquad f_{Ny} = 2 \cdot f_{max}$$

הצגה גרפית של האות מהדוגמה אשר נדגם בדיוק בתדר Nyquist :



איור 13 - דוגמה לספקטרום של אות שנדגם בדיוק בתדר Nyquist

• ממירים DAC/ADC

היחידות שבעזרתם נוכל לדגום\לשחזר אותות אנאלוגיים הם הממירים DAC,ADC.

- **ממירי DAC:**

ממירי DAC הם רכיבים אלקטרוניים שממירים רמות מתח בדידות לרמת מתח אנאלוגי.

לרכיב יש n כניסות דיגיטליות כאשר כל צירוף בינארי שניתן ליצור עם הדקי הכניסה מייצגים רמת מתח אנאלוגי במוצא הרכיב.

לרכיב יש כניסת מתח ייחוס שבעזרתה ניתן לקבוע את רמת המתח המקסימלית שנקראת מהרכיב.

בעזרת שני הפרמטרים (כמות הכניסות ומתח הייחוס) מוגדר פרמטר נוסף של הרכיב הנקרא רזולוציה.

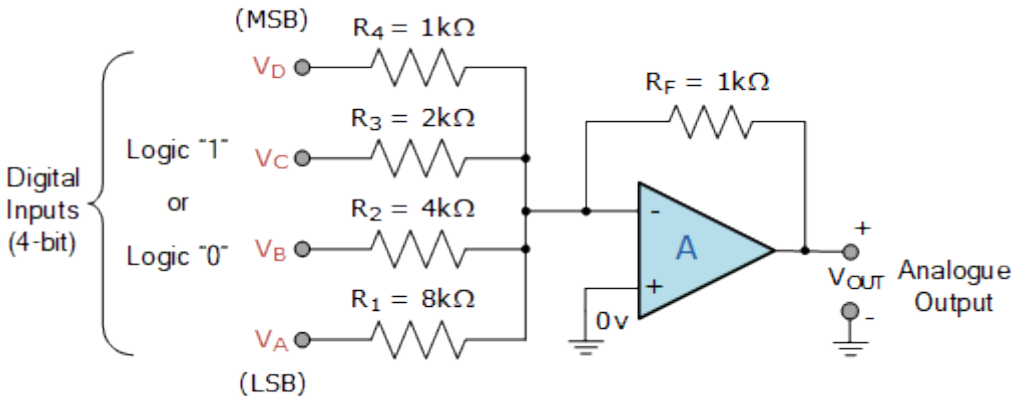
הרזולוציה מגדירה מה השינוי הכי קטן במתח המוצא שהרכיב מסוגל לבצע, פרמטר זה מוגדר לפי הנוסחה הבאה:

$$Resolution = \frac{V_{Ref}}{2^n}$$

עבור כל ערך בינארי שנכניס לרכיב DAC נוכל לחשב את המתח במוצא הרכיב לפי הנוסחה:

$$V_{out} = D[BIN] \cdot Resolution$$

דוגמא למימוש רכיב DAC בתצורת נגדים משוקללים בינארית:



איור 14 - מעגל חשמלי הממש ממיר DAC מסוג נגדים משוקללים בינארית

המגבר בתמונה מסוג מגבר סוכם.

פיתוח הנוסחה:

$$I_f = I_1 + I_2 + I_3 + I_4$$

$$I_x = \frac{V_x}{R_x}$$

$$-\frac{V_{out}}{1[K]} = \frac{V_A}{8[K]} + \frac{V_B}{4[K]} + \frac{V_C}{2[K]} + \frac{V_D}{1[K]}$$

$$V_{out} = -\left(\frac{V_A}{8} + \frac{V_B}{4} + \frac{V_C}{2} + \frac{V_D}{1}\right)$$

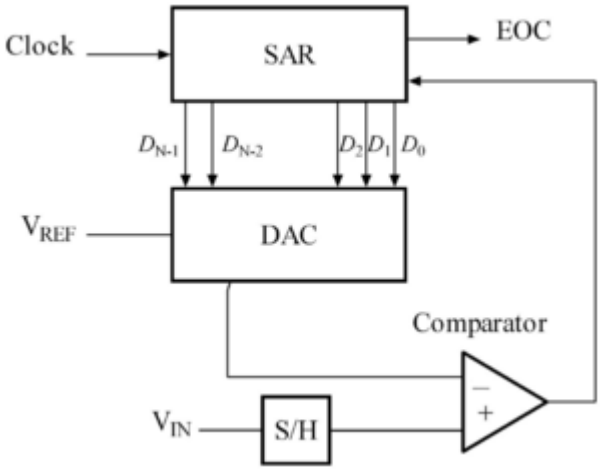
במקרה הנ"ל ניתן לראות שעל ידי הנגדים ניתן להציב ערכים בינאריים בהתאם למתח ייחוס שמוגדר לממיר ולקבל מתח מוצא אנאלוגי הופך מופע.

בנוסף ערכו של הנגד R_f מאפשר הגברת מתח המוצא בהתאם לשימושי הממיר.

- **ממיר ADC:**

ממיר ADC הוא רכיב אלקטרוני שבאמצעותו ניתן לדגום אותות אנלוגיים ולהמיר אותם לאותות ספרתיים.

בפרק זה נסביר על ממיר ADC מסוג SAR.



איור 15 - סכמת בלוקים ש מערכת הממשת ממיר ADC מסוג Successive Approximation

המתח Vin נכנס לרכיב דגימה ושמירה שמעביר את המתח להדק החיובי במשווה, ולהדק השלילי נכנס המתח שיוצא מממיר ה-DAC. כאשר המתח ברגל החיובית גדול יותר מוצא המשווה שווה ל-'1' וכאשר ההדק השלילי גדול יותר מוצא המשווה שווה ל-'0'.

המערכת SAR הינה מערכת שתפקידה לקבוע את ערכם של הסיביות בכניסה של רכיב ה-DAC שהם גם מוצא של רכיב ה-ADC. ה-SAR היא מכונת מצבים שסורקת את הסיביות בכניסה של ה-DAC לפי השלבים הבאים :

- ה-SAR מוציא ערך '1' לוגי בסיבית הגבוהה ביותר כך שהמתח במוצא ה-DAC עולה ב-0.5.
- ה-SAR בודק אם המתח במוצא ה-DAC גדול יותר ממתח הכניסה של הממיר (Vin).
  אם כן, סיבית זו יורדת בחזרה לערך '0' לוגי.
  אם לא, סיבית זו נשארת בערך '1' לוגי והסריקה ממשיכה לסיבית הבאה אחריה.
- שני השלבים מתבצעים עבור כל סיבית בכניסה של ממיר ה-DAC.
- לאחר סיום ביצוע התהליך עבור כל הסיביות בממיר המערכת מוציאה ערך '1' לוגי בהדק המוצא EOC.
  תפקיד הדק זה לעדכן את החומרה החיצונית שמתממשקת לממיר ה-ADC שהממיר סיים לבצע את פעולת ההמרה ואפשר לקרוא את הנתונים.

לאחר סיום פעולת הממיר ניתן לקרוא את הערך הבינארי שנכנס לממיר DAC המייצג את המתח בכניסת הרכיב ADC שנדגם בתחילת תהליך ההמרה.

מסננים

סינון בא במטרה להוציא תכונות או רכיבים לא רצויים של אות (כמו רעש, שכפולים של מידע) וגם לשם מניפולציה על רכיבי התדר/הזמן של האות.

בעיבוד אות דיגיטלי, מדובר במסננים דיגיטליים:

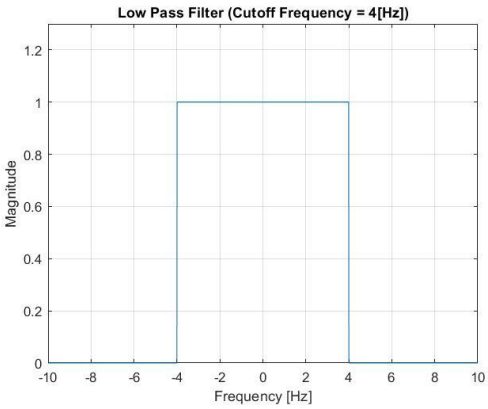
מסננים דיגיטליים מורכבים ממספר קטבים ואפסים והם מכתיבים את אופי הסינון (סינון תדרים גבוהים, נמוכים, אמצעיים)

- **מסננים אנאלוגיים (LPF,HPF)**

<u>מסנן מעביר נמוכים (Low Pass Filter)</u>

מסנן המעביר תדרים נמוכים מתדר הקטעון (ω_c) ומאפס את כל התדרים שמעליו.

ניתן לייצג גרפית את הספקטרום של פונקציית התמסורת של המסנן כפי שמתואר לדוגמה באיור הבא:



איור 16 - דוגמא לתגובת תדר של מסנן LPF

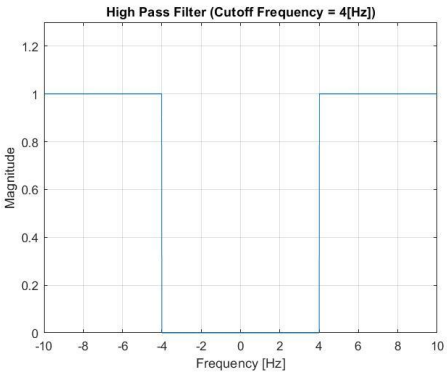
ניתן לראות באופן אינטואיטיבי שכאשר נכפיל את הספקטרום של המסנן עם ספקטרום של האות בכניסה של המסנן (הכפלה בתדר מייצג קונבולוציה בזמן), נקבל במוצא המסנן את האות שהכנסנו רק שנקבל אותו ללא כל התדרים מעל 4[Hz].

<u>מסנן מעביר גבוהים (High Pass Filter)</u>

באופן דומה למסנן מעביר הנמוכים מסנן מעביר גבוהים מעביר רק תדרים גבוהים מתדר הקטעון (ω_c).

ניתן לתאר את הספקטרום של מסנן זה באופן הבא:

עבור מסנן זה ניתן לראות שהתדרים שעוברים הם התדרים הגבוהים מתדר 4[Hz].

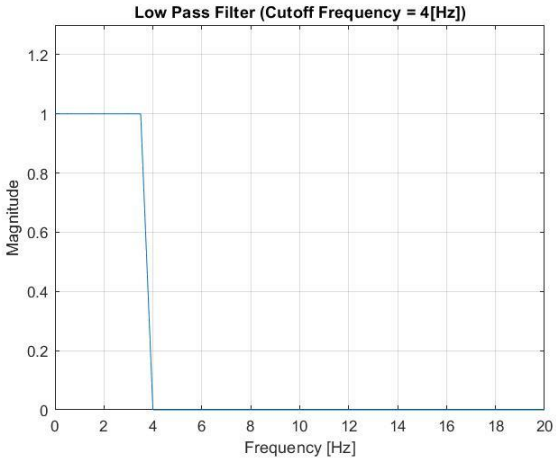


איור 17 - דוגמא לתגובת תדר של מסנן HPF

<u>דוגמה לניתוח במישור התדר של אות שעובר דרך מסנן LPF :</u>

בדוגמה זו נסתכל על מסנן LPF אשר קוטם את כל התדרים מעל ל-4[Hz].

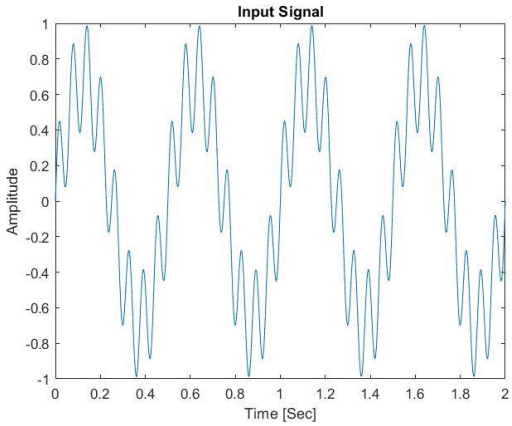
ניתן לראות באיור את התגובת תדר של המסנן :



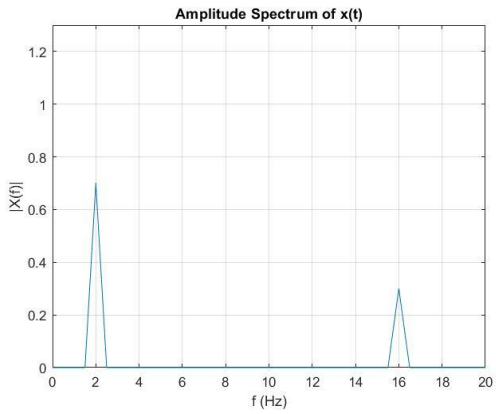
איור 18 - תגובת תדר של מסנן LPF בתדר קטעון 4[Hz]

נכניס בכניסה של המסנן אות בעל שני הרמוניות בתדרים : 2[Hz],16[Hz].

ניתן לראות באיורים הבאים את האותות כפונקציה של הזמן והייצוג הספקטרלי שלהם בתדר :

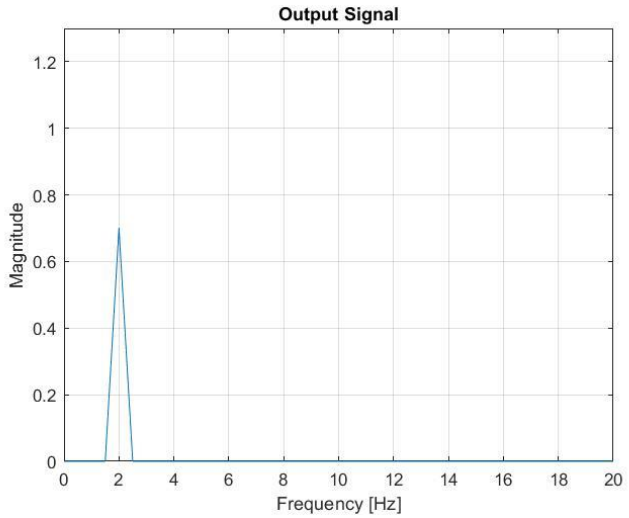


איור 20 - אות כניסה למסנן המכיל 2 הרמוניות



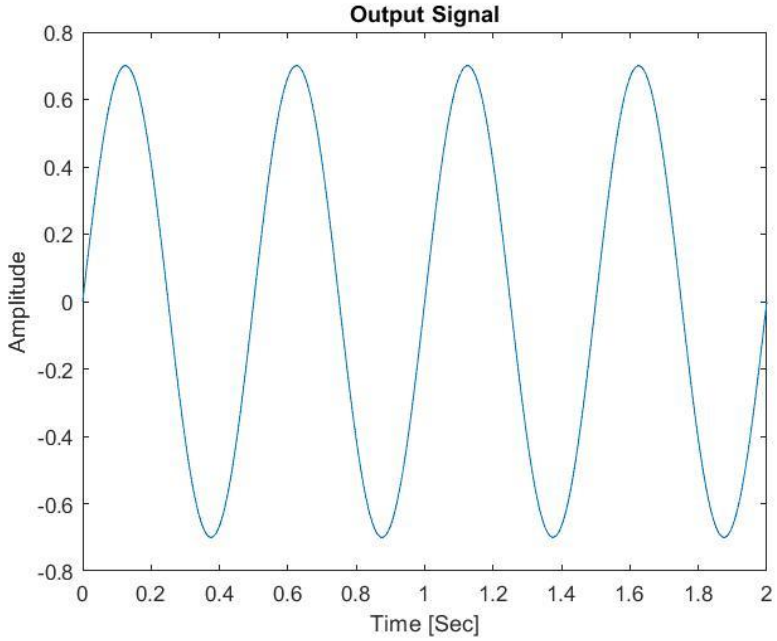
איור 19 - אות הכניסה למסנן המכיל שני הרמוניות במישור התדר

כאשר נעביר את האות דרך המסנן יתבצע קונבולוציה בין האות בכניסה לאות במוצא במישור הזמן, פעולת הקונבולוציה במישור התדר היא הכפלה של שני האותות (לפי התכונות של התמרת פורייה) ולכן, נקבל במוצא המסנן רק את התדרים הנמוכים מ-4[Hz] ולכן האות במוצא מיוצג במישור התדר באופן הבא :



איור 21 - ספקטרום של אות המוצא מהמסנן

אות המוצא שקיבלנו מכיל רק את ההרמוניה בעלת התדר הנמוך ([2[Hz) ולכן נראה במישור הזמן אות פחות רועש במוצא:

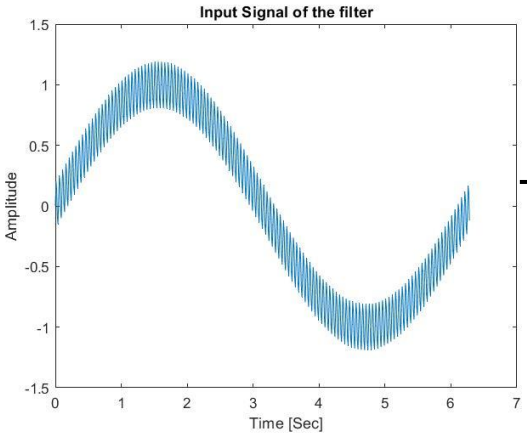


איור 22 - אות המוצא כפונקציה של הזמן

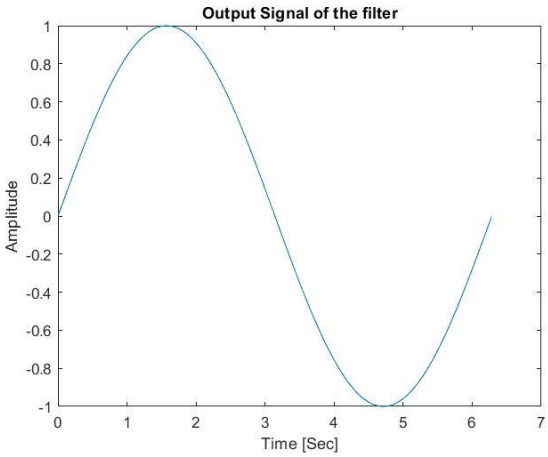
- **מסננים דיגיטליים (FIR)**

מסננים דיגיטליים הוא נושא נרחב בעיבוד אותות ספרתי.
שימוש של מסננים דיגיטליים הוא כאשר רוצים להפריד בין הרמוניות שונות הנמצאות על אות אחד.

בעיקרון דומה למסננים האנאלוגיים המסנן הספרתי מקבל בכניסתו אות עם כמה הרמוניות ובמוצאו מתקבל רק חלקם לדוגמא:



איור 24 - דוגמה לאות הכניסה של ה-FIR



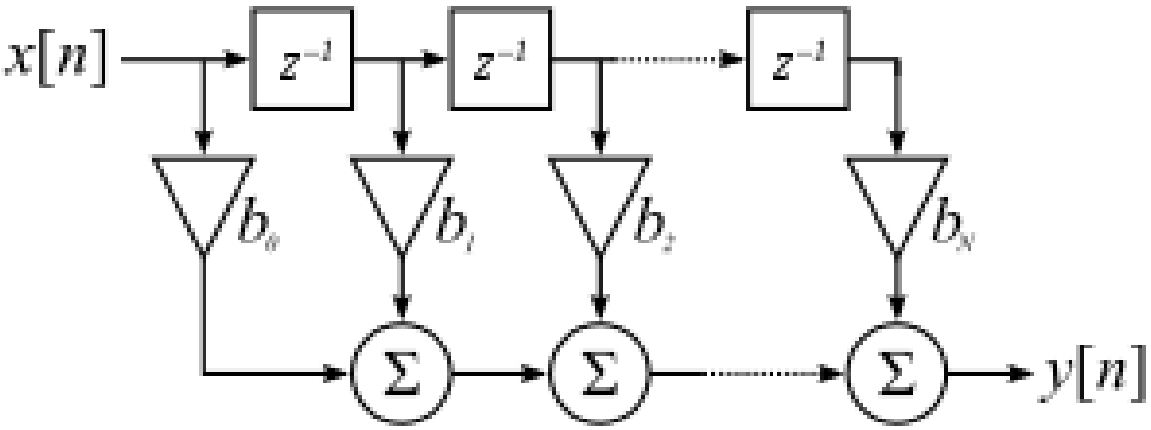
איור 23 - דוגמה לאות מוצא מה-FIR

כאשר מדובר במסנן ספרתי/דיגיטלי האותות בכניסה ובמוצא המסנן הם אותות בדידים, כלומר האותות מיוצגים כסדרת מספרים סופית כאשר כל איבר בסדרה מייצג דגימה של האות האנאלוגי/הרציף.
מרווח הדגימה הינו פרמטר חשוב, הוא קבוע בין כל הדגימות ועלינו להגדיר אותו מראש.
הפרמטר שמאפיין את המסנן נקרא תגובת תדר.
תגובת התדר של המסנן נותנת לנו תמונה כוללת המציגה את היחס בין עצמת האות במוצא לעוצמת האות בכניסה עבור כל התדרים.
<u>(אפשר להוסיף גרף המתאר את העוצמה כפונקציה של התדר)</u>
מכיוון שאנו מממשים מסננים בתחום הספרתי לא נוכל להתייחס לתגובת התדר לאורך כל הספקטרום אלא רק בתחום התדרים שמקיים את תנאי Nyquist ($0 < f[Hz] < \frac{f_s}{2}$).

קיימים שני סוגים של מסננים ספרתיים הניתנים למימוש במערכות דיגיטליות.

1. מסנן FIR – מסנן בעל תגובה להלם סופית.

התכונה העיקרית של המסנן FIR היא שהתגובה להלם של המסנן סופית בזמן, ולכן מערכת זו נחשבת למערכת יציבה לפי הגדרה, היא לא יכולה לצאת מיציבות.



איור 25 - סכמה המתארת את המבנה של מסנן FIR

אם נחלץ את הנוסחה עבור המוצא של המסנן נקבל:

$$y[n] = \sum_{j=0}^{N} h[j] \cdot x[n-j] = x[n] * h[n]$$

לאחר חילוץ הנוסחה ניתן להבחין שהתגובה להלם מוגדרת על ידי הסדרת מספרים שמכניסים למכפלים וכאשר נבצע קונבולוציה בזמן של התגובה להלם עם הכניסה נקבל את המוצא בכל רגע נתון.

הגדרה של תגובה להלם כללית למסנן FIR:

$$h[n] = \{b_0, b_1, b_2, \dots, b_N\}$$

2. מסנן IIR – מסנן בעל תגובה להלם אין סופית.

מסנן מסוג זה כולל חומרה מורכבת יותר הכוללת משוב, פעולת המשוב במסנן עלולה לגרום לאי-יציבות ולכן החלטנו שלא לנסות לממש מסנן מסוג זה.

מימוש מסנן FIR

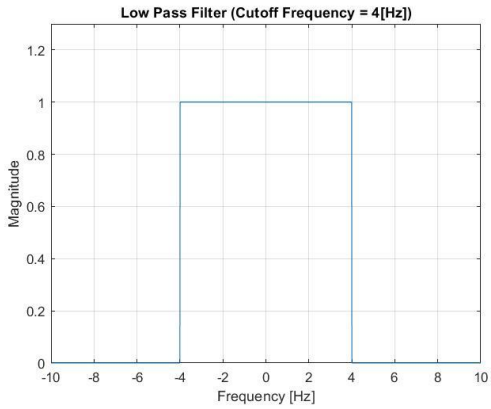
כאשר רוצים לממש מסנן תחילה יש להגדיר את הספקטרום של תגובת התדר שלו, כלומר, עלינו להגדיר איזה תדרים נרצה שהמסנן יעביר ואיזה תדרים נרצה שהמסנן ינחית.

לאחר הגדרת תגובת התדר ניתן להשתמש בנוסחה של ההתמרת פורייה ההפוכה כדי להעביר את תגובת התדר של המסנן לתגובה להלם שלו במישור הזמן.

פעולות אלו נעשות גם בתכנון של מסנן אנאלוגי ההבדל ביניהם הוא שתוצאת ההתמרת פורייה ההפוכה נותנת תגובה להלם כפונקציה רציפה, ולעומת זאת התגובה להלם במסנן הדיגיטלי היא סדרת מספרים סופית.

לצורך הבנת השלבים בתכנון מסנן FIR נסתכל על הדוגמא הבאה:

נניח שרוצים לממש מסנן FIR שיעביר את התדרים הנמוכים (LPF) עד לתדר קטעון מסוים (f_c)



איור 26 - מסנן LPF עם תדר קטעון 4[Hz]

תחילה יש לבצע התמרה הפוכה על תגובת התדר של המסנן כדי לחשב את התגובה להלם הזמנית בתחום הרציף:

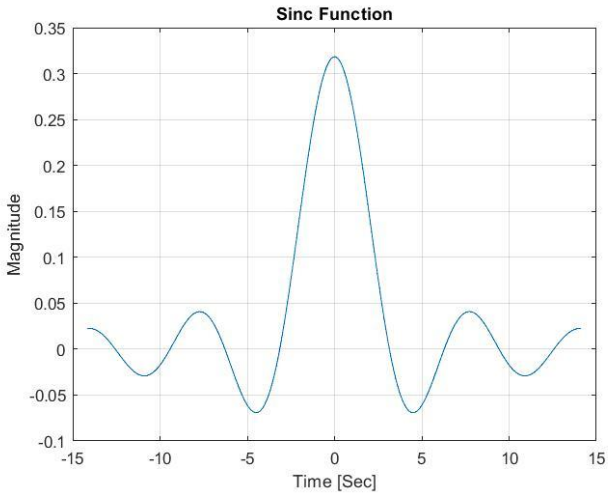
$$h(t) = \frac{1}{2\pi} \int_{\omega=-\infty}^{+\infty} H(\omega) \cdot e^{j\omega t} d\omega = \int_{f=-\infty}^{+\infty} H(f) \cdot e^{j2\pi f t} \; df$$

גבולות האינטגרל מצטמצמים לתחום $[-f_c, +f_c]$ ובתחום זה מקבלים: $H(f) = 1$ ולכן:

$$h(t) = \int_{f=-f_c}^{+f_c} e^{j2\pi f t} \; df$$

לאחר פתרון האינטגרל נקבל פונקציית sinc():

$$h(t) = \frac{\sin(2\pi f_c t)}{\pi \cdot t} \;\rightarrow\; sinc(\cdot)$$

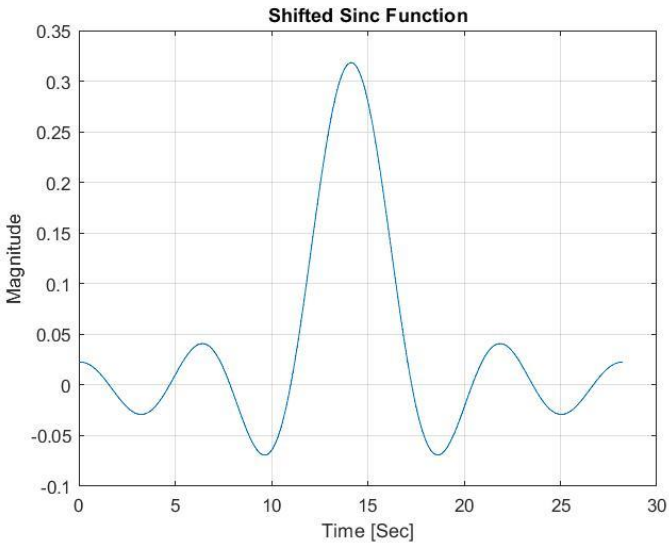


איור 27 - פונקציית Sinc

ניתן לראות שהפונקציה קיימת בכל התחום $(-\infty, +\infty)$ מכיוון שאנו צריכים להכניס לתגובה להלם סדרת מספרים סופית אנו נמצאים בבעיה.

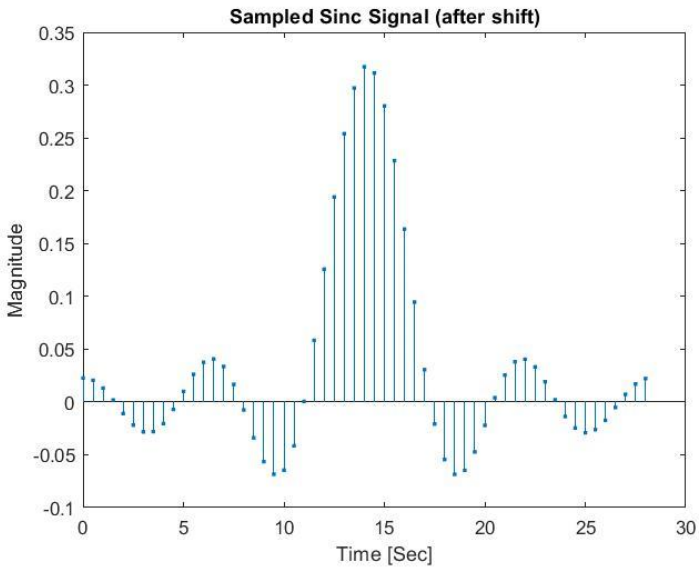
ולכן בשיטה תכנון זו נדרש לעבוד לפי השלבים הבאים:

1. להגדיר תחום סופי בתדר בו המסנן יעבוד (מחוץ לתחום שאנו מגדירים התגובה להלם תהיה שווה לאפס).
2. להזיז את התגובת תדר שהתקבלה מהשלב הקודם לתחום החיובי של ציר התדר.



איור 28 - פונקציית Sinc לאחר ביצוע הזזה

3. לדגום מספר סופי של ערכים מהתגובת תדר:



איור 29 - פונקציית Sinc לאחר דגימות

הערה: כאשר מגדירים תחום סופי לתגובת התדר של המסנן אנו עלולים לגרום לתדרים גבוהים לעבור דרך מסנן שאמור לסנן תדרים אלה.
לצורך התגברות על בעיה זו אנו מבצעים הכפלה של התגובה להלם בפונקציית חלון.

<u>מסקנה</u>: בתכנון מסנן FIR עלינו לקבוע את הפרמטרים הבאים:

1. תדר הדגימה – נדאג שיהיה גבוהה ככל האפשר כדי לא להיות מוגבלים על ידי תנאי Nyquist.
2. אורך התגובה להלם של המסנן – ככל שפרמטר זה יהיה יותר גדול נקבל רזולוציה טובה יותר באות המוצא.
3. בחירת פונקציית החלון – קיימים סוגים שונים של פונקציית חלון שניתן להכפיל בהם את התגובה להלם הסופית לכל אחת יש תכונות אחרות ועלינו לבחור את הפונקציה המתאימה ביותר למה שנרצה לממש.

תיאור הכלים והעזרים בסביבת הפיתוח

רכיב FPGA

רכיב FPGA הוא רכיב אשר ניתן לתאר את המבנה החומרתי שלו על ידי שפות תיאור חומרה (HDL).

לצורך תכנון חומרה על רכיב FPGA נדרש ידע והבנה במעגלים ספרתיים ובשפות תיאור חומרה, ברמה שבה המתכנן יודע לקנפג את החומרה בצורה יעילה ובאמצעות כתיבת קוד על סמך מה שהוא נדרש לבצע.

ניתן לפתח חומרה ספרתית (דיגיטלית) על גבי הרכיב FPGA באמצעות כתיבת קוד.

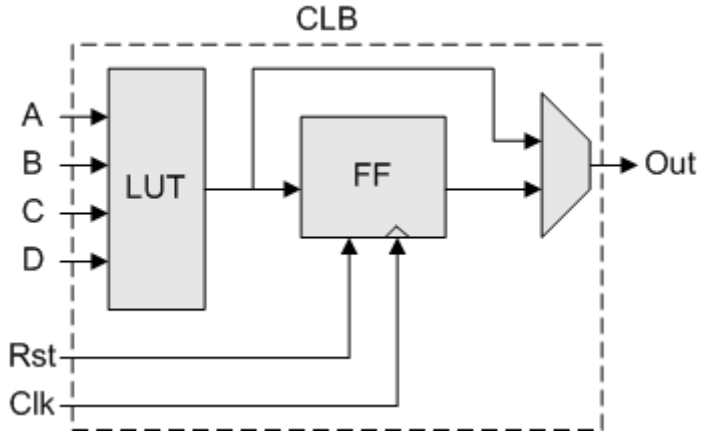
קיים סוג נוסף של רכיב מתוכנת הנקרא CPLD, רכיב זה בעל משאבים איטיים יותר והוא מוגבל יותר מבחינת היכולות שלו לעומת רכיב ה - FPGA.

בפרויקט שלנו החלטנו לממש את החומרה על רכיב FPGA.

- **מבנה רכיב ה - FPGA**

רכיב FPGA מורכב מיחידות המכונות אלמנטים לוגיים (LE – Logic Elements) נקרא גם CLB.

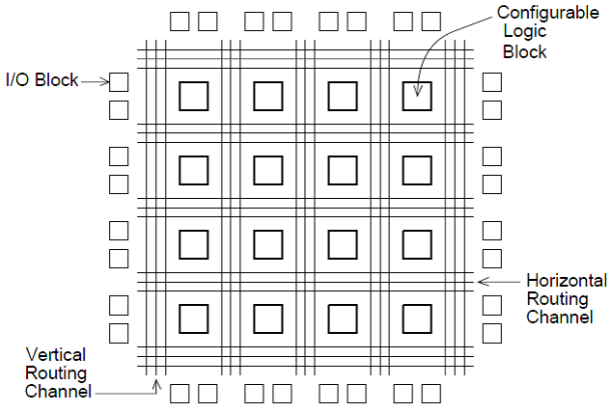
כל יחידה לוגית כזו מורכבת ממעגל צירופי הנקרא LUT שיכול להיות ממומש כשער לוגי, דלגלג (Flip Flop), ומרבב במוצא הרכיב.



איור 30 - המבנה הלוגי של CLB

כל אלמנט לוגי כזה מהווה אבן בניין למעגל אלקטרוני דיגיטלי שניתן באמצעותו לתכנן חומרה עבור מערכות מורכבות.

ברכיב FPGA יש כמות גדולה של אלמנטים לוגיים שמחווטים ביניהם כך שניתנת היכולת למפתח ליצור מעגל ספרתי מורכב.



איור 31 - מבנה כללי של רכיב ה-FPGA

בנוסף לאלמנטים שמממשים את היחידות הלוגיות השונות, רכיב FPGA בדרך כלל מכיל גם חומרה נוספת אשר עוזרת לנו בתכנון, ומגדילה את האפשרויות השונות המאפשרות למפתח להיעזר בהן לצורך מימוש חומרה פשוט ויעיל יותר כמו מערכות PLL, מודולי זיכרונות מובנים ומערכות DSP.

- **שלבי פיתוח**

פיתוח חומרה על גבי רכיב FPGA דורש להעביר את החומרה 3 שלבים עד לקבלת רכיב שעובד כראוי.

א. שלב ראשון – סינתזה

סינתזה זה התהליך הראשוני שבו מנסים להוריד את הרעיון של הרכיב שרוצים לתכנן לקוד שמתאר את החומרה.

לעיתים, הפעולה שנרצה שהחומרה תבצע תהיה פעולה פשוטה ואז נוכל לבטא את החומרה באמצעות קוד בשיטת Dataflow Modeling, ולעיתים קרובות יותר כאשר נרצה לתאר חומרה מורכבת יותר נרצה לתאר אותה בשיטת Behavioral Modeling.

בסוף התהליך של תיאור החומרה באמצעות HDL מתקבל שרטוט סכמתי שמתאר את החומרה (הסכימה מכונה RTL).

בדרך כלל נעבור על המסלול נתונים של הסכימה הלוגית ונוודא שהמערכת שתכננו עומדת בדרישות שלנו.

בנוסף, ננסה להימנע מנועלים (Latch) הממומש במערכות שאנו מתכננים, מכיוון שהם עלולים לגרום לאותות במערכת לצאת מסנכרון של השעון הכללי שמסופק למערכת.

בסוף שלב הסינתזה נוצר דו״ח שנותן לנו נתונים על המשאבים שהחומרה שכתבנו צורכת מרכיב FPGA אתו אנו עובדים.

ב. שלב שני – סימולציה

בשלב זה אנו נדרשים לכתוב Script (הנקרא Test Bench) שמריץ מספר מסוים של אפשרויות בכניסה של המערכת החומרתית, ומתקבל תרשים זמנים (Waveform) שבו ניתן לוודא אם הרכיב

מבצע את הפעולה הדרושה. הסימולציה רצה על תוכנה לפני צריבת החומרה על רכיב ה- FPGA.
מטרת הסימולציה לגלות בעיות בתכנון הרכיב לפני הצריבה.

ג. שלב שלישי – הרצה ודיבוג תקלות
לאחר הרצה רצוי לבדוק את האותות במערכת לצורך זיהוי תקלות (בדרך כלל ניתן לבדוק את האותות במערכת על ידי Logic Analyzer אנו נשתמש בכלי SignalTap של Intel).

שפות תיאור חומרה HDL

יש 3 שפות HDL נפוצות בתעשייה לכתיבת חומרה ברכיבי FPGA: VHDL, Verilog, SystemVerilog.
השפה הנכונה לכתוב בה את הסינתזה של המודול היא VHDL, לכן נתכנן באמצעותה את המודולים בפרויקט למרות שהיא יותר קשוחה משאר שפות ה- HDL.
שפת ה-HDL שבאמצעותה נבצע את הסימולציות תהיה Verilog מכיוון שהיא כוללת כל מיני תוספות שיכולות להקל על כתיבת הסימולציה.

תוכנות Quartus ומodelsim

סביבת הפיתוח שלנו בפרויקט תכלול את התוכנות הבאות:

1. Quartus – תוכנה שפותחה על ידי Intel למימוש חומרה באמצעות שפות HDL על רכיבי FPGA של חברת Altera.

באמצעות תוכנה זו אנו נוכל לבצע סינתזה לרכיבים שאנו כותבים, נוכל לקבל שרטוטי RTL ודיאגרמה של מכונות מצבים במידת הצורך, כמו כן, נוכל לצרוב באמצעות תוכנה זו את החומרה על הרכיב FPGA שאנו עובדים אתו.

2. ModelSim – תוכנה זו נועדה לביצוע סימולציות לחומרה שנכתבה באחת משפות ה HDL.

התוכנה ModelSim מאפשרת לנו לבצע סימולציות בצורה מקצועית הכוללת הרצת סימולציות ארוכות, מדידת זמנים של אותות במערכת, הצגת אותות דיגיטליים ואנאלוגיים והצגת אותות בשיטות קידוד שונות.

הכרטיסים שבחרנו לעבוד איתם

לצורך מימוש הפרויקט בחרנו לעבוד עם הרכיב FPGA, EP4CE115F29C7N של חברת Altera מסדרת הרכיבים Cyclone IV E.

הרכיב FPGA מגיע בכרטיס פיתוח DE2-115 של חברת Terasic.

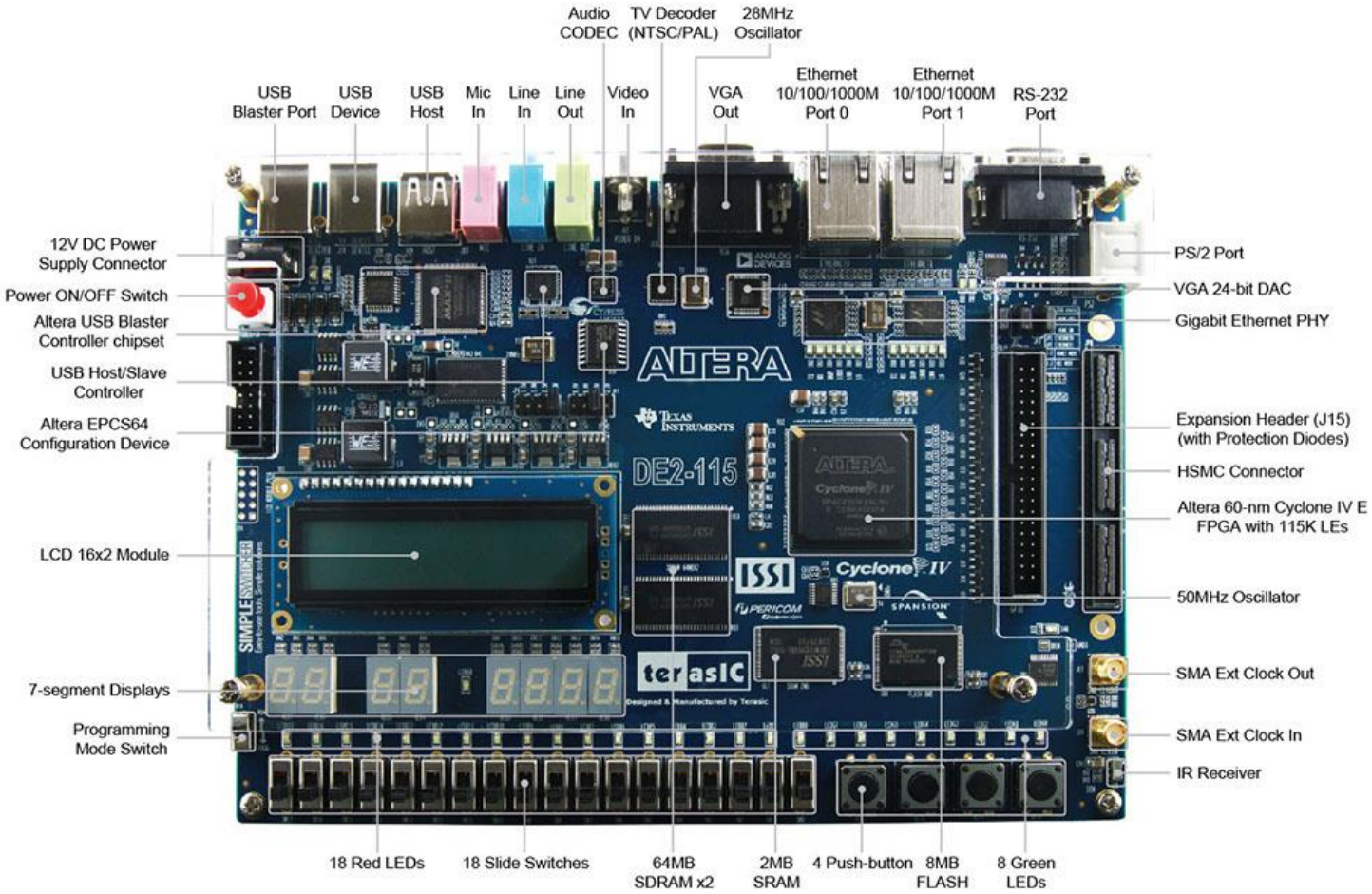
נתוני הרכיב FPGA:

- מכיל 114,480 אלמנטים לוגיים.
- מכיל זיכרון בגודל: 3.888[MBit]
- מכיל 4 רכיבי PLL המשמשים להכפלת תדר השעון.
- 528 פורטים כניסה ויציאה המשמשים לממשק משתמש.

חוץ מהרכיב FPGA הכרטיס מכיל:

- צורב לרכיב FPGA.
- זיכרונות חיצוניים מסוגים שונים (SRAM, Flash, EEPROM) והתממשקות לכרטיס SD חיצוני.
- 4 לחצנים, 18 מפסקים.
- 9 נורות ירוקות, 18 נורות אדומות ו-6 תצוגות מסוג שבעת המקטעים (7 Segment).
- רכיב מקודד\מפענח אותות שמע עם יציאות שמע.
- תצוגת LCD בגודל 16x2.
- יציאת HSMC לצורך חיבור כרטיסי הרחבה.
- יציאת וידאו VGA הכוללת ממיר DAC 4 סיביות לכל צבע (RGB).
- יציאת RS232 לתקשורת UART עם הרכיב FPGA.
- חיבור PS/2 למקלדת\עכבר העובדים בתקנים הישנים.
- יציאות USB היכולות לשמש כHost או כDevice המחוברות לרכיב FPGA.

בנוסף, השתמשנו בכרטיס הרחבה המתחבר לכרטיס DE2-115 בחיבור HSMC המאפשר קריאה של אותות אנאלוגיים בקצבים גבוהים והוצאת אותות אנאלוגיים על ידי שימוש בממירים ADC/DAC.



איור 32 - הכרטיס DE2-115

תוכנית עבודה להמשך

פעילות	חודש 3	חודש 4	חודש 5
מימוש מודולים של המיקרופרוסור + אינטגרציה ובדיקות	V		
תכנון צורב למיקרופרוסור		V	V
מימוש פריפריות חומרה לעיבוד אות ושילובם במיקרופרוסור		V	V
כתיבת הספר פרויקט		V	V

סיכונים להמשך הפרויקט

1. חוסר עמידה בזמנים.

ביבליוגרפיה

1. https://en.wikipedia.org/wiki/Digital_signal_processor
2. https://he.wikipedia.org/wiki/ארכיטקטורת_MIPS
3. https://en.wikibooks.org/wiki/MIPS_Assembly/Instruction_Formats
4. https://en.wikipedia.org/wiki/MIPS_architecture