

מערכת לכיווץ הרמוני של אותות דיבור

בעזרת מיקרו מחשבים

חיבור על פרוייקט

לשם מילוי חלקי של הדרישות לקבלת התואר

מגיסטר למדעים

בהנדסת חשמל

מאת

הלרשטיין רן

הוגש לסנט הטכניון - מכון טכנולוגי לישראל

תמוז תשלי"ט ח י פ ה יולי 1979

הפּרוֹיֵיקט נעשׂה בּהנחײת דר' דוד מלאך
במעבדה לעיבוד אותות בפקולטה להנדסת
חשמל.

תודתי בתונה לדר' דוד מלאך על הנחייתו
המועילה ועזרתו המסורה לכל אורך הדרך
וכן על יחסו האוהד והחם.

כמו כן אני מודה למהנדס המעבדה מר יורם אור חן
על עזרתו ותמיכתו.

תוכן העניינים

עמוד מסי

1	תקציר
3	רשימת סמלים וסימנים
6	פרק 1 מבוא
6	1.1 הקדמה
7	1.2 שלבי הפרוייקט
8	1.3 מבנה העבודה
10	פרק 2 אלגוריתם לכיווץ הרמוני של אותות דיבור
10	2.1 תאור האלגוריתם
18	2.2 חלוקת תדר
21	2.3 הכפלת תדר
22	2.4 בחירת פונקצית החלון
27	2.5 האלגוריתם הממוש
39	פרק 3 סימולציות למערכת מיצוי ה- PITCH
39	3.1 הקדמה
41	3.2 אפנון דלתא
43	3.3 CENTER CLIPPING
45	3.4 שידור אינפורמציה ה- PITCH
51	3.5 בדיקת הדרישות מממזה ה-PITCH
53	3.6 אלגוריתם למימוש
57	פרק 4 סימולציות ל- ADAPTIVE DELTA MODULATION
57	4.1 הקדמה
58	4.2 ADM של JAYANT
59	4.3 ADM - HADM היברידי
60	4.4 CONTINUOUSLY VARIABLE SLOPE DM
61	4.5 תוצאות הסימולציות
69	פרק 5 סימולציה למערכת כולה

ת ק צ י ר

מערכות קשר ספרתיות הן כיום אחת הבעיות העומדות במרכזם של מחקרים ופרוייקטים רבים.

קצבי העברת האינפורמציה דרך קווי הטלפון והאלחוט מוגבלים ע"י רוחב סרט של 3.2KHz שנקבע בהתאם לרוחב הסרט של דיבור. במערכות קשר ספרתי מעבירים כיום בעיקר אינפורמציה אלפא נומרית. העברת דיבור בצורת ספרתית בקווים הרגילים בצורה פשוטה, ע"י דגימה ושליחת סביות אינה אפשרית. ברור גם כי הדיבור כולל עודף אינפורמציה כאשר הוא נדגם ב- 7.4KHz $(3.2KHz \times 2)$ ומשודר בערוץ.

מטרת פיתוח המערכת היתה לממש בזמן אמיתי אלגוריתם אשר יאפשר להקטין את כמות האינפורמציה אותה יש לשלוח בערוץ. הקטנת כמות האינפורמציה מתבצעת ע"י כיווץ הרמוני של אות הדיבור ולאחר מכן ביצוע ADM (Adaptive Delta Modulation)

הכיווץ ההרמוני מתבצע ע"י אלגוריתם המבוסס על פיתוח של התמרת פוריה לזמן קצר של אות הדיבור כאשר ידוע התדר הבסיסי של אות הדיבור (PITCH). האלגוריתם מופעל על הדיבור בתחום הזמן ודורש עקיבה אחרי התדר הבסיסי של הדיבור. ה-ADM מופעל על אות הדיבור לאחר הכיווץ ההרמוני.

במסגרת פיתוח המערכת נעשו סימולציות לכל האלגוריתמים. בעזרת הסימולציות נחקרו אלגוריתמים שונים לגילוי PITCH ול-ADM. לאחר שנבחרו האלגוריתמים למימוש נבחנו פישוטים לאלגוריתמים כדי להתאימה למימוש בזמן אמיתי ואם זאת ללא פגיעה ניכרת בטיב הביצועים שלהם. בנוסף נחקרו הפרמטרים של האלגוריתמים ונבחרו במתאימים.

רשימת סמלים

PITCH	-	אחת השיטות לגילוי	AMDF
	-	מתמר אנלוגי לספרתי	A/D
	-	סיבית	Bit
	-	מקדם הכיווץ	C
PITCH	-	שיטה לקידוד הדיבור באוטוקורלטור גלאי ה-	CENTER CLIPPING
	-	שיטה לגילוי PITCH	CEPSTRUM
	-	מתמר ספרתי לאנלוגי	D/A
	-	תדר בסיסי (PITCH) משוערך	F_p
	-	תדר בסיסי (PITCH) מדויק	F_o
	-	תדר הדגימה	f_s
	-	המסנן הבסיסי	$h_o(nT)$
	-	פונקציית החלון בתהליכי הכיווץ והמתיחה	h_N
	-	קלט/פלט. התקני קליטת אינפורמציה ומסירת אינפורמציה של המחשב	I/O
	-	מספר דגמים במחזור	N_p
	-	מספר הצעדים באלגוריתם הכיווץ	N_c
	-	מספר הצעדים באלגוריתם המתיחה	N_s
	-	תדר בסיסי של אות דיבור	PITCH
	-	זכרון קריאה כתיבה	RAM
	-	זכרון קריאה בלבד	ROM
	-	קו בקרה למיקרו מחשב 8085 המאפשר לעצור את ריצתו	READY
	-	מקדם הכיווץ	S
	-	מצב בו אין דיבור בערוץ	SILENCE

- TAPPER חלון הנוצר בצורה מסוימת של אוטוקורלציה
- UNVOICED אות דיבור לא קולי (או לא גרוני)
- VOICED דיבור קולי (גרוני)
- w_k התדר הבסיסי של המסנן ה- k
- $Y^{1/c}$ אות מחולק תדר ב-C
- Y^S אות מוכפל תדר ב-S
- μp מיקרו מחשב
- Δ_o גודל הצעד בתהליכי DM
- 8085 המיקרו מחשב שבעזרתו מומשה המערכת

רשימת קיצורים

Adaptive delta modulation	-	ADM
Continuously variable delta modulation	-	CVSD
Discrete Fourier Transform	-	DFT
Delta modulation	-	DM
Finite impulse response	-	FIR
Kilo bit per second	-	KBPS
Low pass filter	-	LPF
Modulator - De Modulator	-	MODEM
Mili second	-	mSec
CENTER CLIPPING תוכנית למיצוי PITCH בשיטה	-	PITCLP
CENTER CLIPPING תוכנית למיצוי PITCH בשיטה	-	PITCHD
אנליזת פוריה לזמן קצר	-	STFT
Micro Second	-	μ Sec

פרק 1 : מ ב ו א

1.1 ה ק ד מ ה

מטרת העבודה היא פתוח מערכת לכיווץ הרמוני של אות דיבור. על המערכת לאפשר העברת קידוד ספרתי של אותות דיבור במערכות תקשורת חוטיות ואלחוטיות תוך הקטנה של כמות האינפורמציה שיש להעביר דרך קו התקשורת. המערכת שפותחה משתמשת בשני אמצעים להקטנת כמות האינפורמציה המשודרת בקו.

1. כיווץ הרמוני ביחס של 2 ל-1 של אות הדיבור ע"פ האלגוריתם לכיווץ הרמוני [1].

2. שמוש ב- ADM בשיטת ה- CVSD [2] המקטין פי ארבע את כמות האינפורמציה.

האמצעים הללו מופעלים בצורה טורית על הדיבור.

יש לציין כי מבחינת מימוש המערכת, שתי תת המערכות, לביצוע כיווץ הרמוני ולביצוע ADM בלתי תלויות אחת בשניה. המטרה העיקרית של הפרוייקט היתה לפתח ולממש את הכיווץ הרמוני נה- ADM התווסף מכיוון שהוא שיטה נוחה ופשוטה לבצוע קידוד. ניתן להחליף את ה- ADM בשיטות יותר מתוחכמות (A.P.C.) ולהגיע ע"י כך לקצבים יותר נמוכים. המערכת שפותחה יכולה לעבוד במספר מודים:

1. כיווץ דיבור ביחס 1:2 בקצבי דגימה עד 14.4KHz.

2. מתיחת דיבור ביחס 1:2 בקצבי דגימה עד 10KHz.

3. כיווץ דיבור ביחס 1:2 וביצוע ADM על האות המכווץ בקצבי דגימה עד 14.4KHz.

4. D-ADM (התמרת סיביות ה- ADM לדגמים) ולאחר מכן ביצוע מתיחה ביחס 1:2 בקצבים עד 7.2KHz.

המערכת מומשה בעזרת זוג מיקרו מחשבים מסוג INTEL-8085 ואת כל מודי העבודה השונים מייגים ע"י שינוי התוכנה בלבד ללא כל שינוי בחומרה.

קיימות היום מערכות MODEM (להעברת אינפורמציה בערוצי טלפון ואלחוט) הפועלות בקצבים עד 9600 סיביות לשניה ולכן ניתן להשתמש במוד השלישי (משדר) והרביעי (מקלט) כדי להעביר דיבור דרך קווי טלפון וערוצי אלחוט. הקצב המתקבל אז הוא 7200 סיביות לשניה.

פיתוח המערכת התחלק לשני חלקים עיקריים:

1. שלב הסימולציות

בשלב זה נחקרו אלגוריתמים שונים לזיהוי חדר יסודי (PITCH), באותות דיבור ולבצוע ADM ונבדקו ופותחו קירובים לאלגוריתמים הללו אשר איפשרו מימוש האלגוריתם בזמן אמת בצורה פשוטה.

תוכניות הסימולציה נכתבו על מחשב NOVA-2 המצוייד בהתקני קלט/פלט המאפשרים דגימת דיבור בתדר רצוי, עיבודו והשמעתו לאחר העיבוד.

בהמשך לתכנון האלגוריתמים אשר ימומשו נחקרו ונמצאו הפרמטרים לשלושת התהליכים (זיהוי PITCH, כיווץ/מתיחה, ADM) פרמטרים אשר ישמשו אותנו במימוש.

כל התהליכים הנ"ל נעשו על קטעים קצרים של דיבור. בסיום שלב זה בוצעו האלגוריתמים הנבחרים על קטעי דיבור של 20-30 שניות כדי לקבל תוצאות שהן לא רק מתמטיות אלא מאפשרות האזנה והבחנה בטיב התוצאות.

2. שלב המימוש

שלב זה התנהל במקביל ובהמשך לשלב הסימולציות. בשלב זה מומשו 2 הסוגים של המערכת לזיהוי PITCH שנבדקו בסימולציות. לאחר מכן נבחנה מערכת שמבצעת את תהליך כיווץ/מתיחה ושלושת היחידות, יחידת כיווץ מתיחה, יחידת מיצוי ה-PITCH ויחידת ה-ADM צורפו למערכת אחת המהווה את המערכת לכיווץ/מתיחה הרמוני של אותו הדיבור.

בציר מס' 1 נתונה דיאגרמת בלוקים כללית של המערכת המוצעת. מתוארים המקלט והמשדר משני צידי הערוץ (יש לציין שהמערכת לא כוללת את היחידות הדרושות כדי לאפנן את הסיביות ולטדרם בערוץ קרי MODEM ומעגלי סינרון סיבית).

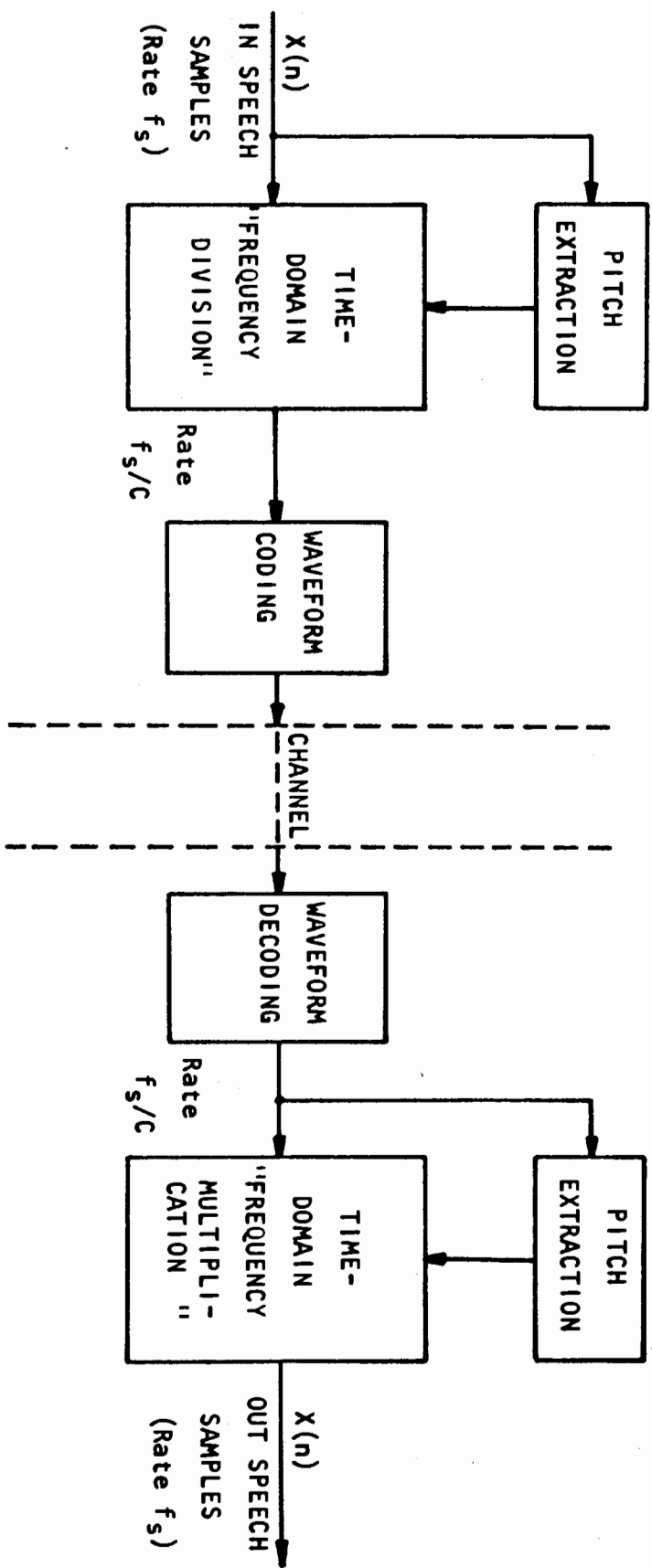
מקלט מקבל כאות כניסה דיבור הנדגם בקצב f_s עד ל-14.4KHz. האות מועבר ל-2 יחידות,

למציאה ה-PITCH ולמערכת הכיווץ. בממצה נעשית דצימציה ביחס של 1:2 ומהאות הדגום בקצב

$f_s =$ משוערך ה-PITCH. תוצאת PITCH מעודכנת מתקבלת כל כ-15msec ובהשהיה של 45msec

יציאת לדגם הכניסה (תוצאת ה-PITCH ביציאת הממצה היא לגבי הקטע שנדגם 30-45msec

קודם לכן).



דאגאראמא בלולקאס כללית של המערכת המוצעת.

Figure 1 Block diagram of proposed vocoder system.

אות הדיבור מגיע למערכת הכיווץ שם הוא מושהה 45msec (כדי ליצור התאמה בין ה- PITCH לדיבור) ולאחר מכן מתבצעת עליו פעולת כיווץ ביחס $1:2$ וביציאתו נקבל דגימות בקצב $f_s/2$ וחצי רוחב סרט של אות הכניסה. אות זה מועבר למערכת ADM שביציאתו נקבל סיביות בקצב $f_s/2$. זרם סיביות זה מועבר בערוץ למערכת המקלט. במערכת המקלט קיים ADM המפענח את זרם הסיביות וביציאתו נקבל דגימות בקצב של $f_s/2$. האות מועבר בצורה דומה לזו שנעשתה במקלט לממצה ה- PITCH ולאחר דצימציה של $1:2$, ערך ה- PITCH המשוער והדגימות נכנסים למערכת המתיחה. ביציאת מערכת המתיחה נקבל דגמים בקצב f_s וברוחב הסרט המקורי של אות הכניסה.

1.3 מבנה העבודה

העבודה תתחלק ל-3 חלקים עיקריים, אלגוריתם לכיווץ/מתיחה, סימולציות של המערכת ומימוש המערכת.

בפרק שני יתוארו האלגוריתם לכיווץ/מתיחה הרמוני של אותות דיבור ופיתוחו והתאמתו למימוש.

בפרק שלש יתוארו שיטות שונות לזיהוי PITCH, השיטות שנבדקו בסימולציות ותוצאות ומסקנות מהסימולציות לגבי מימוש גלאי ה- PITCH.

בפרק רביעי יתוארו מספר סוגי ADM שנבדקו במסגרת העבודה. ינתנו תוצאות סימולציות של הסוגים השונים והמסקנות שהביאו לבחירת ה- ADM למימוש.

בפרק חמישי יובאו תוצאות של סימולציות לכל המערכת כולה מדגימת הדיבור במשדר עד שחזרו במקלט.

בפרק ששי יתואר מימוש גלאי ה- PITCH.

בפרק שביעי יתואר מימוש יחידת הכיווץ/מתיחה ויחידת ה- ADM.

בפרק שמיני יובאו תוצאות של הפעלת המערכת, סיכום והמלצות לגבי שיפורים לעתיד.

$\Delta\omega$ נבחרה כך שהמסננים מכסים את התחום $[-\pi/T, +\pi/T]$ כאשר T הוא תחום הדגימה. ע"י שימוש בקונבולוציה דיסקרטית נקבל ביציאת המסננים את התוצאה הבאה :

$$(2) \quad y_k = 2 \sum_{r=-\infty}^n x(rT)h(nT-rT) \cos[\omega_k(nT-rT)] \\ = 2\text{Re}\{\exp[j\omega_k nT]X(\omega_k, nT)\} \quad , \quad k = 1, 2, \dots, L$$

כאשר

$$(3) \quad X(\omega_k, nT) \triangleq \sum_{r=-\infty}^n x(rT)h(nT-rT) \exp[-j\omega_k rT] .$$

$X(\omega_k, nT)$ היא התמרת פוריה הדיסקרטית לזמן קצר של אות הכניסה $X(nT)$. אם נביע את $X(\omega_k, nT)$ בצורה של אמפליטודה $|X(\omega_k, nT)|$ ופזה $\phi(\omega_k, nT)$ נקבל את הביטוי הבא:

$$(4) \quad y_k(nT) = 2|X(\omega_k, nT)| \cos[\omega_k nT + \phi(\omega_k, nT)] .$$

בשימוש של וקודר פזה (PHASE-VOCODER) מתקבלת חלוקת תדר במשדר והכפלת תדר למצבו המקורי במקלט ע"י כפל ב- q של כל ω_k ושל כל נגזרות של הפאזות. q במקלט שווה ל- $1/q$ של המשדר).

כדי להראות כיצד משפיעה ידיעת התדר הבסיסי של הדיבור (PITCH) על התהליך נניח סייגנל הכניסה הוא מחזורי עם מחזור בסיסי F_0 .

נניח כי התדר המשוערך של הסיגנל הוא F_p לעומת התדר המדויק F_0 . נבחר את Δf המרחק בין תדרי מרכז סמוכים להיות שווה ל- F_p .

כל מה שרטט יהיה גם הוא ברוחב של $\Delta\omega$ ומספרם L יהיה כזה ש- $\Delta\omega + L$ יהיה גדול או שווה לרוחב הסרט של סייגנל הכניסה.

נניח גם כי השגיאה בהערכה :

$$\Delta f/2L \quad \text{קטנה} \quad \Delta F_p = |F_p - F_o|$$

כלומר

$$(5) \quad \epsilon_p \triangleq \Delta F_p / F_p < 1/2L$$

בהתאם ל-STFT ובהנחה ש- $h(nT)$ הוא קירוב טוב לתגובת הלהם של מסנן מעביר נמוכים נקבל שהנגזרת $\dot{\phi}(\omega_k, nT)$ מתארת את המרחק של ההרמוניה ה-k-ית מהתדר המרכזי ω_k של התת סרט בו נמצאת ההרמוניה. על מנת להדגים את הדבר נבחר ℓF , $1 \leq \ell \leq L$, אם נוסחה (5) מתקיימת נקבל את ω_ℓ המתאימה להרמוניה הנ"ל ב- $\omega_\ell = \ell \cdot 2\pi F_p$ ובמישור הזמן

$$(6) \quad x_\ell(nT) = A_\ell \cos(\Omega_\ell nT + \phi_\ell)$$

הצבת (6) ל-(3) ופישוט התמרת פוריה של $h(nT)$ ל- $\exp(-j\omega\tau)$ בתחום $|\omega| < \Delta\omega/2$ יאפס ל- $|\omega| > \Delta\omega/2$ (מעביר נמוכים אידאלי עם השהיה בזמן של τ) נקבל

$$(7) \quad X(\omega_\ell, nT) = (A_\ell/2) \exp[j(\Delta\Omega_\ell(nT-\tau) + \phi_\ell)]$$

$$\Delta\Omega_\ell = \Omega_\ell - \omega_\ell$$

אשר

נקבל מכאן כי

$$\phi(\omega_\ell - nT) = \Delta\Omega_\ell(nT - \tau) + \phi_\ell$$

נגזרת הפאזה ניתנת עידי $\Delta\Omega_\ell$ שהיא המרחק מהמרכז של התת סרט ה- ℓ .

אם ההזזה היא קטנה כלומר שיערוך ה-PITCH הוא טוב הרי חלוקת והכפלה תדר יכולים להתבצע ע"י חלוקה והכפלה של תדירויות המרכז ω_k , בהזנחת נגזרות הפאזה. נקבל מ-(2), ע"י הזנחת נגזרות הפאזה את האות משונה התדר בקבוע q

$$(8) \quad y_k^q(nT) = 2\text{Re}\{\exp(jq\omega_k nT)X(\omega_k, nT)\}, \quad k = 1, 2, \dots, L.$$

בסכמה בצירור 2 מתארת את הפעולות הנדרשות לביצוע הנייל. בהמשך נראה פישוט ניכר לסכמה זו.

נחזור שוב לדוגמא לשם חישוב השגיאה בשיערוך ה-PITCH.

במצבה (7) ב-(8) נקבל

$$(9) \quad y_\ell^q(nT) = A_\ell \text{Re}\{\exp[j(\Omega_\ell^! nT + \phi_\ell^!)]\}$$

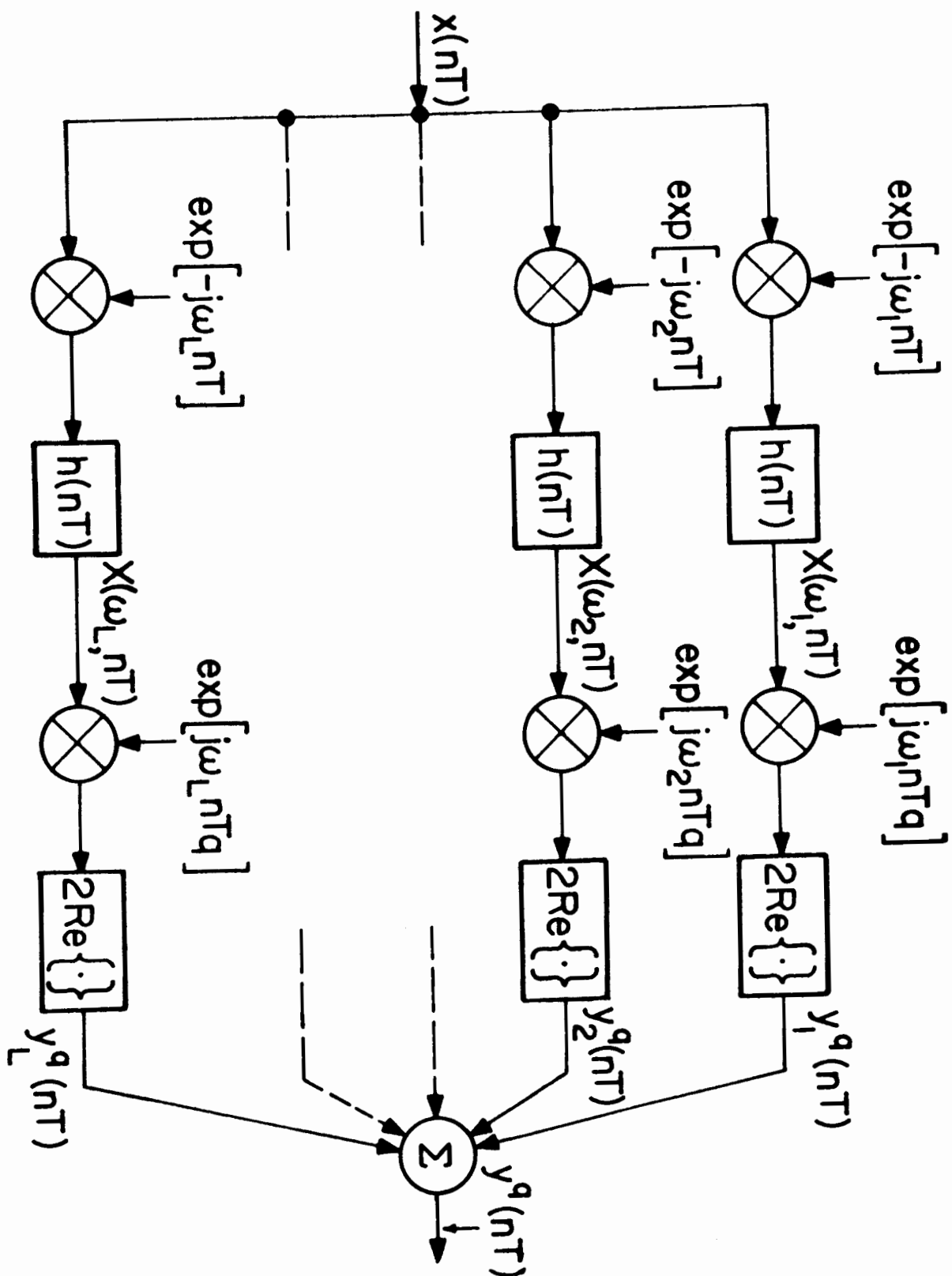
$$(10) \quad \Omega_\ell^! = q\omega_\ell + \Delta\Omega_\ell = q\Omega_\ell + \Delta\Omega_\ell(1-q)$$

$$(11) \quad \phi_\ell^! = \phi_\ell - \Delta\Omega_\ell\tau = \phi_\ell - 2\pi\ell(F_o - F_p)\tau.$$

היחסית המתקבלת היא

$$(12) \quad \epsilon_f \triangleq |\Omega_\ell^! - q\Omega_\ell|/\Omega_\ell = |\Delta\Omega_\ell(1-q)|/\Omega_\ell \approx \epsilon_p |1-q|.$$

$$F_o \gg \Delta F_p \Rightarrow |\Delta\Omega_\ell|/\Omega_\ell \approx \epsilon_p$$



סכמה מורכבת לכיווץ דיבור בקבוע q ע"י התמרת פורייה לזמן קצר.

Figure 2 A complex scheme for frequency scaling of speech signal by factor q via the short time Fourier transform.

לאות מחולק תדר C פעמים נקבל

$$(13) \quad \Omega_{\ell}^1 = \omega_{\ell}/C + \Delta\Omega_{\ell} = \Omega_{\ell}/C + \Delta\Omega_{\ell} (C-1)/C$$

אם נריץ אות זה C פעמים יותר מהר נקבל שכל הרמוניה תחזור לטרט התדר המקורי שלה
רק אם

$$(14) \quad \varepsilon_p = \Delta F_p / F_p < 1/(2C\ell) \quad ,$$

אם נכפיל אות זה ב-S נקבל

$$(15) \quad \Omega_{\ell}^1 = S\omega_{\ell} + C\Delta\Omega_{\ell} = S\Omega_{\ell} + (C-S)\Delta\Omega_{\ell} \quad .$$

$$\Omega_{\ell}^1 = S\Omega_{\ell} \quad \text{אם } C = S \text{ נקבל}$$

במצב הסיגנל במהירות קטנה פי S נקבל שכל הרמוניות יחזרו למקומם.

לפיכך אם M הרמוניות של סיגנל משוחזרות לאחר חלוקה והכפלת תדר באותו קבוע C
נני על ε_p שהוא השעור היחסי של הסטיה בין ה-PITCH המשוערך ל-PITCH האמיתי, לקיים
התנאי הבא

$$(16) \quad \varepsilon_p \leq 1/(2CM) \quad .$$

כעת על פיתוח של האלגוריתם המאפשר הקטנה ניכרת של מספר החישובים יחסית
ציר 2.

$$\text{נני } q \text{ בתחום } -1 \leq q \leq 3 \quad q \neq 0$$

$$(17) \quad q = \mu/\delta \quad ,$$

מספרים שאין להם מחלק משותף.

נניח כי קצב הדגימה הוא כזה ש:

$$(18) \quad T' = T/\mu .$$

כאשר T קטן שווה למרווח ניקויסט. נקבל ביציאות K הערוצים

$$(19) \quad y_k^q(nT') = \begin{cases} 2\operatorname{Re}\{\exp(jq\omega_k nT')X(\omega_k, nT')\} & , \quad k = 1, 2, \dots, L \\ X(0, nT') & , \quad k = 0 \end{cases} .$$

נחלק את תחום התדר $(-\pi/T', \pi/T')$ ל- N חלקים שווים במרווח

$$\Delta\omega = 2\pi/(NT')$$

כך שתדירות המרכז תהיינה $\omega_k = 2\pi k/(NT')$ כאשר $k = 0, \pm 1, \dots, \pm(N-2)/2$ (אי זוגי).

בבינה מחדש של נוסחה (3) עם T' התחליף ל- T כסכום של אוסף סכומים סופי על N דגימות של הסיגנל לאחר המסנן נקבל לפי [3]

$$(20) \quad X(\omega_k, nT') = W_N^{-kn} G(k, nT')$$

$$(21) \quad W_N \triangleq \exp(j2\pi/N)$$

$$(22) \quad G(k, nT') \triangleq \sum_{r=0}^{N-1} g(rT', nT') W_N^{kr}$$

$$(23) \quad g(rT', nT') = \sum_{i=0}^{\infty} x(nT' - rT' - iT')h(rT' + iT')$$

$$r = 0, 1, \dots, N-1$$

אם נסתכל על $G(k, nT')$ ב-(22) נראה שזו DFT על N נקודות ב- $g(rT', nT')$. נשים לב כי $G(k, nT') = G(N+knT')$ בגלל המחזוריות ב- k של W_N^{kr} . נניח עתה ללא כל הפסד בכלליות, כי מעביר הנמוכים הבטיסי הוא מסוג FIR, $h(nT) = 0$ ל- $n < 0$ ו- $n \geq M$ וכי $M = mN$. נקבל כי $g(rT', nT')$ מקבל את הצורה הבאה:

$$(24) \quad g(rT', nT') = \sum_{i=0}^{m-1} x(nT' - rT' - iT')h(rT' + iT'),$$

$$r = 0, 1, \dots, N-1$$

נבחר את מספר הערוצים L להיות שווה ל- $(N-1)/2$ ועייי הצבת (20) ו-(21) ב-(19) נקבל:

$$(25) \quad y_k^q(nT') = \begin{cases} 2\operatorname{Re}\{W_N^{nk(q-1)}G(k, nT')\} & , k = 1, 2, \dots, (N-1)/2 \\ G(0, nT') & , k = 0 \end{cases}$$

כי סיכום יציאות $C + 1$ ערוצים כולל זה של $k = 0$ יקבל הסיגנל $y^q(nT')$.
 צורה הבאה:

$$(26) \quad y^q(nT') = G(0, nT') + 2\operatorname{Re} \sum_{k=1}^{(N-1)/2} W_N^{nk(q-1)} G(k, nT')$$

הצורה הבאה בטור בפקטור q .

נגדיר מספר שלם p_c

$$(30) \quad p_c \triangleq n(\delta - \mu) \pmod{N}, \quad 0 \leq p_c \leq N-1 .$$

כיוון ש- W_N^ℓ מחזורי ב- ℓ אפשר לקבל במקום (29) את

$$(31) \quad y^{1/c}(nCT) = \sum_{k=0}^{N-1} G(k, nCT) W_N^{-kp_c} ,$$

כאשר $q=1/c$ ו- p_c הוגדר ב-(30).

צפי ההגדרה של $G(k, nCT)$ (ניתנת ב-(22)) רואים כי הצד הימני של (31) הוא התמרת צנריה ההפוכה של $G(k, nCT)$ ב- $N = p_c$ נקבל לכן:

$$(32) \quad y^{1/c}(nCT) = Ng(p_c T', nCT) = Ng(p_c T/\mu, nCT) ,$$

$g(\dots)$ נתונה לפי נוסחה (24).

נניח כי T_p הוא משך מחזור PITCH המשוערך ונניח כי

$$T_p = N_p T , \quad \text{שלם } N_p$$

פי הדיון הקודם תחום הסרט של הדיבור $(-\pi/T, \pi/T)$ מחולק ל- N_p תת סרטים כך ש-

היא כפולה של תדר ה- PITCH. ע"י נוסחה $T' = T/\mu$ וההנחה שהסרט מתחלק

→ לתת סרטים

$$(33) \quad N = \mu N_p .$$

נגדיר מחדש את p_c

$$(34) \quad p_c = n(\delta - \mu) - \alpha_c \mu N_p ,$$

כאשר α_c הוא החלק השלם של $n(\delta - \mu)/N$

$$(35) \quad \alpha_c \triangleq \text{Int}[n(\delta - \mu)/N_p \mu] = \text{Int}[n/N_c] ,$$

כאשר

$$(36) \quad N_c \triangleq \mu N_p / (\delta - \mu) = N_p (C - 1) .$$

בעזרת (34) ו-(35) נחשב את $p_c T'$

$$(37) \quad p_c T' = p_c T / \mu = nT(C - 1) - \alpha_c N_p T = nT_c - \alpha_c N_c T_c ,$$

כאשר

$$(38) \quad T_c \triangleq T(C - 1) = TN_p / N_c ,$$

ונגבל

$$(39) \quad p_c T' = (n \bmod N_c) T_c$$

ע"י שימוש ב-(24) ו-(32) נ-1 $T' = T/\mu$ $C = \delta/\mu$ נקבל:

$$(40) \quad y^{1/c}(nCT) = \sum_{i=0}^{m-1} s(nT + \alpha_c N_p T - i N_p T) h_N(i N_c T_c + (n \bmod N_c) T_c)$$

$$(41) \quad h_N(\cdot) \stackrel{\Delta}{=} Nh(\cdot) .$$

יש לשים לב כי נוסחה (40) ניתנת כפונקציה של אות הכניסה $x(nT)$ וללא צורך בדגימת יתר לפי T .
 ביציאה ניתנת בקצב של CT אולם יש לזכור כי גם רוחב הסרט של הסיגנל ביציאה קטן פי C ולכן אין בעיות של קיפול בגלל דגימה בקצב איטי מקצב ניקוויסט.
 נאגור את הסיגנל ונשמיע אותו לאחר מכן בקצב של $1/T$ נקבל סיגנל מכווץ בזמן (בדיבור ישמע בתחום התדר הנכון אך מהיר פי C פעמים) בעל אותו רוחב סרט של המקור.

2.2 סכילת תדר

במקרה זה נגדיר את S בתור כופל תדר רציונלי

$$q = \mu/\delta = S \quad S > 1 \Rightarrow \mu > \delta$$

במקרה הנ"ל דומה מאד למקרה של חלוקת תדר מלבד העובדה ש- $(\mu-\delta)$ הוא בסימן הפוך.

נגדיר את p_S המקביל - ב- (30)

$$(42) \quad p_S \stackrel{\Delta}{=} n(\mu-\delta) \bmod N, \quad 0 \leq p_S \leq N-1 .$$

כך מצגה ב- (29) נקבל

$$(43) \quad y^S(nT/S) = Ng((N-p_S)T/\mu, nT/S) .$$

במקרה דומה למה שנעשה בחלוקת תדר נקבל:

$$(44) \quad \alpha_s \triangleq \text{Int}[n/N_s]$$

$$(45) \quad N_s \triangleq N_p / (\mu - \delta) = N_p S / (S-1)$$

$$(46) \quad T_s \triangleq TN_p / N_s = T(S-1)/S ,$$

נקבל הנוסחה לאות מוכפל תדר

$$(47) \quad y^S(nT/S) = \sum_{i=1}^m x(nT - \alpha_s N_p T - i N_p T) h_N(i N_s T_s - (n \bmod N_s) T_s) .$$

משמיעים את $y^S(nT/S)$ בקצב S/T סיגנל היציאה נמשך אותו זמן כמו הכניסה אולם בעל רוחב סרט פי S גדול יותר. אם היציאה נאגרת ואז משמיעים אותה בקצב $1/T$ סיגנל נמתח פי S (הדיבור נשמע איטי) אך בעל אותו רוחב הסרט כמו המקור.

2. פונקציית החלון

בסיסיות מבנק המסננים הן שיהיו בעלי פאזה לינארית ובעלי תגובה יוניפורמית במליסודה בתחום הסרט של הדיבור. מדרישה זו נקבל [1]

$$(48) \quad h((\ell - m/2)N_p T) = \begin{cases} 1 & \text{for } \ell = m \\ 0 & \text{for } \ell \neq m \end{cases} ,$$

נוספת היא שאם נקבל בכניסה אות מחזורי במחזור $N_p T$ ו- N_p נבחר כך ש- $N_p = N_o$ הדיבורי בתדר עקב חלוקת תדר או הכפלתו צריך להיות מדוייק.

מקבלים מדרישה זו [1]

$$\sum_{i=0}^{m-1} h(t + i N_p T) = 1 , \quad 0 \leq t \leq N_p T .$$

$$\hat{h}(t) = w_m(t - nN_p T/2)$$

המבנים מתקיימים. עבור $m = 1$ מתקבל החלון המלבני הפשוט. עבור m זוגי הביטוי האנליטי לפונקציות החלון הנדונות הוא:

$$(49) \quad w_m(t') = \begin{cases} (\sin \pi t') / (m \sin \pi t' / m), & m \text{ odd} \\ (\sin \pi t') (\cot \pi t' / m) / m, & m \text{ even} \\ 0, & |t'| > m/2 \end{cases} \quad |t'| < m/2$$

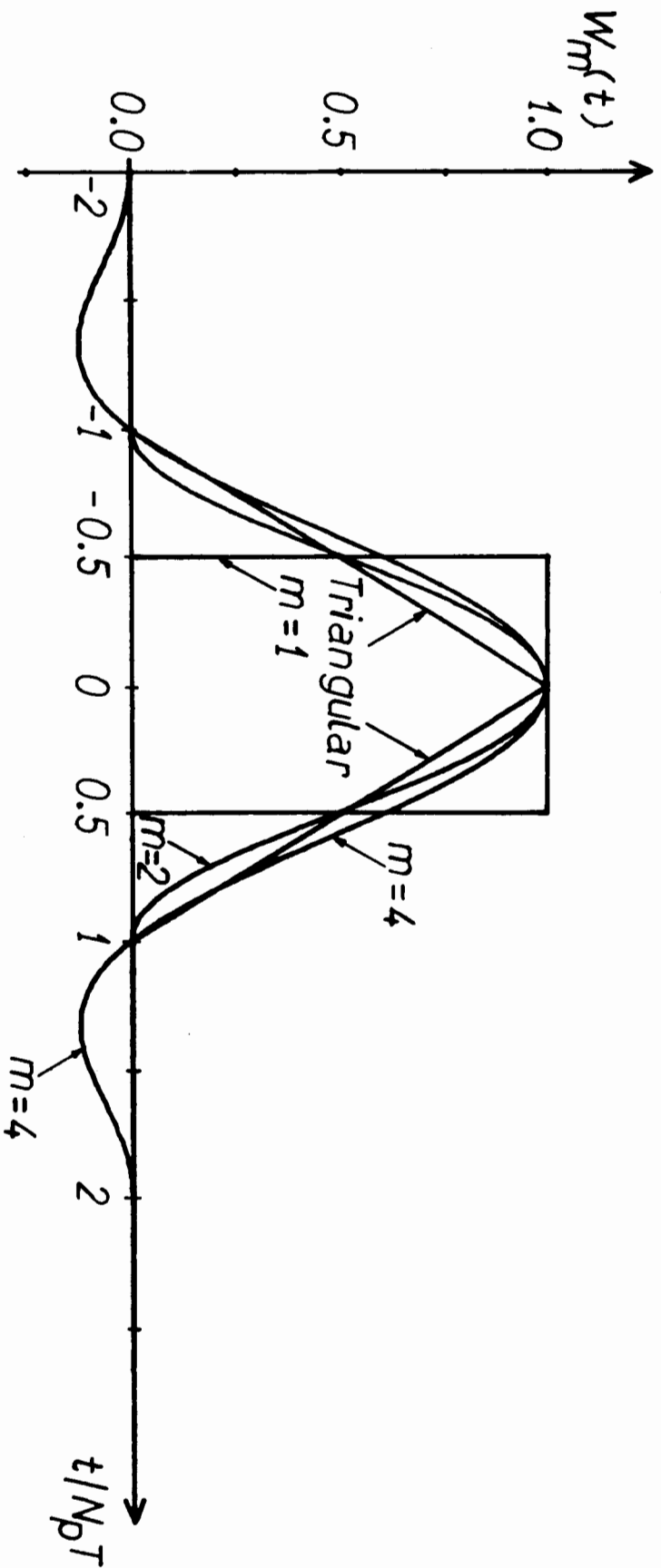
עבור $m = 1$ אי זוגי הפונקציה שונה במעט אולם היא אינה שימושית עקב אי הרציפות המתקבלת בצורת החלון. (דוגמת $m = 1$) זו אינה המשפחה היחידה המקיימת התכונות הנדרשות ונראה כי עבור $m=2$ גם חלון משולש מקיים התכונות הנדרשות.

עבור $m = 1$ כאמור מתקבל חלון מלבני פשוט בעל משך זמן $N_p T$ ואילו עבור $m = 2$ מתקבל חלון Hanning הידוע. בציורים 3, 4 בהתאמה נתונות הפונקציות עבור $m = 1, 2, 4$, עבור חלון משולש ותגובת התדר של פונקציות אלו. רואים בברור כי עבור $m = 2$ (חלון מלבני) תגובת התדר גרועה מאד עקב אונות הצד הגבוהות ועקב השפוע המתון של התדר בתחום המעבר מתדר הקטעון של המסנן מעביר הנמוכים ($wN_p T/2\pi = 0.5$) לתדר שני של המסנן הסמוך ($wN_p T/2 = 1$). עבור $m = 2$ רואים מציור 4 כי חלון Hanning טוב מהחלון המשולש בתחום ה- STOP-BAND מכיוון שאונות הצד נמוכות מאלו של חלון המשולש, אולם בתחום המעבר בין שני התחומים הנייל החלון המשולש טוב יותר.

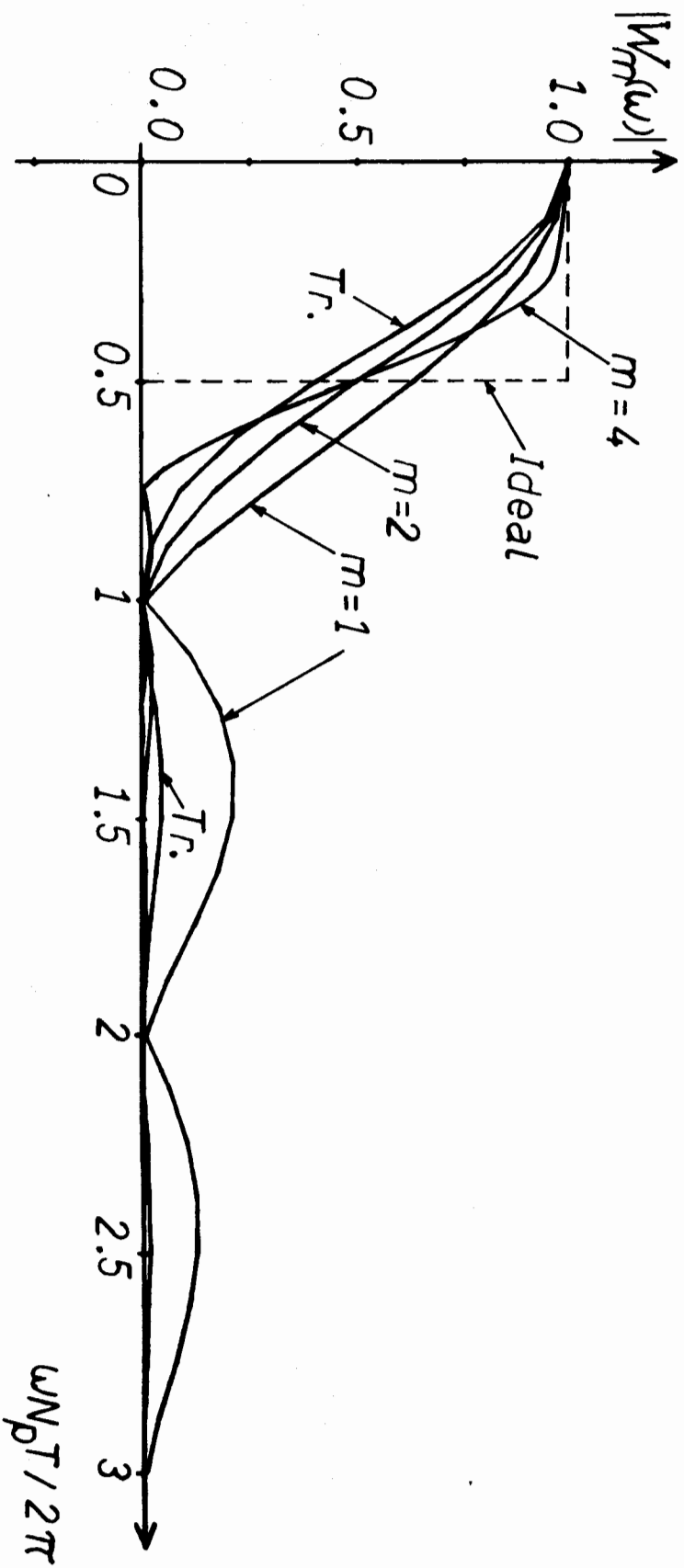
הצגתם של סימולציות [4] התברר כי חלון Hanning טוב במעט מהחלון המשולש אולם בניגוד את החלון בעיקר בגלל הפשטות במימושו ובמיעוט החישובים הנדרש.

ציור 5 נתון קטע דיבור והשיחזור שלו לאחר כיווץ ומתיחה עם שני החלונות השונים.

עבור $m = 4$ קיימת בעיה של שמירה על הרציפות בין קטעים סמוכים תוך כדי שינוי מספר מקטע לקטע, עקב העובדה שאות הדיבור הינו קווי סטציונרי ומשתנה בתחום



ציור 3. פונקציות החלון עבור $W_m(t)$ עבור $m = 1, 2, 4$ ועבור חלון משולש.
 Figure 3 The window functions $W_m(t)$ for $m = 1, 2, 4$ and the triangular window.



ציור 4 מגובה מדר של פונקציות הנמונות בציר 3 והמגובה האיזאלית הרצויה (קו עבה).
 Figure 4 Frequency response of the window functions shown in Fig. 3 and the
 desired ideal response (darked line)

20-40msec שהם מספר מחזורי PITCH. ואכן מתוצאות של סימולציות [4] השימוש ב- $m = 4$ נותן תוצאות פחות טובות מאשר עם $m=2$ שם לא נוצרות בעיות רציפות. (ראה תאור האלגוריתם הסופי).

זוהי גם שרצוי להשתמש בחלון בעל ה- m הקטן ביותר הנותן תוצאות טובות עקב העובדה ש- m הוא פרמטר הקובע את מספר החישובים עבור דגם כניסה בצורה לינארית. לסיכום: עבור האלגוריתם שמומש נבחר החלון המשולש גם בגלל ש- $m = 2$ וגם בגלל הפשטות במימוש החלון עצמו.

אלגוריתם הממומש

אלגוריתם הממומש נבחר $C = S = 2$ ונבחר חלון משולש. הדבר מפשט את האלגוריתם וצביא אותו לצורה הבאה

כור כיווץ:

$$(50) \quad y^{1/c}(\ell) = x(\ell)h_N(\ell) + x(\ell-N_p)h_N(\ell+N_c)$$

$$N_c = N_p \quad C = 2 \quad - \text{ } \square$$

הוא חלון משולש ולכן

$$(51) \quad h_N(\ell) = \ell/N_p = 1 - h_N(N_p - \ell)$$

הכל

$$(52) \quad \begin{aligned} y^{1/c}(\ell) &= x(\ell)h_N(\ell) + x(\ell-N_p)(1-h_N(\ell)) \\ &= x(\ell-N_p) + h_n(\ell)[x(\ell) - x(\ell-N_p)] \\ &= x(\ell-N_p) + \ell/N_p[x(\ell) - x(\ell-N_p)] \end{aligned} \quad \ell = 0, 1, \dots, N_p-1$$

גזור מתיחה עם $N_s = N_p$ $S = 2$ ונקבל:

$$(53) \quad \begin{aligned} y^S(\ell) &= x(\ell - N_p) h_N(N_s - \ell) + x(\ell - 2N_p) h_N(2N_s - \ell) \\ &= x(\ell - 2N_p) + h_N(N_s - \ell) [x(\ell - N_p) - x(\ell - 2N_p)] \end{aligned} \quad \ell = 0, 1, \dots, N_s - 1$$

הגשים לב למספר נקודות:

• התהליך של כיווץ ומתיחה הינו פשוט מאוד ודורש 2 פעולות חבור ופעולת כפל אחת לדגם יציאה.

• התהליך נעשה בקטעים. לאחר קבלת ערך PITCH מגלאי ה-PITCH תתבצע הפעולה על חלון שלם ולאחר מכן קוראים ממשערו ה-PITCH ערך נוסף ומבצעים את התהליך שוב עם הערך החדש.

• פונקציית החלון גם הוא פשוט ודורש פעם במחזור (ביצוע הפונקציה לאורך מחזור $1/N_p$ שלם) חישוב של $1/N_p$ מספר הדגמים במחזור PITCH אחד). ולאחר מכן כל דגם יש לבצע

$$(54) \quad h_N(\ell) = h_N(\ell - 1) + 1/N_p$$

• 6, 7 מתוארים התהליכים של כיווץ ומתיחה עם PITCH קבוע עבור $C = 2, 3$ $S = 2, 3$ בהתאמה.

• 8 מתוארים התהליכים של כיווץ ומתיחה עם PITCH משתנה בזמן עבור $C = 2$ $S = 2$.

• 9 מתואר קטע מהמלה שלוש לפני ואחרי כיווץ ב-2.

• 10 נתונות הספקטוגרמות של המלה שלוש לפני ואחרי כיווץ.

• 11 ניתנות ספקטוגרמות של כיווץ ומתיחה (של המקור) של קטע דיבור באנגלית.

בניורים 12,13 נתונים קטעי דיבור מהמלה "שלום" לאחר כיווץ ומתיחה חזרה עם

$C = 1$ ו- $C = 3$ בהתאמה.

בניור 14 ניתנות ספקטוגרמות של המשפט "שלום על ישראל" במקור ולאחר כיווץ/מתיחה

בגודל 3.

בייחוס הדיון בפרק זה נציין כי האלגוריתם בצורתו הסופית מאפשר לממש בצורה פשוטה

בעזרת המקטינה את רוחב הסרט של סיגנל הדיבור פי שנים ולאחר השידור בערוץ למותחו

דריה לרוחב הסרט האוריגינלי.

יש לציין כי טיב השיחזור והכיווץ תלוי במידה רבה במשעך ה- PITCH שהיווה את

אחת הבעיות הקשות של עבודה זו. אומנם האלגוריתם מאפשר סטיה בשיערוך ה- PITCH

בניור של

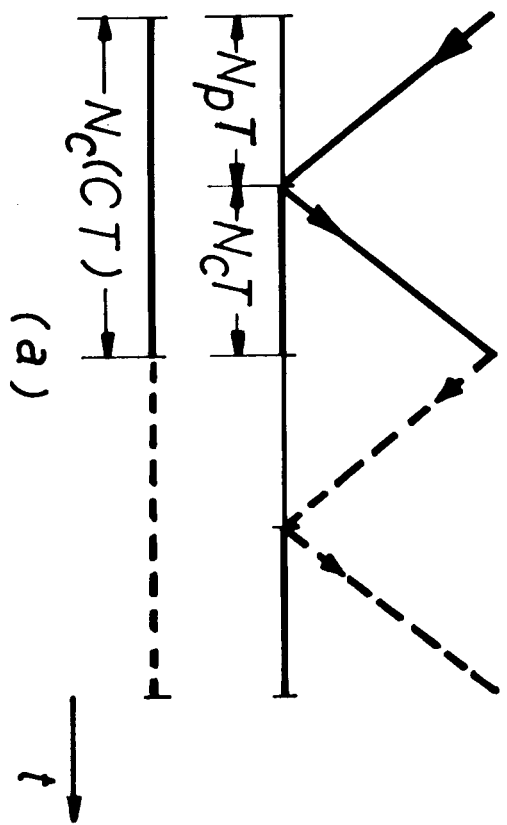
$$\varepsilon_p \leq 1/2CM$$

אלו סטיות של אחוזים בודדים ולא סטיות גסות. לדוגמה עבור $M = 10$ לכסוי

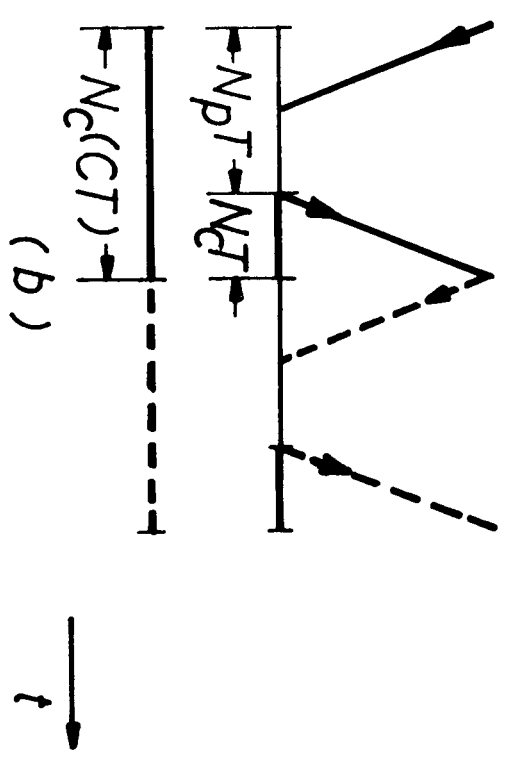
בפירמנט הראשון הכולל את מרבית האנרגיה של האות $C = 1-2$ נקבל שנדרש דיוק

2.5.

$C=2$, $m=2$



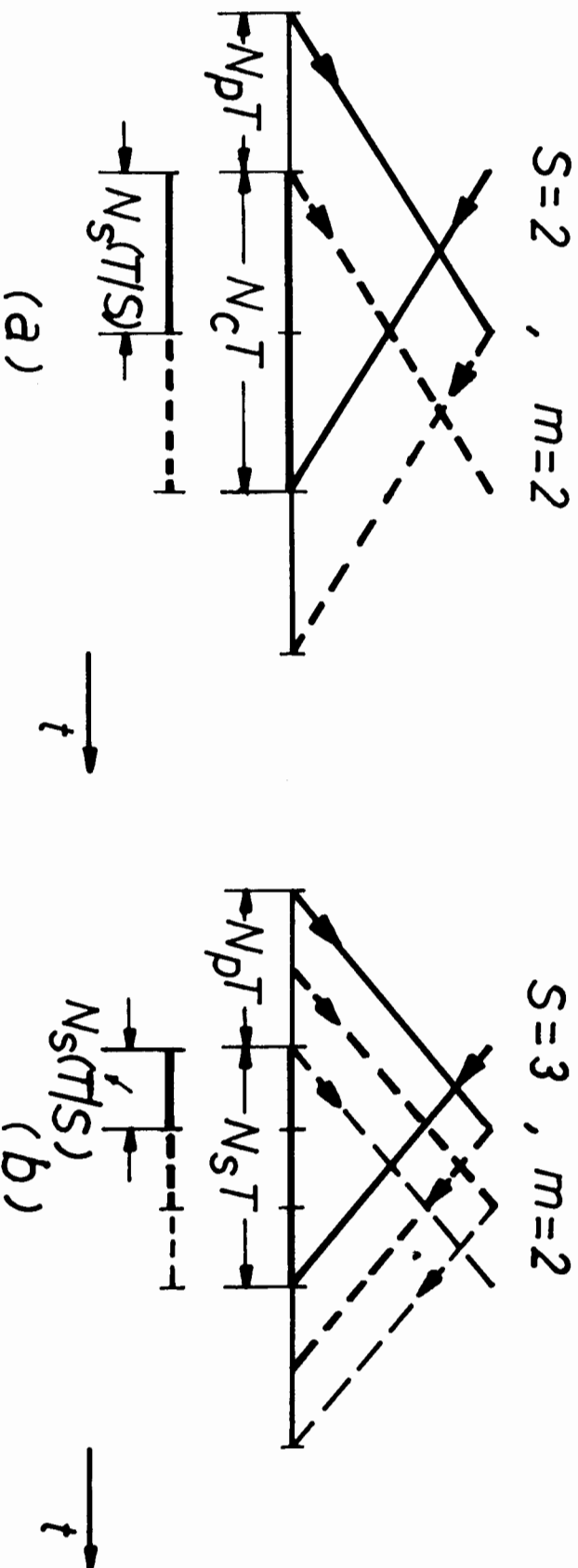
$C=3$, $m=2$



ציור 6 מבנה אלגוריתם הכיורוי בתחום הזמן במבנה של קטע לאחר קטע.

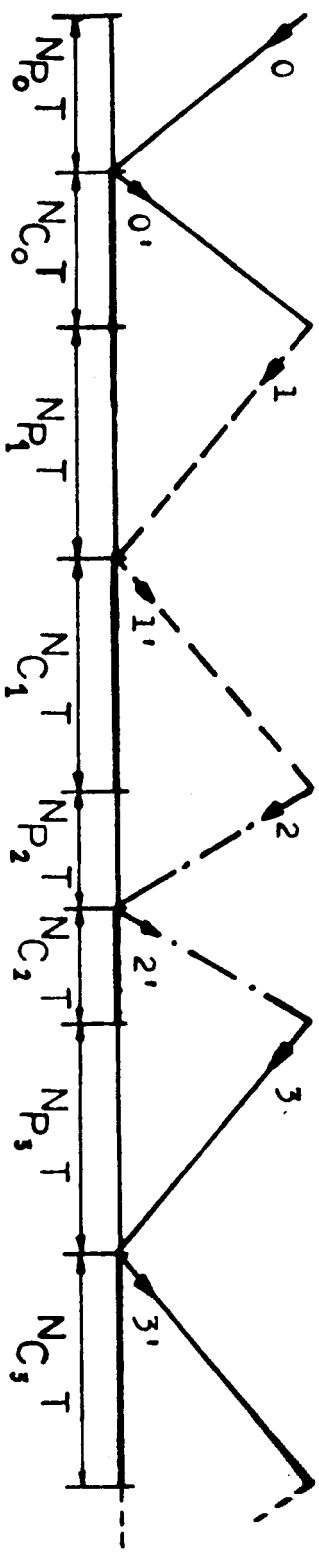
השימוש ב-2 מ ובחלון משולש מוצג עבור מקדמי כיורוי 2 ו-3. (b).

Figure 6 Organization of the time-domain algorithm for frequency division on a segment by segment basis. The use of $m = 2$ and a triangular window is shown for scaling factors of (a) $C = 2$; (b) $C = 3$.



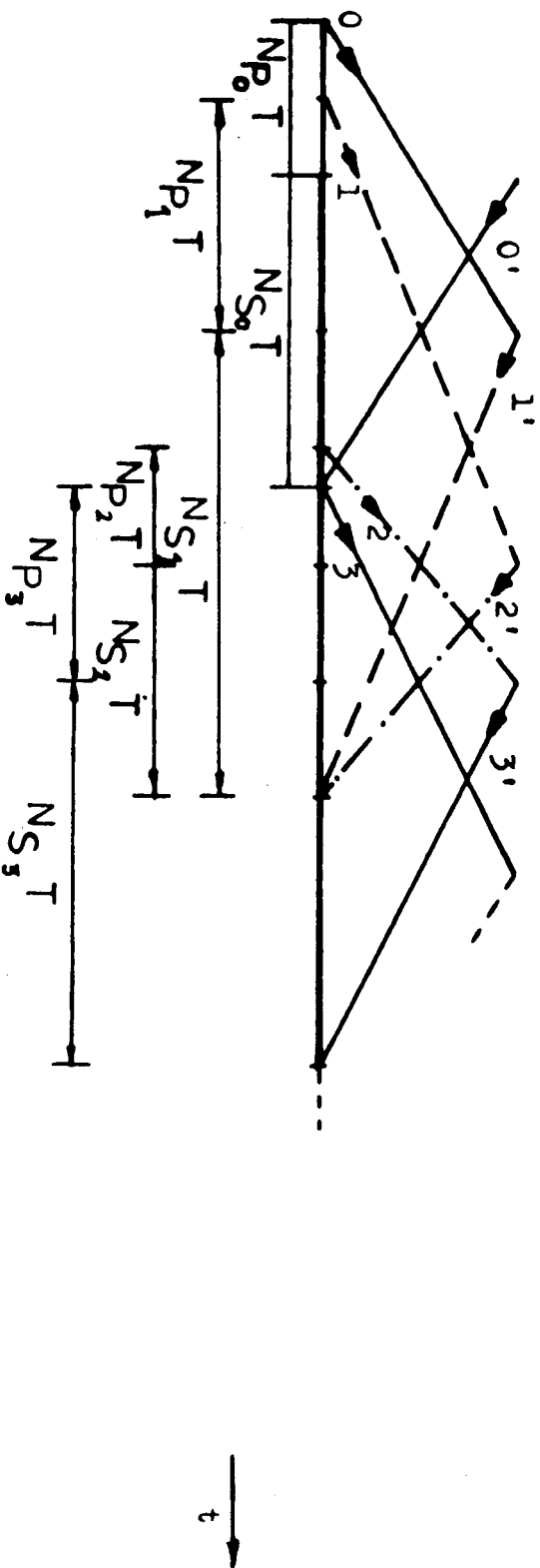
ציור 7 מבנה אלגוריתם המתיחה במחום הזמן במבנה של קטע לאחר קטע. השימוש ב- $m = 2$ ובחלון משולש מוצג עבור מקדמי כינון 2 (a) ו-3 (b).

Figure 7 Organization of the time-domain algorithm for frequency multiplication on a segment by segment basis. The use of $m = 2$ on a triangular window is shown for scaling factor of (a) $S = 2$; (b) $S = 3$.



$S = 2 \quad m = 2$

t

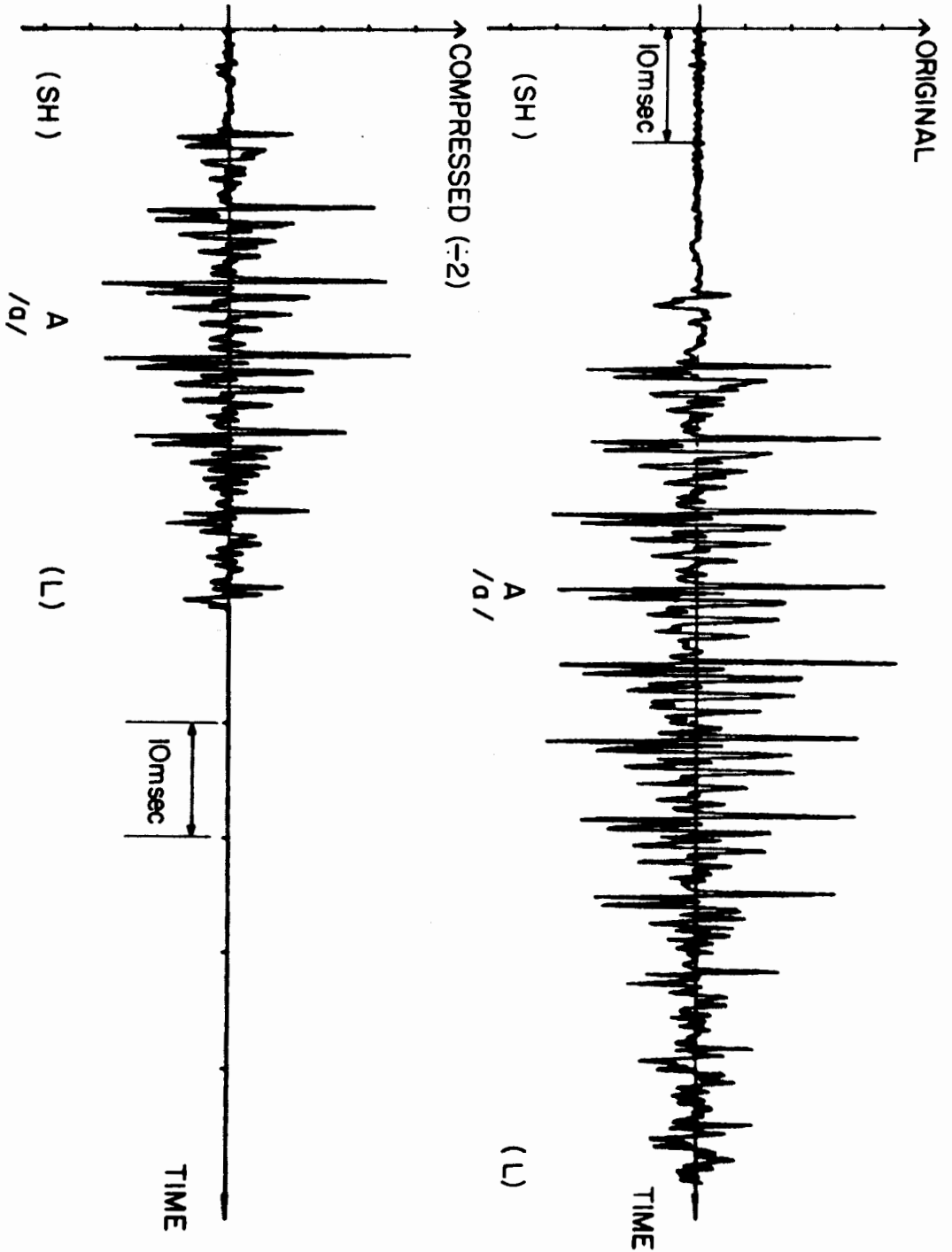


t

האלגוריתמים שחוארו בגיוריים 6,7 נהווים עבר PITCH משונה בזמן.

ציור 8

Figure 8 The algorithms shown in Fig. 6 and Fig. 7 are given for changing in time PITCH.



תמונה 9
 גרורות מליים של אות מקור ואות מכווץ בקטור 2 בתחום הזמן של ההברה הראשונה
 באות מלוט.

Figure 9 Original and time-compressed waveforms of the voice portion in the first syllable of the Hebrew word "Shalosh".

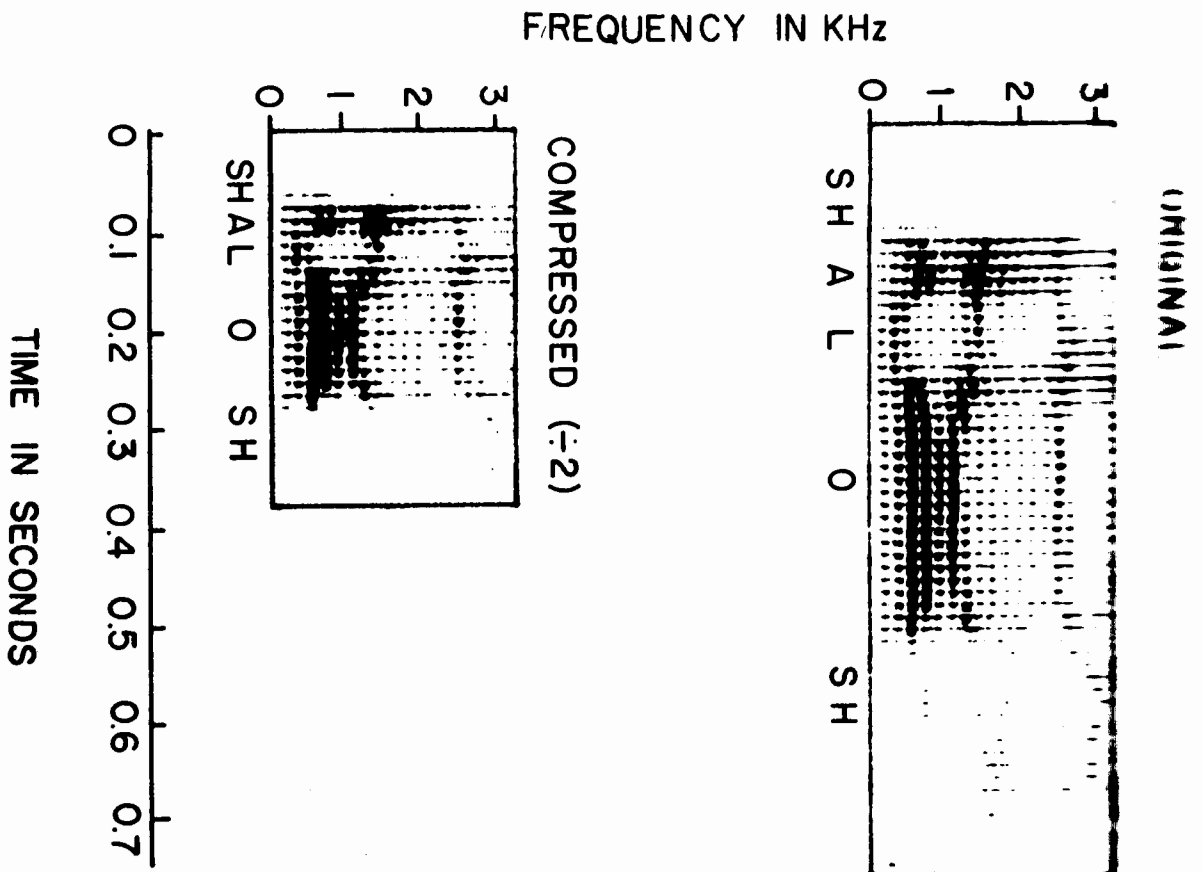


Figure 10 .קטוגרמות של המלה שלוש לפני ואחרי כיווץ ב-2 עם חלון משולש. Spectrograms of original and time-scaled Hebrew word "Shalosh".

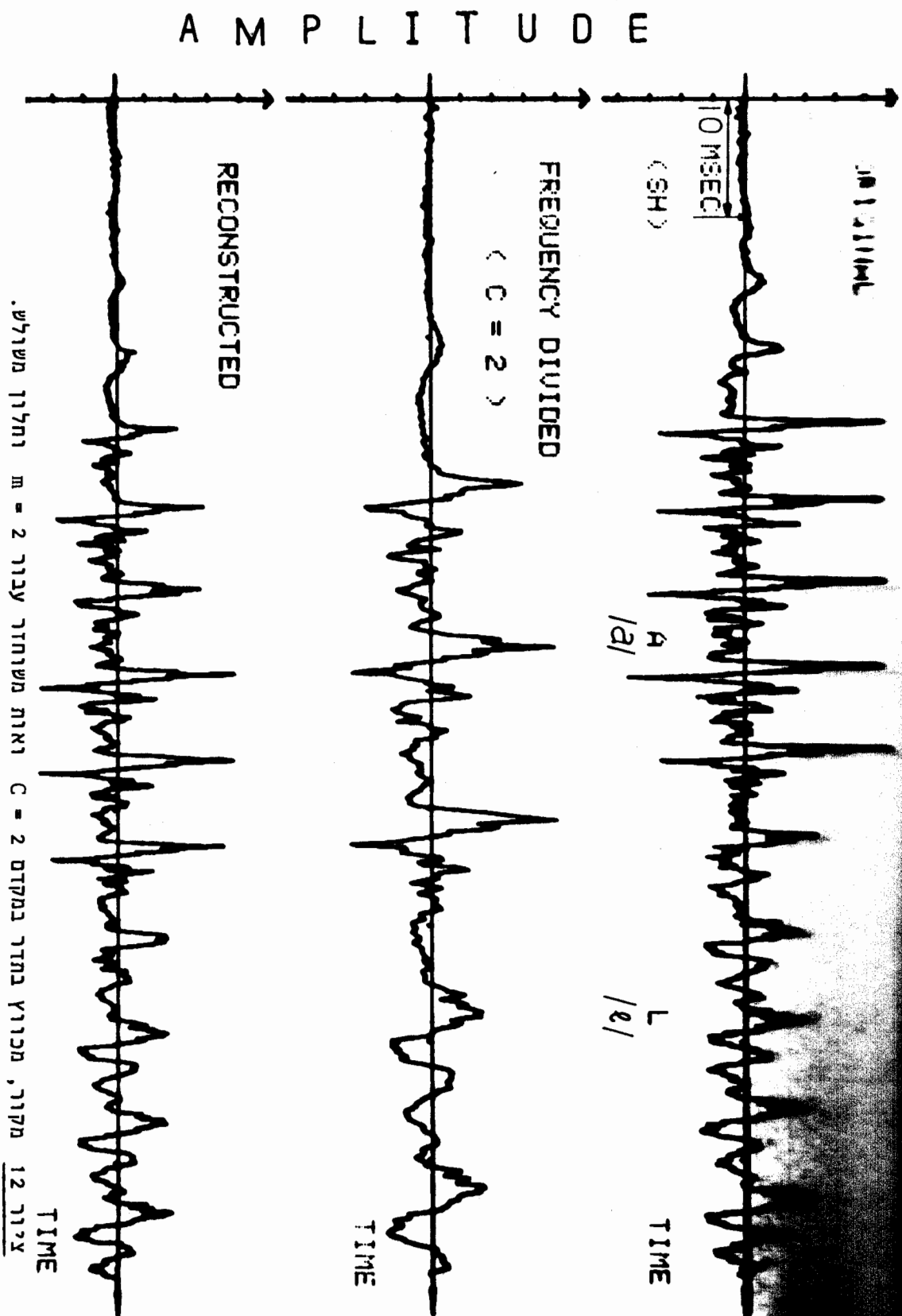
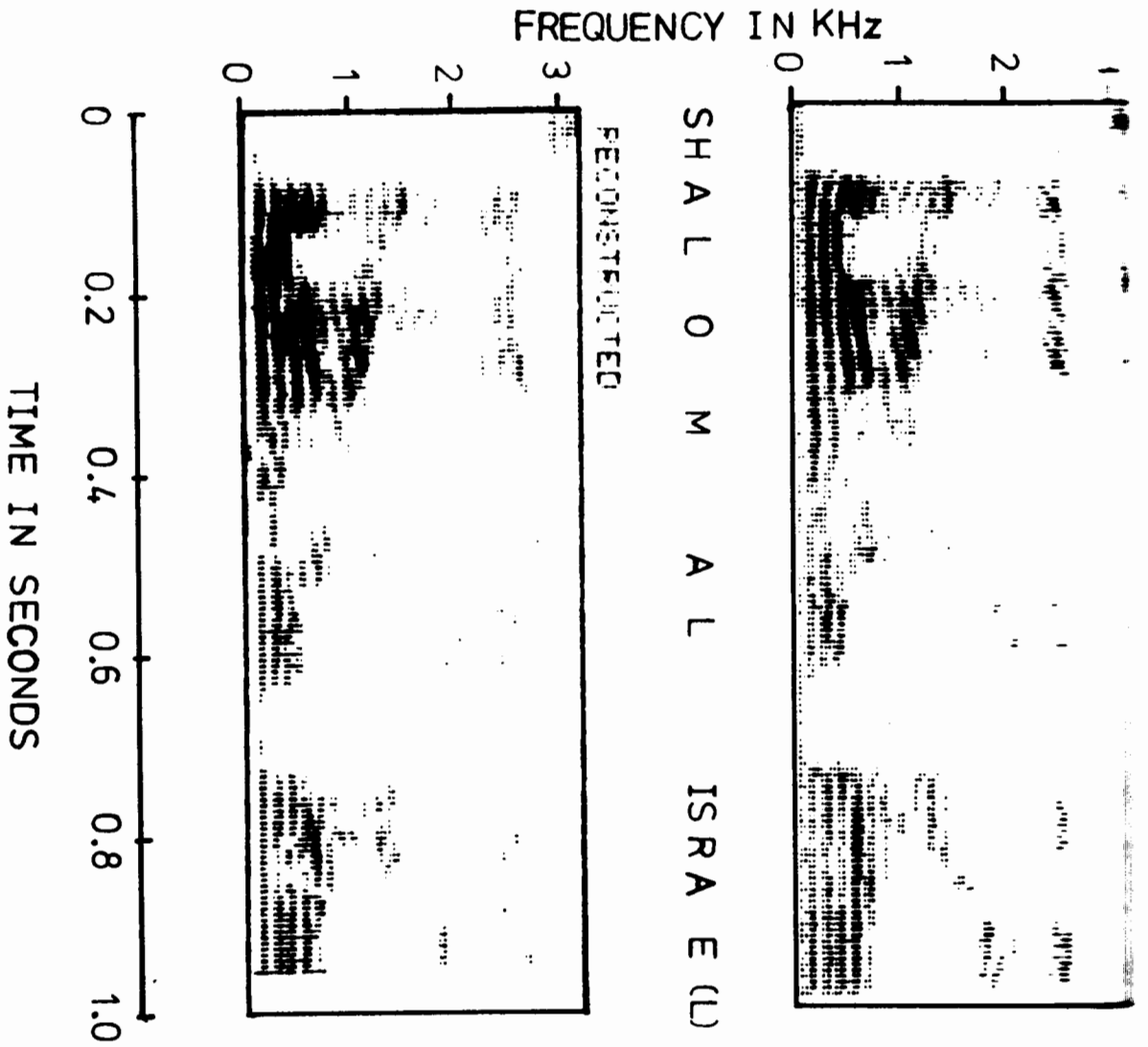


Figure 12 Original, frequency divided by a factor $C = 2$ and reconstructed signals as obtained for $m = 2$ and a triangular window function.



14 11

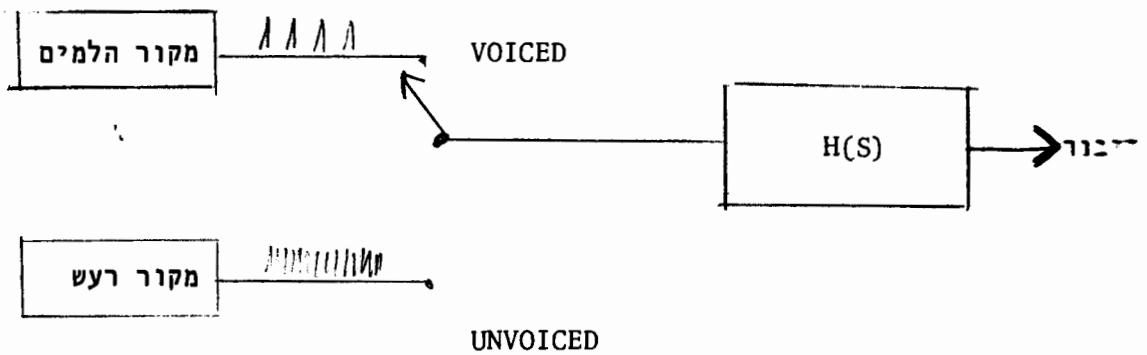
המחזור של מקור ואות משוחזר לאחר כיווץ בד של המשפט "שלום על ישראל".

Figure 14 Spectrograms of original and reconstructed sentences (in Hebrew) ~~5.11.14~~ frequency division and multiplication by factor 3.

פרק 3 : סימולציות למערכת מיצוי ה-PITCH

הדגמה

תפקיד היסודי של מנגנון הדיבור מתואר בשרטוט הבא:



דיבור מתחלק באופן בסיסי לשני סוגים

~~בסיסי~~ - במקרה זה מעוררים את $H(S)$ ברכבת הלמים במחזור השווה לתדר הבסיסי של הדיבור ה-PITCH.

~~UNVOICED~~ - במקרה זה מעוררים את $H(S)$ במקור רעש.

~~בסיסי~~ $H(S)$ מתארת את העבוד המתבצע על האויר הנוצרים במתריהקול (במקרה של

~~בסיסי~~) ובגרון ע"י דרכי הקול הגרון הפה הלשון והאף.

~~בסיסי~~ המחזור של ה-PITCH נע בתחום 3-16msec. אצל ילדים קטנים אפשר למצוא

~~בסיסי~~ של PITCH של 2msec ואילו אצל גברים בעלי קול עבה במיוחד אפשר למצוא

~~בסיסי~~ של עד 20msec. אות הדיבור מכיל כמובן גם הרמוניות גבוהות יותר ולכן

~~בסיסי~~ לעבירו דרך מסנן מעביר נמוכים לפני מיצוי ה-PITCH.

~~בסיסי~~ למיצוי ה-PITCH על המערכת הנייל להחליט האם הדיבור הוא מסוג של VOICE

~~בסיסי~~ UNVOICED וזאת ע"י מספר פרמטרים שהמערכת מפיקה מהדיבור.

דימות שיטות שונות למציאת PITCH, חלקן בתחום התדר, חלקן בתחום הזמן וחלקן בזוויות.

בתחום התדר קיימת למשל שיטת CEPSTRUM [5] המבוססת על התמרה לא לינארית $DET^{-1}[\text{LOG}(\text{ABS}(S))]$ של הספקטרום של אות הדיבור המנותח. שיטות המבוססות על תחום התדר אינן מעשיות למימוש סביר בגודלן כי הן דורשות ביצוע התמרת פוריה על אות הדיבור בזמן אמיתי, דבר שהיה דורש הגדלת החומרה בצורה גודל.

שיטות מעורבות זמן-תדר. לדוגמה השיטה בה נעזרים ב-L.P.C. כדי לשערך את הפרמטרים של קוטבי H(S) ובעזרת זאת מקבלים את $[H(S)]^{-1}$. מפעילים אותה על הדיבור ומקבלים את שגיאת החיזוי. על סדרה זו מבצעים אוטוקורלציה ומקבלים את ה-PITCH [5], גם שיטה זו צורכת זמן חישוב וכמות חומרה גדולים ואינה מתאימה למימוש סביר למערכת בזמן אמיתי.

ביצוע האותות בתחום הזמן אינו דורש ביצוע DFT ולכן הוא המתאים ביותר למימוש בזמן אמיתי. האלגוריתמים בתחום הזמן מבוססים בעיקר על אוטוקורלציה. הצורה המינימלית ביותר היא בצוע אוטוקורלציה וחיפוש מקסימום על פונקציית האוטוקורלציה. שיטה זו דורשת מספר כפלים רב. בקצבי הדגימה וברוחב החלון שמומשו בעבודה היינו מקבלים עבור קצב דגימה של 7.2KHz וחלון של 100 דגמים כפל כל 1.4μ או 100 דגמים עבור כל דגם. בעיה זו ניתנת למימוש עם המכפלים המהירים (והיקרים) קיימים היום אולם היא היתה משלשת את כמות החומרה במערכת האוטוקורלטור. פתרון זה זמן החישוב הנדרש מהמיקרו-מחשב לטיפול באוטוקורלטור. קיימים אלגוריתמים המקטינים את כמות הכפלים בחישוב פונקציה שהיא כעין אוטוקורלציה AMDF [AVERAGE MAGNITUDE DIFFERENCE FUNCTION] [5].

במחום הזמן היא הפעלת אלגוריתם על אות הדיבור המקטין את כמות החישובים לסיבית אחת או שתיים ולאחר מכן לבצע את פעולת הכפל שהופכת כמובן את פעולת XOR ו-OR במקביל. גישה זו נוחה למימוש ומאפשרת ביצוע

אלגוריתם בתוכנה ומשאירה למימוש בחומרה ביצוע של פעולות לוגיות בלבד ללא כל פעולות כפל וחיבור כפי שנדרש באוטוקורלציה מהירה.

בעזרת העבודה נחקרו ואף מומשו אוטוקורלטורים עם שני אלגוריתמים אפנון דלתא .CENTER CLIPPING

ב-3 אפנון דלתא [7]

כאשר זהו מקבלים ביט אחד e_n עבור כל דגם

$$e_n = \text{SIGN}(\bar{X}_{n-1} - X_n)$$

(55)

$$\bar{X}_n = \bar{X}_{n-1} + e_n \cdot \Delta_0$$

X_n - הדגם ה- n

\bar{X}_n - האות המשוחזר ה- n

Δ_0 - גודל הצעד של אפנון הדלתא

e_n - תוצאת אפנון הדלתא יכול לקבל את הערך +1 או -1

כאשר נעביר את אות הדיבור דרך מאפן דלתא נקבל באות מסוג VOICED (קולי)

האות של +1 ו-1 החוזרות על עצמן בצורה די דומה. כאשר נעביר אותם

באוטוקורלטור נקבל פונקציית אוטוקורלציה שמחזורה זהה למחזור ה- PITCH

שיקום במקסימום של פונקציה זו יהיה זהה למחזור ה-PITCH.

במקרה זה Δ_0 במידה זה Δ_0 קטנה יתר על המידה נקבל זיהוי

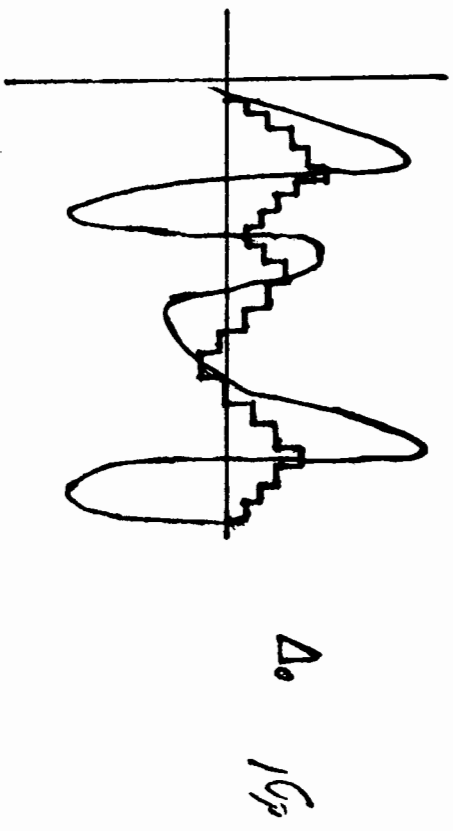
של מחזורים שהאפנון לא יבדיל ביניהם לבין המחזור עצמו. במידה ו- Δ_0 גדולה

של נקבל סדרות של +1 ו-1 אלא סדרה אקראית. ציור 15 מבהיר בעיות אלו.

הבעיה נובעת הוא להכניס השהיה של מחזור בין הדגימות לבצוע האפנון ועל סמך

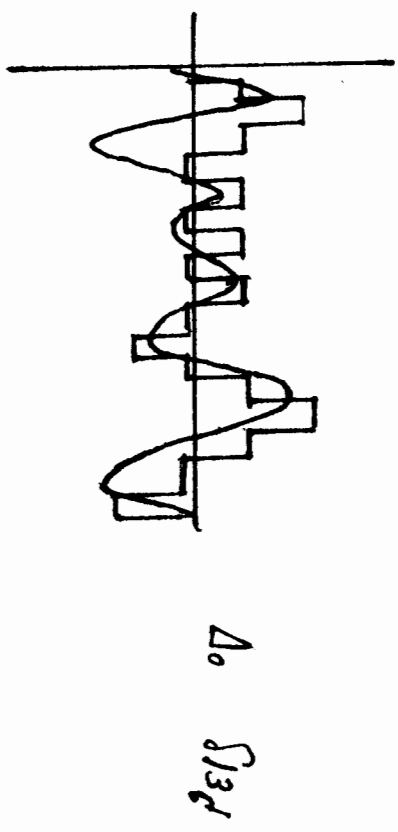
האות במחזור להחליט על גודלה ה- Δ_0 . דבר זה הופך את Δ_0 לאדפטיבית לגודל

של המחזוריות תתגלה לכל עוצמה של אות.



צירור 15 הנבייה שגורם השימוש ב- Δ_0 קבוע בגלאי ה- PITCH.

Figure 15 The problems that fixed Δ_0 cause in PITCH detector.



בשיטה זו מקבלים עבור כל דגם 3 רמות $-1, 0, +1$ דבר הדורש הקצאת זוג ביטים עבור כל דגם.

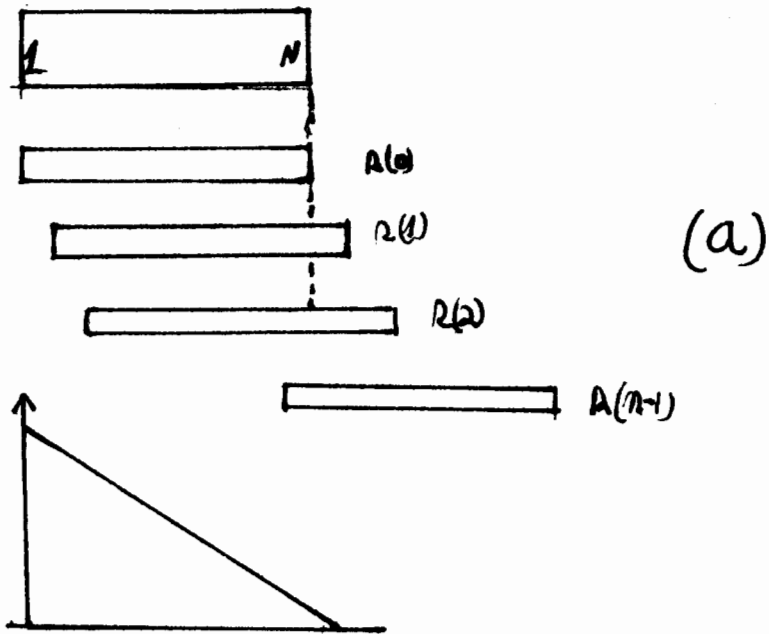
$$(56) \quad C_n = \begin{cases} -1 & -X_n > CLV \\ 0 & |X_n| < CLV \\ +1 & X_n > CLV \end{cases}$$

X_n - הדגם ה- n
 CLIPPING LEVEL - CLV
 C_n - התוצאה ה- n -ית

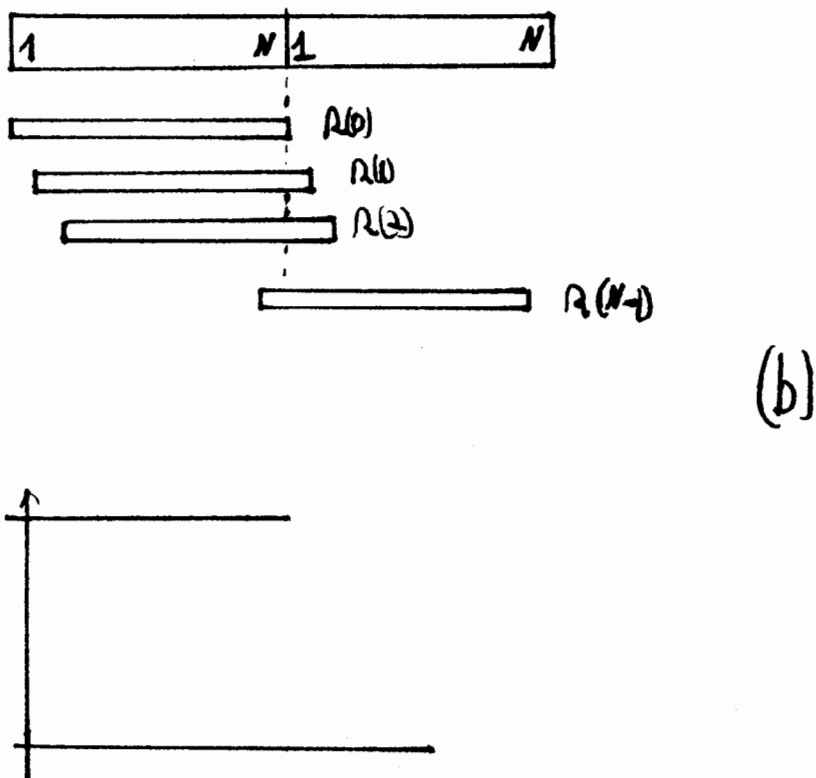
כך רואים ש- CLV הוא הפרמטר החשוב באלגוריתם. בשיטה הקלאסית [5] הוא נקבע במקסימום הנמוך ביותר בשלושת הקטעים האחרונים בדרך כלל 40%-60% ממנו. זו ביצוע אוטוקורלציה דורש ביצוע xOR ו- OR במקביל בניגוד ל- xOR בלבד ולכן דלתא וכן דורש הכפלת זכרון האוטוקורלטור מכיון שכל דגם בו דורש יצוג של 2 ביטות.

שיטות העבודה היא בבלוקים. דוגמים בלוק ומבצעים עליו אפנון DELTA שיטת CLIPPING לאחר החלטה על גודל Δ_0 או CLV . לאחר מכן מעבירים את התוצאה שיטת אוטוקורלציה ולפי המקסימום מחליטים על גודל ה-PITCH וחוזר חלילה. היה לנו קולרציה ציקלית או לא ציקלית. בקורלציה ציקלית יכולה להיות שגיאה כיוון שיש יחס שלם בין רוחב החלון למחזור ה-PITCH ואילו בקורלציה שאינה ציקלית יש נזק בגלל הקטנת החפיפה בין חלקי הבלוק דבר שגורם להקטנת החלון (ראה יחס שלם) במרום שנבחר הוא הזזת בלוק ברוחב החלון על פני שני בלוקים ועיי כן על ביטות.

הערה על דיבור לא מכווץ התקבלו תוצאות טובות בשתי השיטות באזורים הקוליים של דיבור. מנגנון דלתא התקבלו תוצאות לא טובות ושגיאות גסות באיזורי המעבר בין



ציור 16 ביצוע אוטוקורלציה עם ניחות (a) וללא ניחות (b). Figure 16 Autocorrelation with tapering (a) and without



קולי לשקט ובין אזור קולי לאזור לא קולי בגלל אקראיות הפונקציה שם. לצורך
 PITCH מדיבור לא מכווץ פשוט יותר היה להשתמש באפנון דלתא אולם במקרה שלנו
 הבעיה האם לשדר בערוץ את ערך ה-PITCH יחד עם הדיבור המכווץ
 בהמשך בפירוט אולם טוב היה אילו היינו יכולים להמנע מכך. דבר זה אומר
 PITCH במקלט על האות המכווץ. האות המכווץ מכיל קטעים רצופים של דיבור
 במקדס הכיווץ מהאות המקורי ובמקרים רבים נמצא 3-5 מחזורים
 בלבד. במקרים אלו יש חשיבות רבה לתופעות הקצה שהן רבות יותר ויכולות
 בשיטה של CENTER CLIPPING האיזורים השקטים והאיזורים
 בתחום ה- $C_n=0$ כך שהפרעות הרבה יותר קטנות. מסיבה זאת על
 יותר מסובך נבחר אוטוקורלטור המבוסס על CENTER CLIPPING.
 ה-PITCH צריך להחליט גם על אופי הקטע האם הוא קולי או לא והאם יש בכלל
 בערוץ - SILENCE. החלטות אלו מתבצעות לפי אנרגיית האות, מספר חציות
 ההתאמות המקסימלי בפונקציית האוטוקורלציה. שלושת גורמים אלו מאפשרים
 בין טובה בין סוגי הדיבור השונים. האלגוריתם של CENTER CLIPPING
 בו בשלב הראשון של הסימולציות זהה לאלגוריתם המופיע במאמר [6]
 מספר חציות אפס. האלגוריתם מנמש בתוכנית PITCHD והוא מתואר
 17.

זה נבדקו מספר אספקטים:

ביקת האפשרות למנוע שידור אינפורמציה ה-PITCH ומצוינו במקלט ובאיזה מקדס
 כיווץ הדבר אפשרי.

ביקת הדרישות מממצה ה-PITCH.

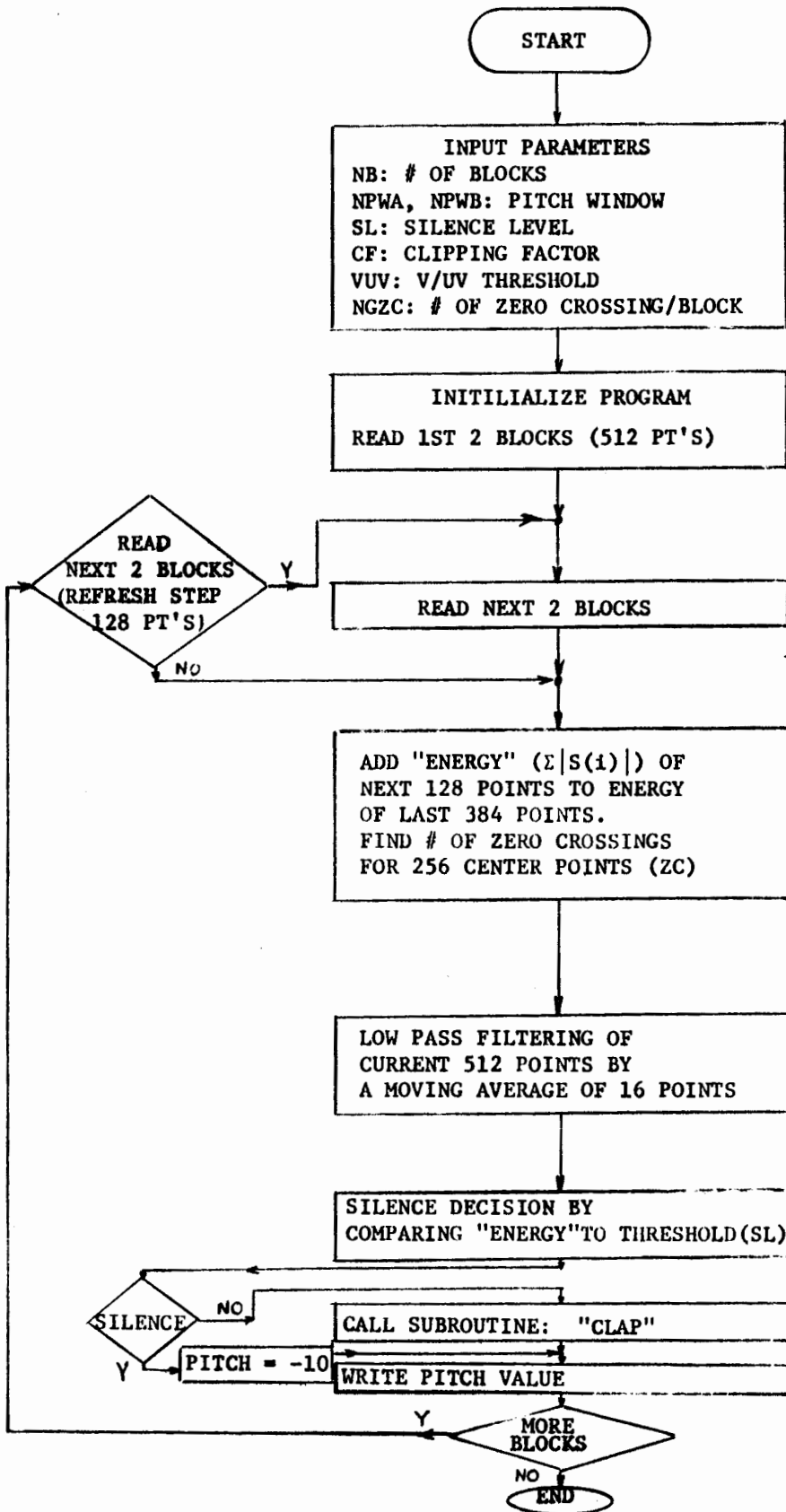
שידור אינפורמציה ה-PITCH

שידור אינפורמציה ה-PITCH מפשטת בהרבה את המערכת. שידור כזה היה

שידור של אינפורמציה ה-PITCH במסגרות כל 20msec והיה צורך בסינכרון

היה גורם הגדלת הרגישות לשגיאות בערוץ ולהגדלת קצב התמסורת.

FIG 17 FLOW CHART OF PITCH EXTRACTION PROGRAM - PITCHD



SUBROUTINE CLAP

- SEARCH FOR NEXT PEAK (IN SEGMENT OF 128 PT'S)
- DETERMINE NO. OF ZERO CROSSINGS IN WINDOW OF 512 PT'S
- LEVEL ACCORDING TO CF* (SMALLEST PEAK IN WINDOW OF 512 PT'S)
- CENTER AND IMPULSE CLIPPING OF ALL SIGNALS TO PRODUCE A 3 POINT SEQUENCE (0, 1, 0)
- COMPUTE AUTOCORRELATION FUNCTION FOR PITCH WINDOW AND REPEAT
- FIND PEAK OF AUTOCORR. FUNCT. IN PITCH WINDOW
- NORMALIZE: $RAT = \frac{P}{E}$
- UNVOICED DECISION: $RAT < VUV$ OR $RAT < NGZC$
- OTHERWISE: $PITCH = \text{PITCH} + \text{PITCH_STEP}$
- AUTOCORR. FUNCT.

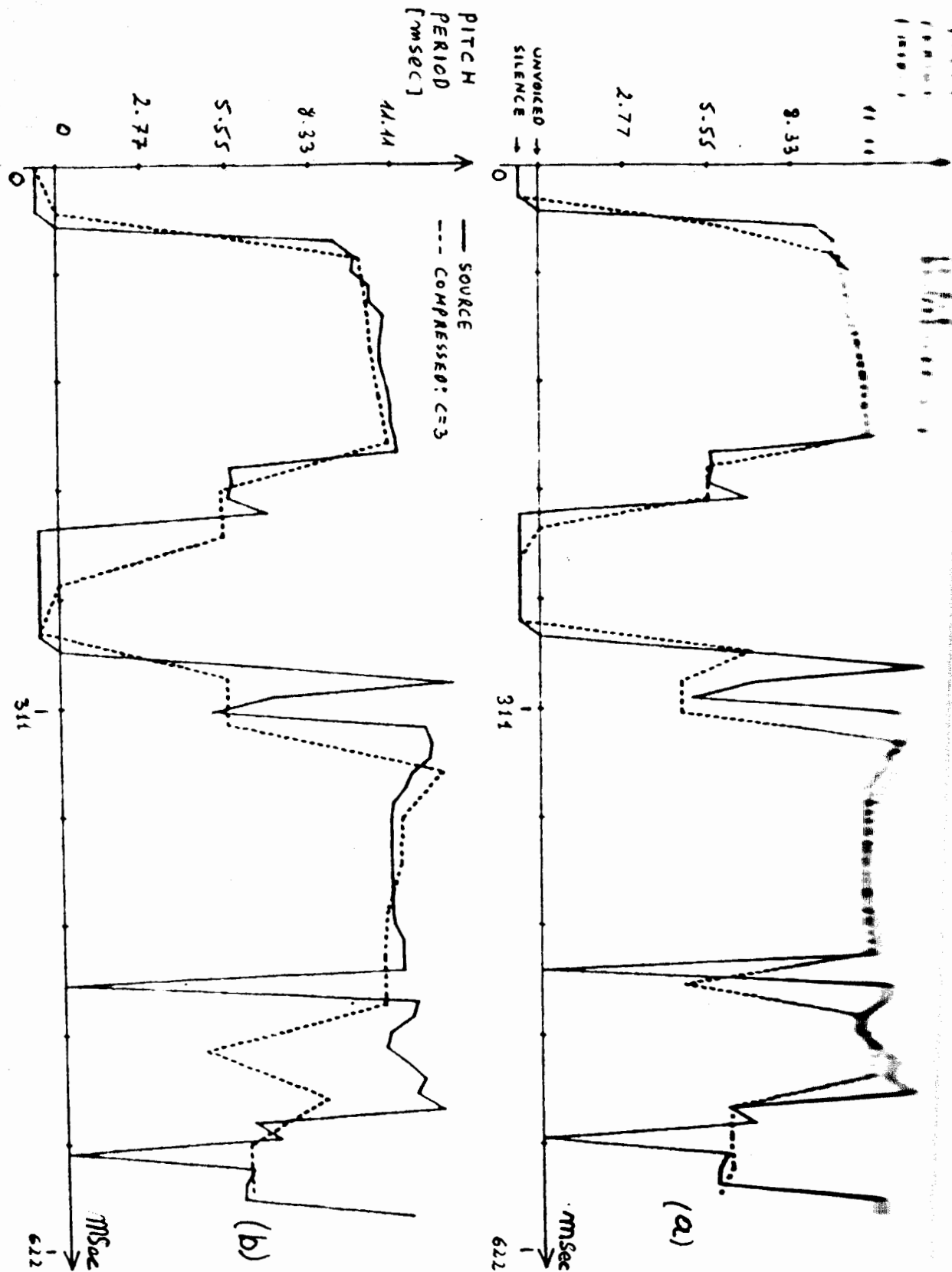
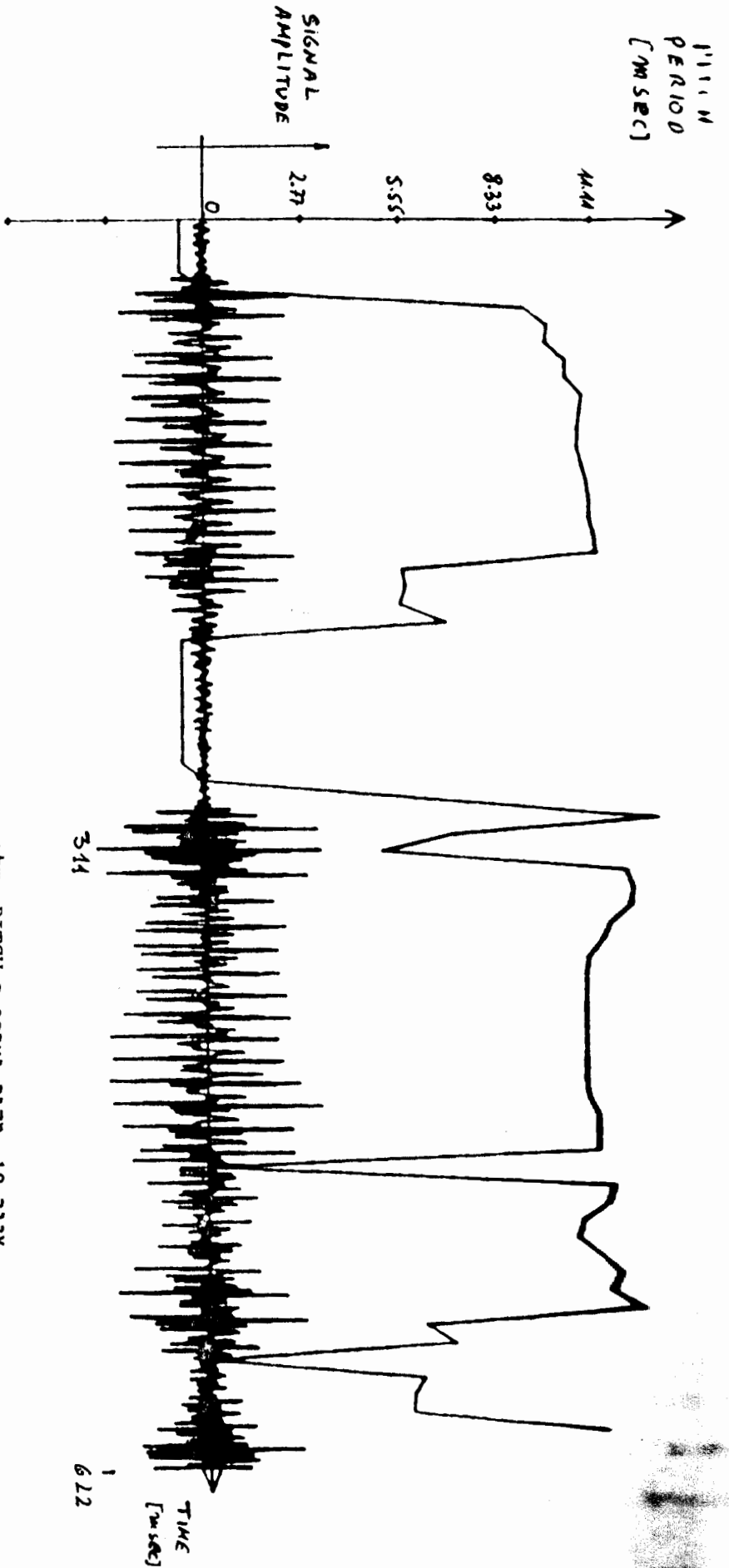
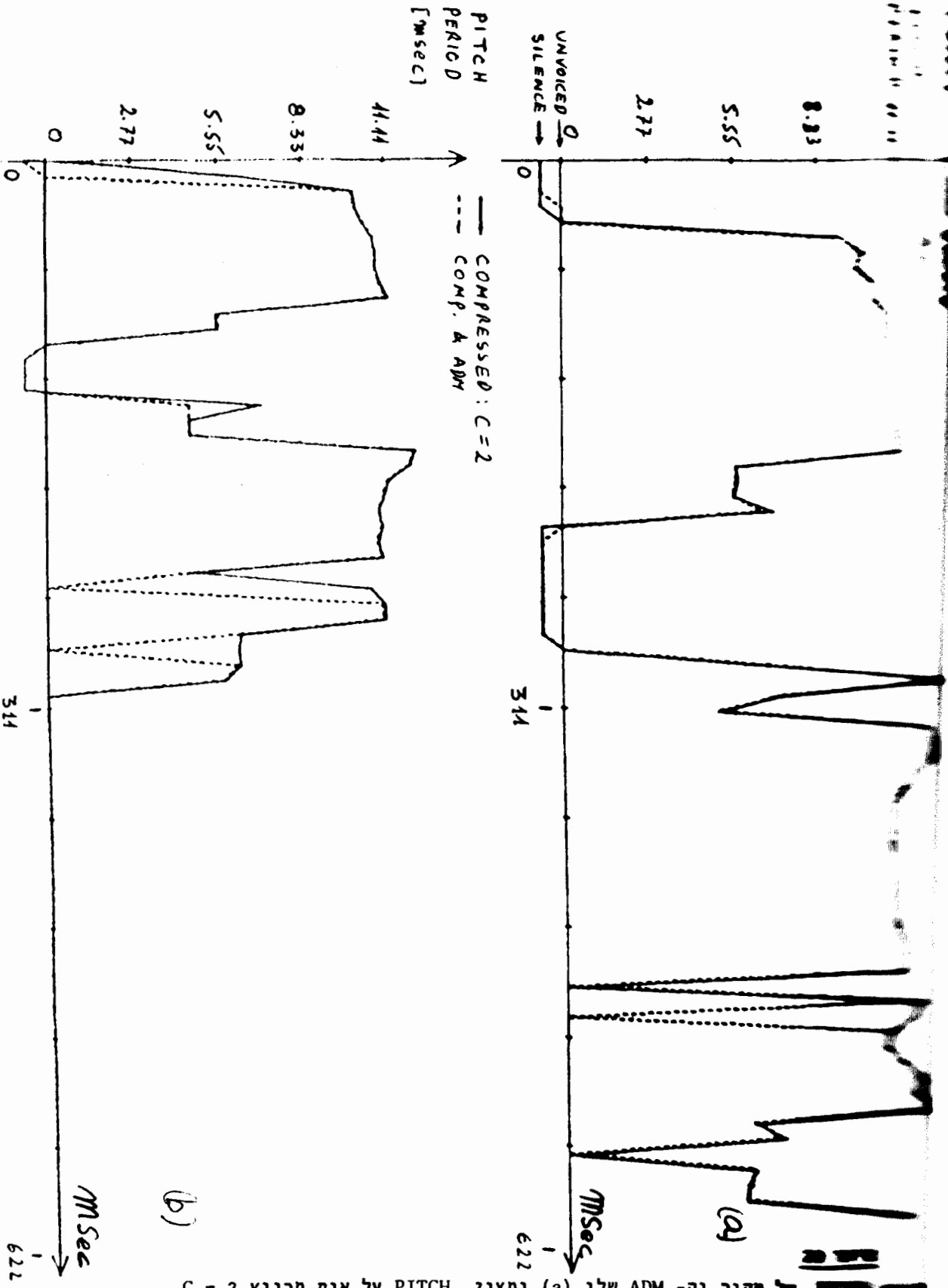


Figure 18 PITCH detection on original and divided by a factor $C = 2$ (a), and on original and divided by a factor $C = 3$ (b).



ציור 19 מקור וערכי ה-PITCH שלו בקטע שנבדק בהמשך.
 Figure 19 Source and it's PITCH on the tested segment.



ל מקור וה- ADM שלו (a) ומצוי PITCH על אור מכווץ $C = 2$ (b)

Figure 20 PITCH detection on original and it's ADM (a) and PITCH divided by a factor $C = 2$ and it's ADM (b).

ייגיסות הרבה יותר לתת-מחזור מאשר לעל-מחזור ברור מתוך תכונות האלגוריתם
 ייתחה אשר בו מספר המסננים האקוילנטיים המחלקים את רחב סרט האות הוא
 ייגיסות במחזור ה- PITCH. אם מחליטים על תת-מחזור יש פחות מסננים ויתכן
 מהרמוניה אחת של ה- PITCH נמצאת במסנן מסויים דבר שיגרום לשבירת
 הרמוניים באות המשוחזר. נובע מכאן כי עבור אות סטציונרי, החלטה על
 (במדויק) אינה צריכה לגרום לפגיעה כלל באות המשוחזר (בהנחה של
 כיוון שהאות אינו סטציונרי והמסננים אינם אידיאליים (תגובת
 והמחזור הבסיסי או הכפול אינו מדויק הרי שקיימת דגרדציה
 המשוחזר בהחלטה על מחזור כפול, אך זו קטנה בהרבה מהמקרה של החלטה
 תופעה בולטת לגבי החלטת על-מחזור היא יצירת מעין הד באזורי מעבר
 אות קולי לאיזור שאינו קולי.
 PITCH גם הוא אינו אבסולוטי.

$$|\Delta F_p|/F_p < 1/2CM$$

- 2 - פתח
- 3 - התדר היסודי
- 4 - מספר ההרמוניות אותם מבקשים לשחזר
- 5 - מקדם הכיווץ

זה $M = 10$ לכסוי תחום הפורמנט הראשון הכולל את מרבית האנרגיה של
 $C = 2.5\%$ נקבל שנדרש דיוק של 2.5% אם משדרים את ה- PITCH בערוץ.
 מיצוי PITCH במקלט על האות המכווץ נדרש דיוק נמוך יותר

$$|\Delta F_p|/F_p < 1/(2M)$$

הדרישות הנייל הן במקרה האידיאלי כלומר PITCH ה- 5% במיצוי ה- PITCH.

ההיגורני ומסננים אידיאליים ובמקרה המעשי רצוי דיוק טוב יותר. (לפי מודל
 של PITCH של האות המשוחזר אינו זהה ל-PITCH של האות המקורי וכבר דנו בבעיה

למימוש

למיצוי PITCH שבו השתמשנו עד לשלב זה (מופיע בציור 17) הינו אלגוריתם
 של חישובים וזכרון גדול. הבעיה העיקרית שלו היא שלכל דגם יש לבצע
 CLIPPING עם CLIPPING LEVEL שונה בכל פעם ועל כן עלינו לשמור את
 מספר פעמים CLIPPING מספר פעמים. הדבר היה מגדיל את זמן החישוב
 בזמן אמיתי בלפחות פי שמונה.
 מומש ב-RANDOM LOGIC לפעולה בזמן אמיתי [6] ומספר הרכיבים
 גדול פי 7 לערך ממספר הרכיבים במערכת שמומשה כאן.

האלגוריתם המתוקן מופיעה בציור 21 והתוכנות עצמה (PTCLP)

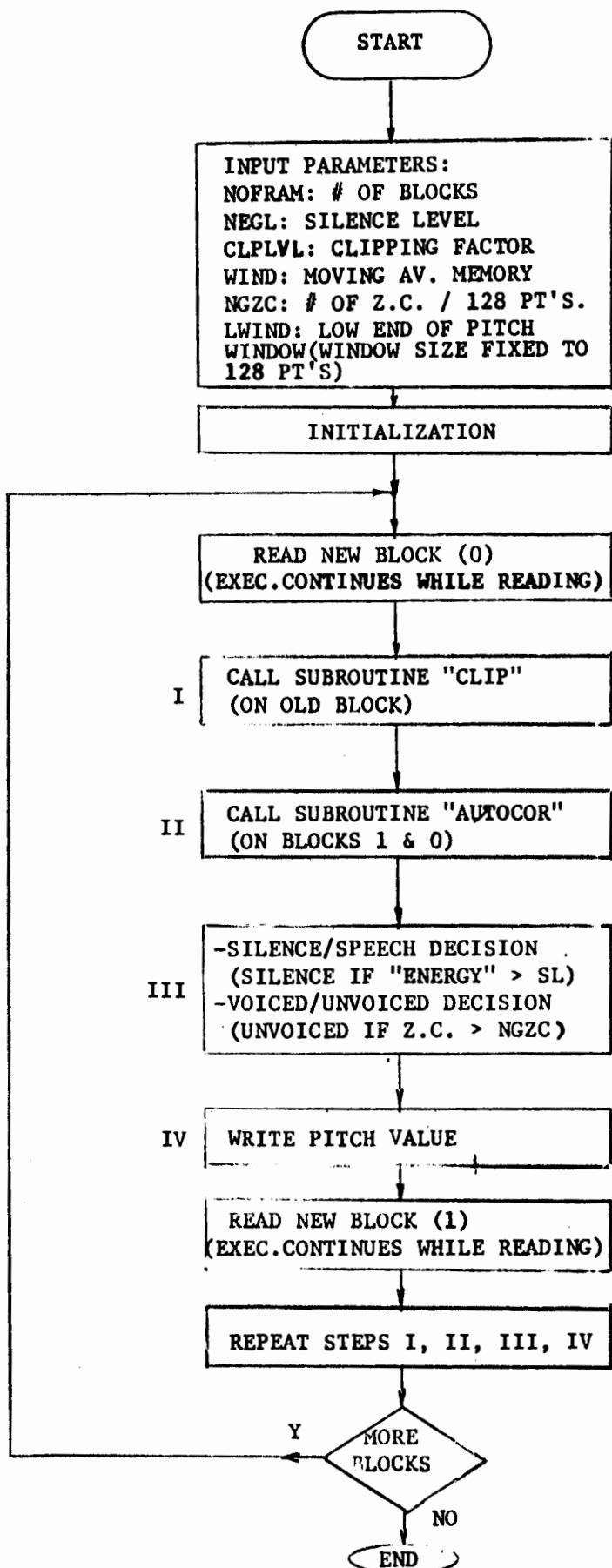
בינו לבין האלגוריתם הקודם, הוא שלכל דגם מבוצע פעם אחת CLIPPING
 מאוחסן באוטוקורלטור ואין צורך לזכור או לשמור את הדגם. בחירת
 CLIPPING נעשית בצורה דומה אולם רמה זו קיימת רק לגבי הקטע החדש שנדגם.
 השוואה בין שני סוגי האלגוריתמים עבור אות מקור ואות שעבר כיווץ
 $C = 1$.

העקיבה אחר ה-PITCH בשתי השיטות נובעים בעקר מהעובדה שהאלגוריתם
 על אות שעבר דצימציה ביחס 2:1 וקצב העדכון הוא על כן איטי יותר

האלגוריתם המוצע הדגדציה באיכות האות המשוחזר הינה קלה
 בזמן אמיתי מצדיקים את השמוש בו.

Fig 21

FLOW CHART OF PITCH EXTRACTION PROGRAMM - PTCLP



SUBROUTINE CLIP

- COMPUTE "ENERGY" OF THE BLOCK $(\sum |S(i)|^2)$
- FIND # OF ZEROES IN BLOCK
- LOW PASS FILTER BY MOVING AVERAGE OF POINTS $(L = WIND)$
- FIND PEAK IN BLOCK
- IF ENERGY < NEGL & Z.C. > NGZC SET ENERGY TO MAX LEVEL
- DETERMINE CLIPPING LEVEL $CLPLVL * (\text{SMALLER OF LAST 2 BLOCKS})$
- PERFORM CLIPPING OF ENERGY WITH 2:1 RATIO

SUBROUTINE AUTOCOR

- PERFORM CORRELATION BETWEEN LAST CURRENT BLOCK AND PREVIOUS BLOCKS
- TAPPER CORRELATION BY $R(x) = \exp(-x)$
- FIND PEAK OF CORRELATION FUNCTION IN BLOCK

1KHz להעביר את האות הנכנס לממצה ה-PITCH דרך L.P.F. ברוחב סרט של 1KHz
 מציג מציג ה-PITCH משתמשים ב-L.P.F. אנלוגי על האות בכניסה. מכיון
 לנו המקלט קולט סידרה של ביטים המשוחזרת לסדרת דגמים ספרתיים הרי
 להעבירו דרך L.P.F. סיפתי ללא D/A ולאחר מכן A/D.
 זכיה זה הוא לבצע L.P.F. ספרתי פשוט שימוש עיי המיקרו מחשב של
 ה-PITCH. המסנן שנבחר הוא מסנן מיצוע עם זכרון סופי

(57) (לאחר דצימציה)
$$\tilde{X}_n = \frac{1}{8} \sum_{i=0}^7 X_{n-i}$$
 7.2KHz של

(58) (לפני דצימציה)
$$\tilde{X}_n = \frac{1}{16} \sum_{i=0}^{15} X_{n-i}$$
 14.4KHz של

פונת לאחר הפילטר

פונת הדגום

נרוע ובעל אונות צד גבוהות (SIN(X)/X) אולם מטרת המסנן במערכת
 בחום הנמוך ומסנן זה עונה על דרישות המערכת, ומצד שני קל
 המיקרו מחשב.

סימולציות ל- (ADM) ADAPTIVE DELTA MODULATION

ה - ז - ם - ה

DM בערכת בה גודל הצעד הוא אדפטיבי. לכל דגימה נעשית השוואה עם
 וסימן התוצאה +1 או -1 נשלח בערוץ. האות המשוחזר נוצר ע"י
 מוכפל בסימן לאות המשוחזר הקודם

$$(59) \quad e_n = \text{SIGN}(X_n - \bar{X}_n)$$

סיבית משודרת -
 אות משוחזר -
 אות דגום -
 גודל הצעד ה- n -

$$(60) \quad \bar{X}_{n+1} = \bar{X}_n + e_n * \Delta_n$$

דגיל Δ_n קבוע ולכן יש צורך להשתמש בקצב דגימה גבוה על מנת לשחזר
 גודל ADM-ב. Δ_n תלוי באופי האות ומסיבה זו אפשר להקטין את קצב
 בין הסוגים של ה-ADM הוא באופן עידכון Δ_n .
 נבדקו והושוו בסימולציות 3 סוגי ADM.
 [7] [JAYANT] ששימש בעיקר כנקודת יחוס ובו נקבע
 סיביות הקודמות.

בו נקבע גודל הצעד הן ע"י הסיביות הקודמות והן ע"י אנרגיית

[2] (CONTINUOUSLY VARIABLE SLOPE D.M.) CVSD

ע"י הביטים הקודמים, אולם בצורה עקיפה גם אנרגיית האות משפיעה.
 להשוואה בין שלושת ה-ADM-ים וכן מומשו השנים האחרונים על
 טובה יותר על טיבם ע"י שמיעת קטעים ארוכים.

שלושת סוגי ה-ADM-ים ולאחר מכן את תוצאות הסימולציות של ההשוואה

ADM-ב - הדיברת ההיברידי גודל הצעד נקבע בשיטה המשלבת
 INSTANTANEOUS COMPANDING עם SYLLABIC COMPANDING. גודל הצעד נקבע הן ע"י
 אחרונות בדומה ל-1, והן ע"י אנרגיית האות בחלון של מספר מילי שניות
 הן באופי המקומי של הדיבור והן באנרגיית ההברה שנאמרת
 שימוש בשני גורמים אלו במקביל ניתן לשפר את איכות ה- ADM.
 נקבע בצורה הבאה

$$(65) \quad \Delta_n = F(Y_n * G_n)$$

$$\Delta_n < \Delta_{max}$$

$$(66) \quad G_n = M_n * G_{n-1}$$

$$G_{min} < G_n < G_{max}$$

$$(67) \quad M_n = Fe(e_n, e_{n-1}, e_{n-2})$$

$$(68) \quad Y_n = \mu Y_{n-1} + |\bar{X}_n|$$

- סדרת הכניסה

- סדרה משוחזרת

- סיבית משודרת

- גודל הצעד

- עוצמה הממוצעת של האות המשוחזר

- גודל המקביל ל- Δ_n באלגוריתם 1. המשתנה לפי האופי המקומי של הדיבור.

מבצענו עם אלגוריתם זה נמצא $\tau = 6msec$ כגודל מתאים. באמצעות G_n

מתבצעת האדפטציה הרגעית ואילו ע"י שקלול Y_n עם $f(Y_n)$ מבוצע

עוצמת האות המשוחזר בקטע זמן ארוך יותר (לפי קבוע הזמן τ)

למימוש $f(Y_n)$

ב-כ"י בגלל קצב הדגימה הנמוך $L = 3$.

הסיבה לכך היא כי באופן מקומי נקבע הצעד לפי אופי האות באזור הקרוב אולם הוא משתנה בצורה יחסית לאלגוריתמים הקודמים. (חיבור ולא כפל) גודל הצעד שסביבו נעשים חישובים המקומיים נקבע לפי אנרגיית האות בקטע הקודם קרי - ככל שהאנרגיה היתה גבוהה יותר מספר הפעמים בהם $e_n = e_{n-1} = e_{n-2}$ היה גדול יותר ולכן גודל הצעד גדול יותר.

הצעד יחסית יותר לרעש מהשיטות הקודמות משום שרעש אקראי ישנה רק במעט את הצעד בעוד שבשיטות 1 ו-2 הוא עלול להכפיל את הצעד.

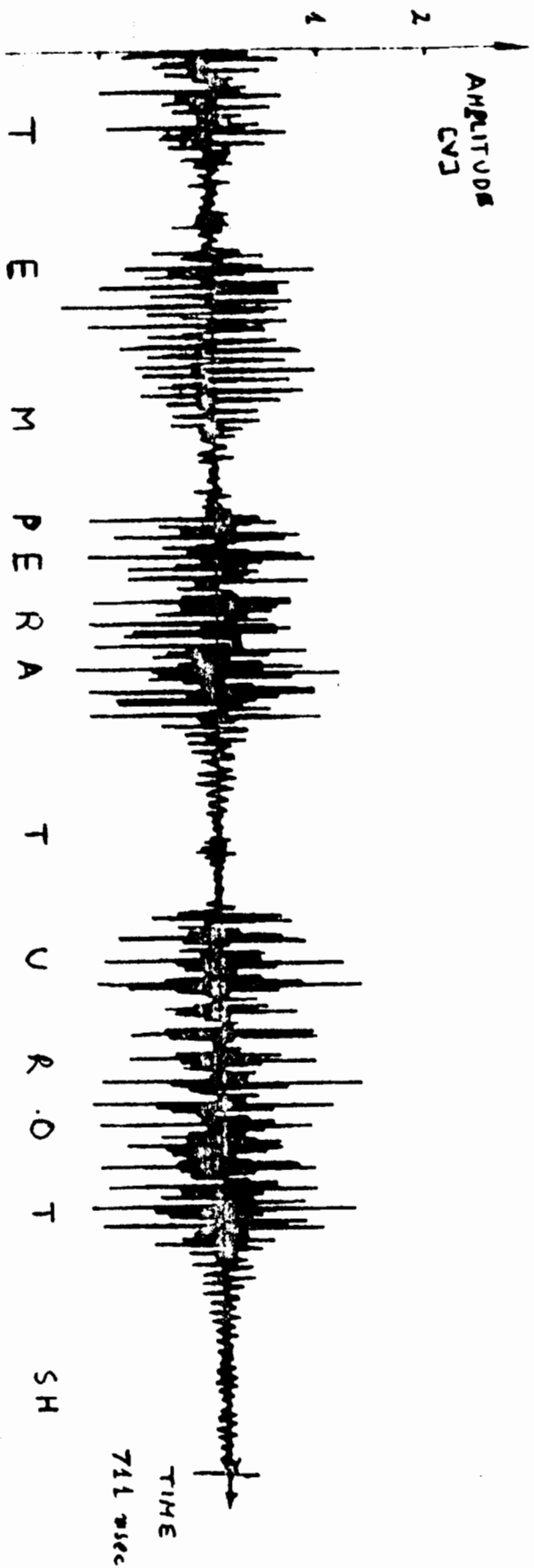
תוצאות סימולציות

בדיקות שנעשו על שלושת האלגוריתמים שהוצגו הוברר שכדי לקבל איכות סבירה של חיבור בין לרדת מקצב של 14KBPS לערך. במטרה להשוות את בצועי האלגוריתמים הוצגו סימולציות על קטע דיבור של 10,000 דגימות בקצב דגימה של 14.4KHz. עם דיבור כולל קטעי דיבור קוליים ולא קוליים והוא מוצג בציר 23. תוצאות נבדק יחס האות לרעש באות המשוחזר בתלות בעצמת האות הכניסה. התוצאות בציר 24 ו-25. בציר 25 היציאה היא לאחר מסנן בציר 26. בטבלה הבאה מושוים שלושת האלגוריתמים מבחינת התחום הדינמי (S/N) וכן גם ה-S/N המקסימלי עבור כל אחת מהיחידות. ההשוואה נעשת גם עם מסנן וגם עבור אות ללא סינון.

טבלה 1 השוואה בין הסוגים השונים של ADM.

Table 1 Comparison between different types of ADM.

PROPERTY TYPE of ADM	DYNAMIC RANGE [dB]		Max S/N [dB]	
	NON-FILTERED (S/N > 8dB)	FILTERED (S/N > 8dB)	NON FILTERED	FILTERED
ADM	13.1	20	10	
HADM	23.3	28.9	10.25	
CVSD	22.2	32.5	10	



המלה ששימשה להשוואה בין הסוגים השונים של ה-ADM. ציור 23

Figure 23 Plot of the word that was used for the comparison between the various types of ADM.

Figure 1 shows the magnitude of signal to noise ratio for different types of ADP, non filtered output.

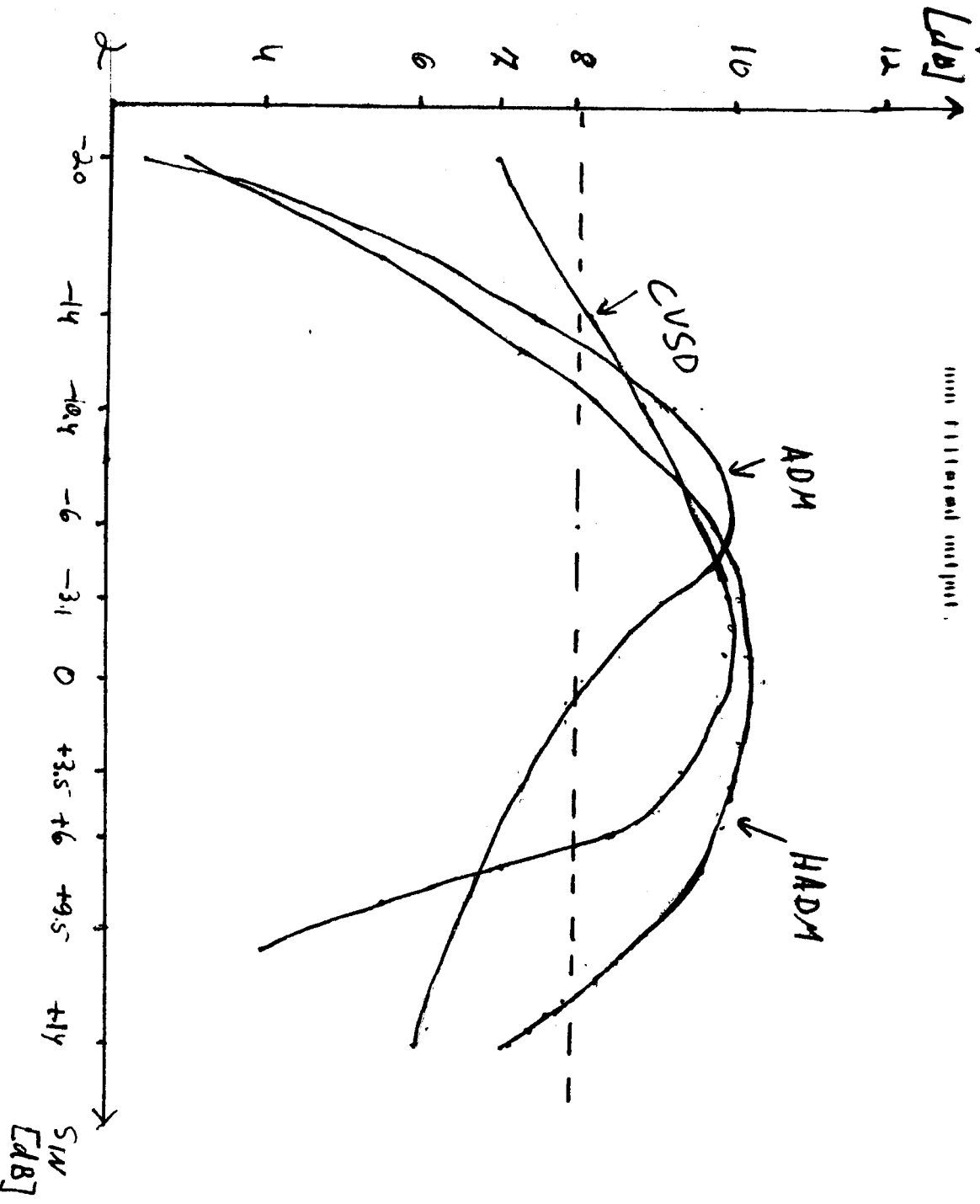
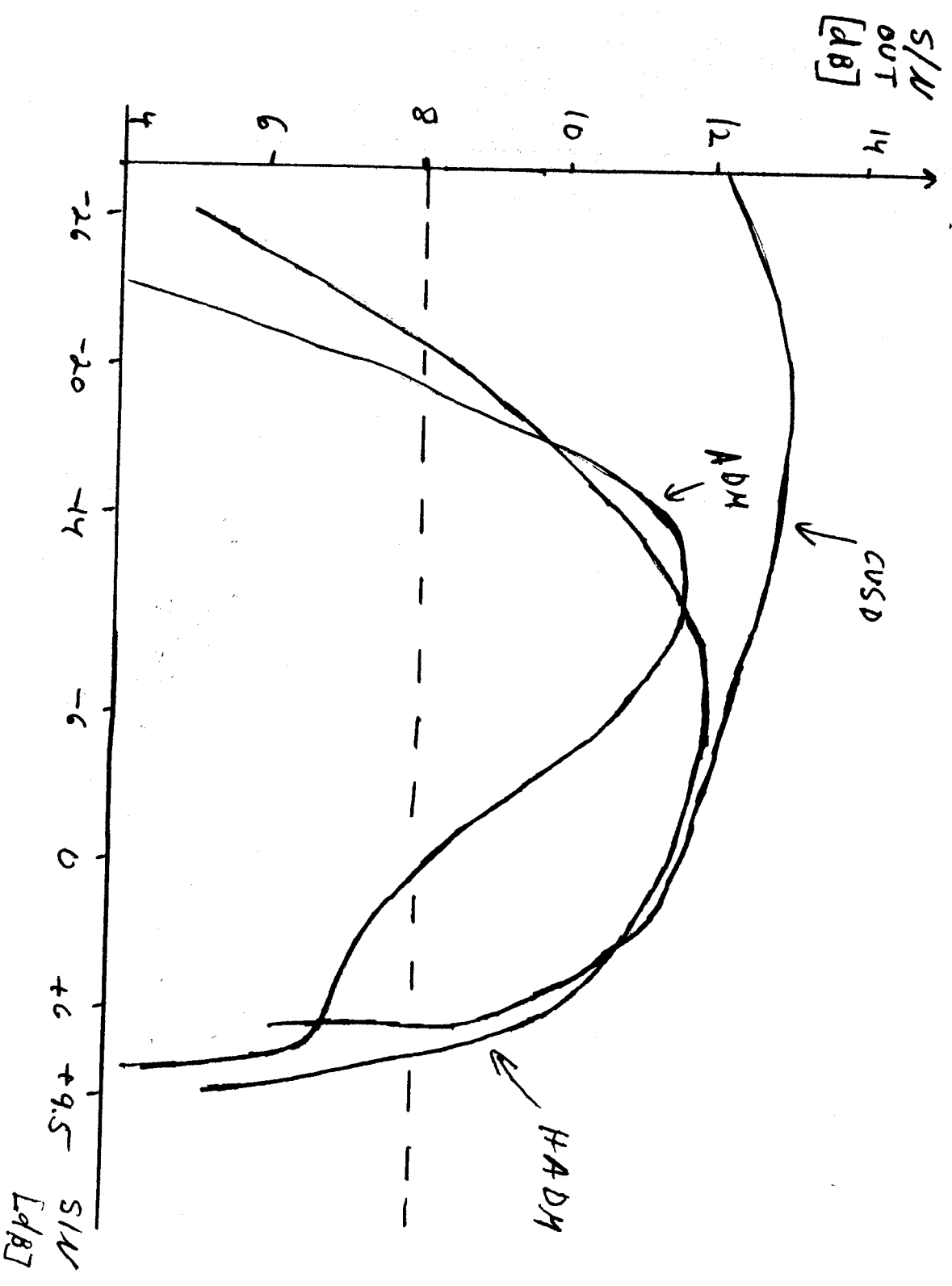
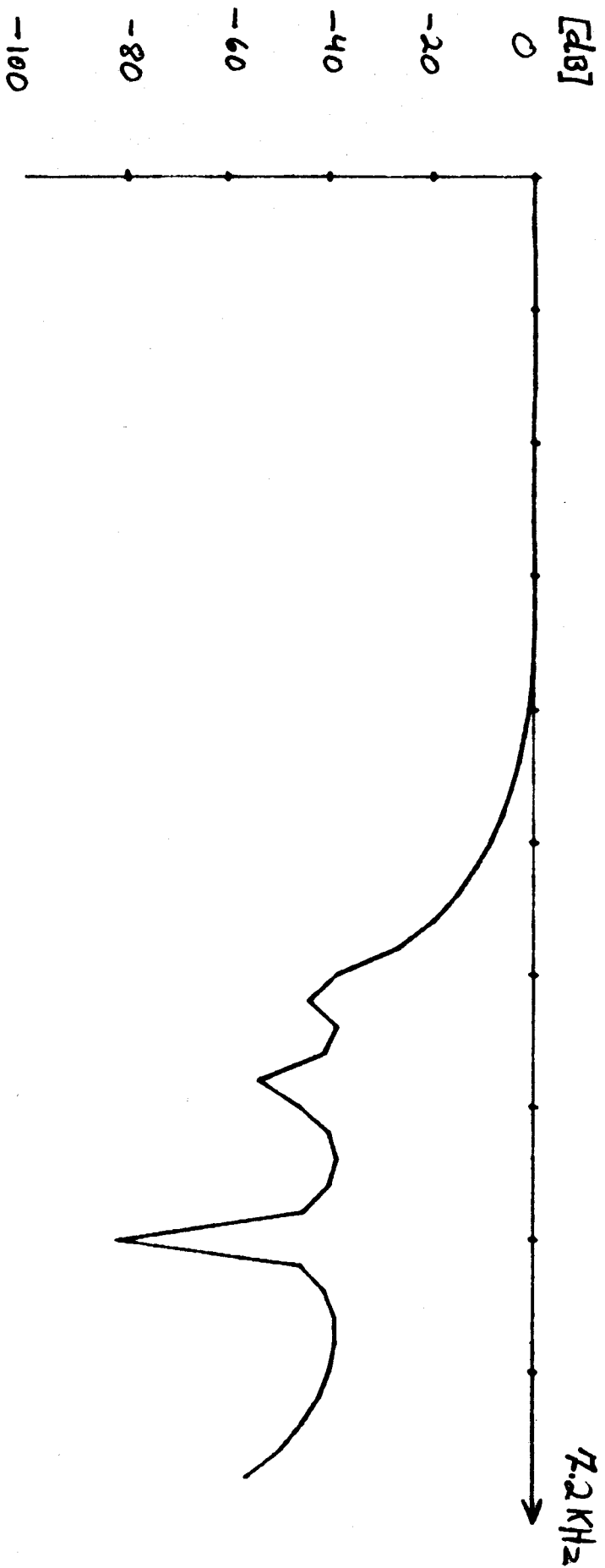


Figure 25 Graphs of signal to noise V. input signal for different types of ADM, filtered output.





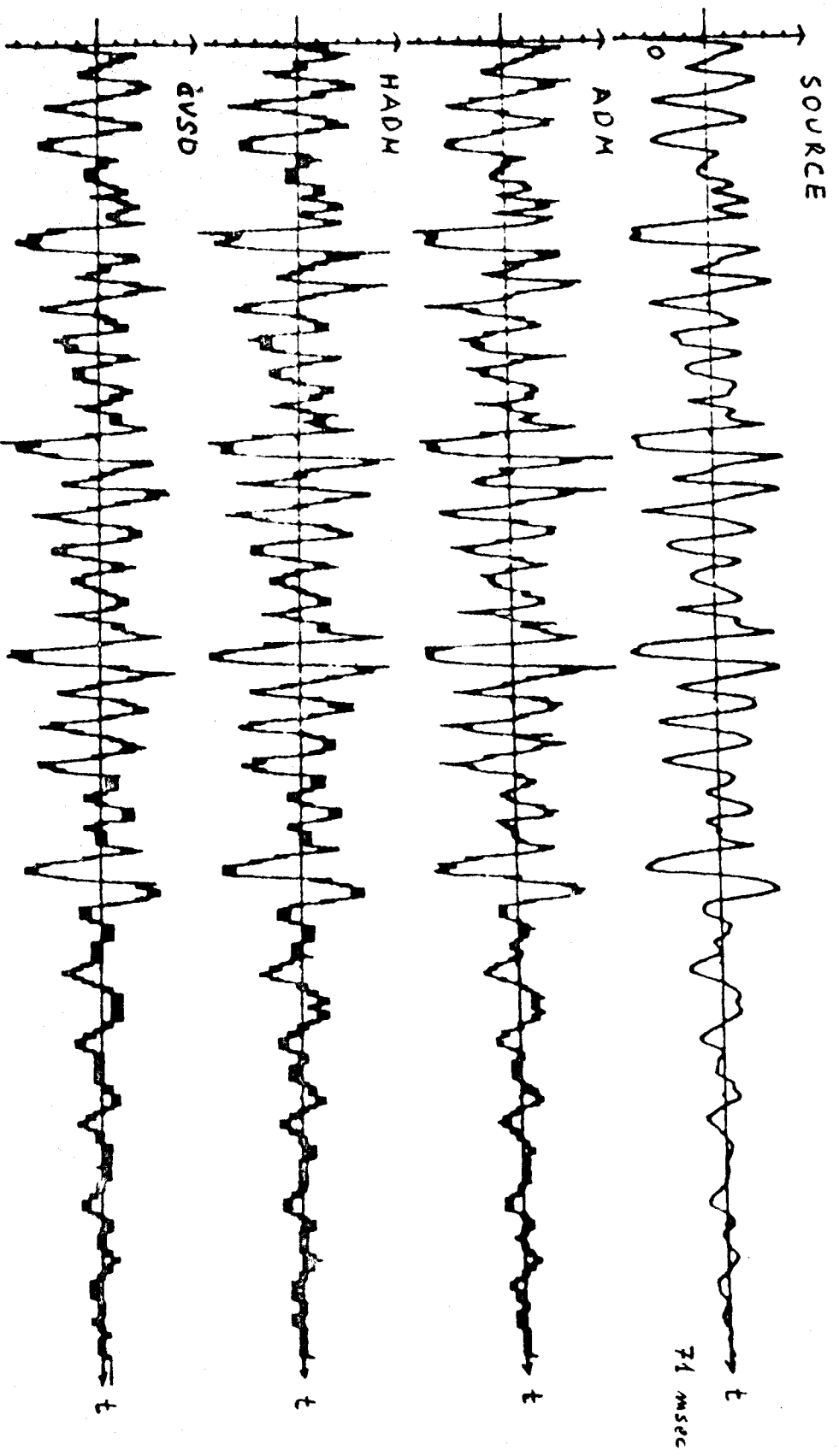
ציור 26 מסנן מעביר נמוכים שאותות היציאה מה-ADM הועברו דרכו.

Figure 26 The low pass filter that was used for the ADM's output.

מהציוורים ומהטבלה רואים בברור כי יתרונם העיקרי של ה-ADM ההיברידי ושל ה-CVSD על ה-ADM הרגיל (על פי JAYANT) היא בהרחבה ניכרת של התחום הדינמי. סנון אות היציאה נותן תוצאה טובה יותר הן מבחינת התחום הדינמי והן מבחינת יחס האות לרעש. שיפוע זה גדול יותר ב-ADM ההיברידי וב-CVSD מאשר ב-ADM הרגיל (2dB לעומת 1dB בממוצע) נזאת עקב תכולה רבה יותר של רכיב בחצי תדר הדגימה (ב-7.2KHz) כפי שניתן לראות מצורת הגל בציור 27. הציור מראה קטע אות כניסה ותוצאות השיחזור בכל אחת משלושת האלגוריתמים שנבדקו. בציור 28 מתואר הקטע לאחר סינון.

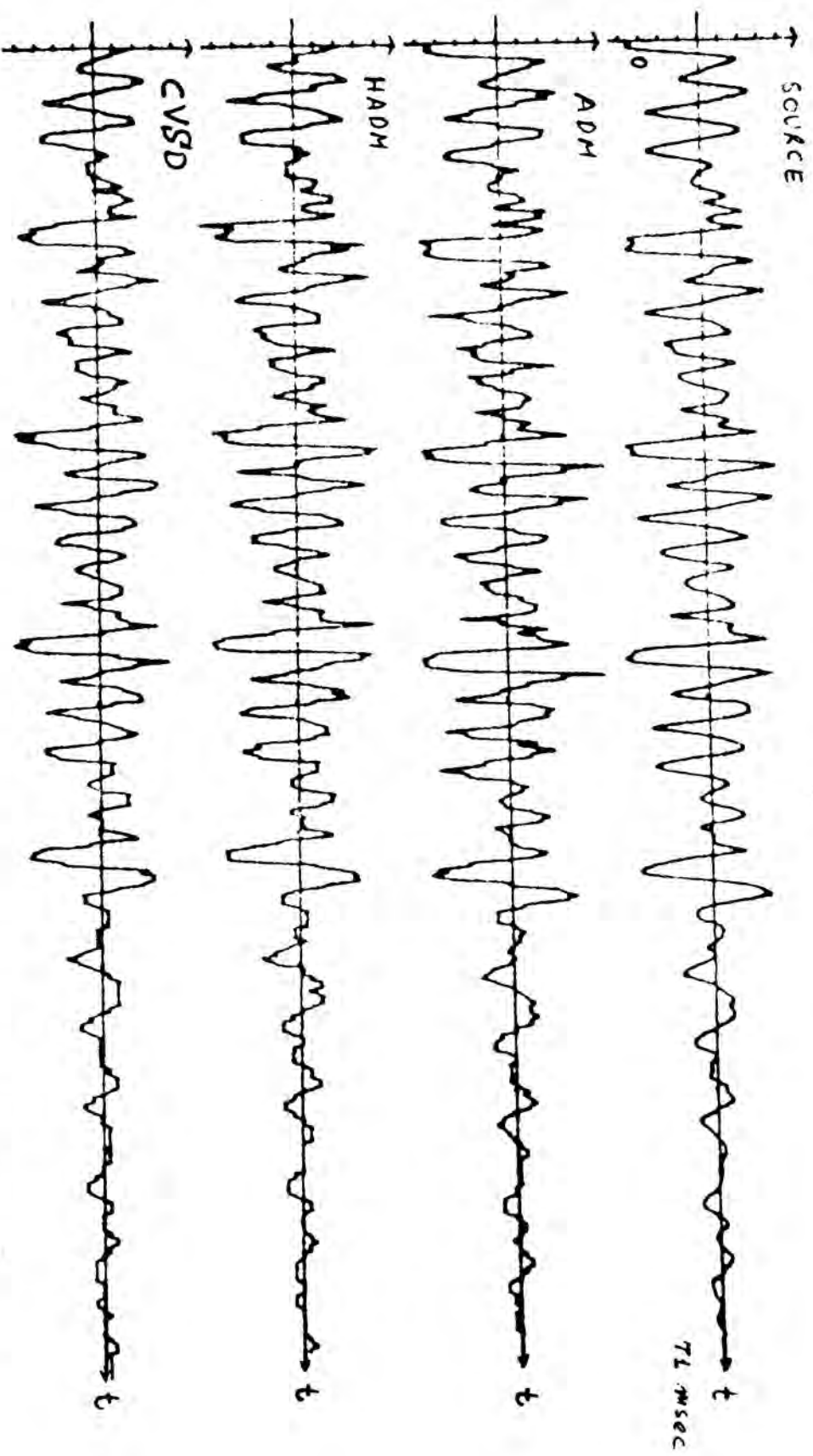
ממכלול התוצאות רואים כי אלגוריתם ה-CVSD ואלגוריתם ההיברידי עולים בהרבה על ה-ADM לפי JAYANT ה-CVSD עולה במעט על ה-ADM ההיברידי. מתוצאות הסימולציה היה עלינו להחליט באיזה אלגוריתם לבחור למערכת הסופית. יש לציין כי במקביל לסימולציות מומשו 2 האלגוריתמים וניתן היה להעריך את טיבם ע"י שמיעת קטעים ארוכים. גם משמיעת קטעים אלו נראה כי שני האלגוריתמים ה-ADM ההיברידי וה-CVSD קרובים זה לזה בטיבם, אם כי ה-CVSD עולה במעט על האלגוריתם ההיברידי. ההחלטה על ה-CVSD נפלה בעיקר משום שאותו אפשר לממש בתוכנה בלבד בניגוד לאלגוריתם ההיברידי שדורש תוספת טבלות (המאוחסנות ב-ROMS) המכילות לוח כפל לדלתא

קיים בשוק רכיב בודד המבצע CVSD. אולם רכיב זה בנוי ל-16K קצב דגימה ואילו אנו זקוקים לקצב של 7.2KHz שהוא קצב הדגימה לאחר כיווץ. הפרמטרים שונים ולכן כדי לקבל תוצאות טובות הועדף להשתמש במימוש (הדורש רק תוספת תוכנה) ולא להודיף רכיב חיצוני לביצוע CVSD. בנוסף המערכת שלנו סיפרתית וכדי להתחבר לרכיב CVSD אנלוגי היה צורך ב-D/A במשדר ו-A/D במקלט.



ציור 27 מקור ואות משוחזר ללא פילטר עבור סוגים שונים של .ADM

Figure 27 Source and nonfiltered reconstructed signal for different types of ADM.



מקור ואות משוחזר עם מסנן מעביר נמוכים עבור סוגים שונים של ADM. ציור 28

Figure 28 Source and filtered reconstructed signal for different types of ADM.

פרק 5 : סימולציות למערכת כולה

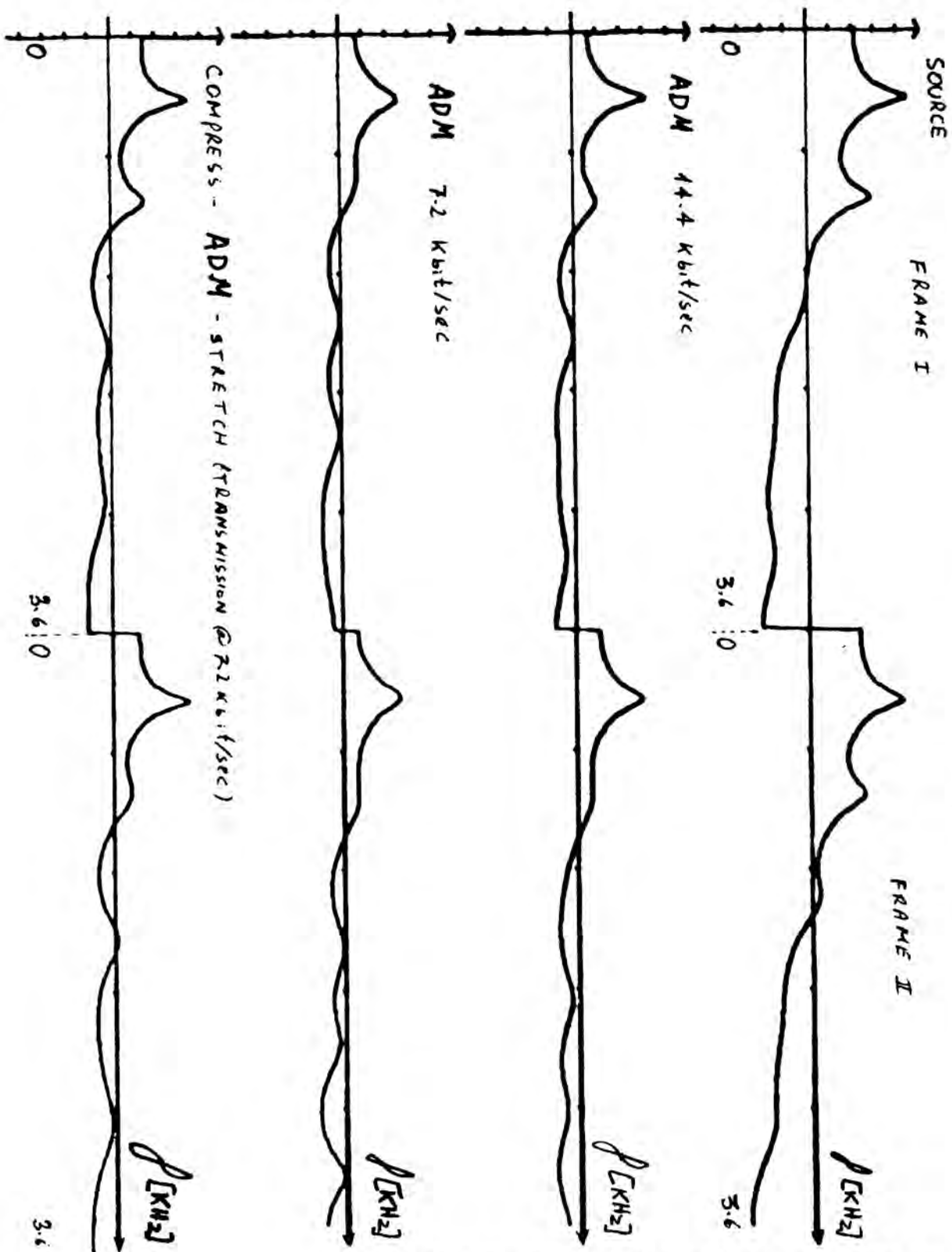
בשלב זה של הסימולציות בוצעה סימולציה של כל המערכת בשלבים

1. זהוי PITCH
2. ביצוע כיווץ
3. ביצוע ADM וקבלת האות המשוחזר כפי שהיה נוצר במקלט
4. זהוי PITCH של האות המשוחזר
5. ביצוע מתיחה של האות המשוחזר.

בעזרת חמשת שלבים אלו בוצעה סימולציה מלאה של המערכת.

בציור 29 רואים את שלבי הסימולציה על קטע קולי של כ- 70msec ובציור 30 רואים את תגובת התדר של קטע זה בכל אחד משלבי הסימולציה. להשוואה ניתן ADM בקצב 14.4K מהתוצאות רואים כי הפורמנט הראשון מועבר במלואו ואילו הפורמנט השני מונחת בעיקר בגלל ה-ADM. יש לציין כי הפורמנט השני מונחת יותר ב-ADM בקצב 14.4K מאשר בכל התהליך של כיווץ, ADM בקצב 7.2KHz, ולאחר מכן מתיחה. בנוסף נעשו סימולציות על קטעים ארוכים של כ-20 שניות לזוג קרייני חדשות גבר ואשה. התוצאות שהתקבלו מראות על עיוותים קלים בלבד ועל מובנות טובה של הדיבור. אין גם קושי בזהוי הדובר. מתוצאות הבדיקות של סעיף זה וסעיפים קודמים הגענו למסקנה כי המערכת הפשוטה ביותר למימוש אשר תתן תוצאות טובות היא זו אשר בה אין שידור PITCH והיא משלבת מערכת ADM מסוג CVSD. האילוצים שהוזכרו לעיל מתייבים את הספציפיקציות הבאות.

1. מקדם כוויץ $C = 2$ בגלל העובדה שאין משדרים PITCH בערוץ.
 2. קצב תמסורת 7200 סיביות לשניה (עם אפשרות לרדת ל-4800 סיביות לשניה אם ימצא ADM שיאפשר זאת).
- בפרק הבא תתואר המערכת שנבנתה, השיקולים לתיכנונה ותוצאות הפעלתה בזמן אמיתי על אותות דיבור.



ציור 30 תגובה לחדר של האותות הנחונים בציור 29.

Figure 30

Frequency response of the signals in Fig. 29.

פרק 6 : מימוש המערכת

6.1 מ ב ו א

המערכת שמומשה כוללת שני מיקרו מחשבים מסוג INTEL 8085. המערכת מחולקת לשני חלקים עיקריים כשכל חלק נמצא בשליטת מיקרו מחשב אחד.

מיקרו מחשב ראשון שולט על מערכת זיהוי ה-PITCH והוא כולל אוטוקולטור. מיקרו מחשב שני מבצע את מערכת ה-COMPRESS/STRETCH ומערכת ה-ADM והוא כולל מכפל חיצוני.

בפרק זה יתואר גלאי ה-PITCH ובפרק הבא מערכת הכיוון. הקשר בין שתי היחידות הוא

עיי קריאה במקביל מאותו A2D ועיי העברת אינפורמציה ה-PITCH (הכוללת

PITCH PERIOD ו-1/PITCH PERIOD). ממערכת גלאי ה-PITCH למערכת כיוון

מתיחה/ADM.

המערכות יכולות לעבוד גם כמערכות עצמאיות בלי תלות אחת בשניה כלומר ניתן להחליף למשל את גלאי ה-PITCH בגלאי אחר ללא צורך בפעולות מיוחדות.

בציור 31 מתוארות יחידות המקלט/משדר ואילו בציור 32 מתואר הקשר בין שתי יחידות.

נתאר את המערכת לפי חלקיה, גלאי ה-PITCH, מערכת STRETCH/COMPRESS ומערכת ה-ADM.

בכל מערכת כזו הדיון יתחלק לשני חלקים, מבנה ותאור החומרה ומבנה ותאור התוכנה.

במסגרת הדיון יתוארו גם מערכות שנבנו לבדיקה אך לא הוכנסו למימוש הסופי (ADM)

היברידי ואוטוקורלטור של סיבית אחת). יש לציין שבגלל המימוש עם מיקרו מחשבים

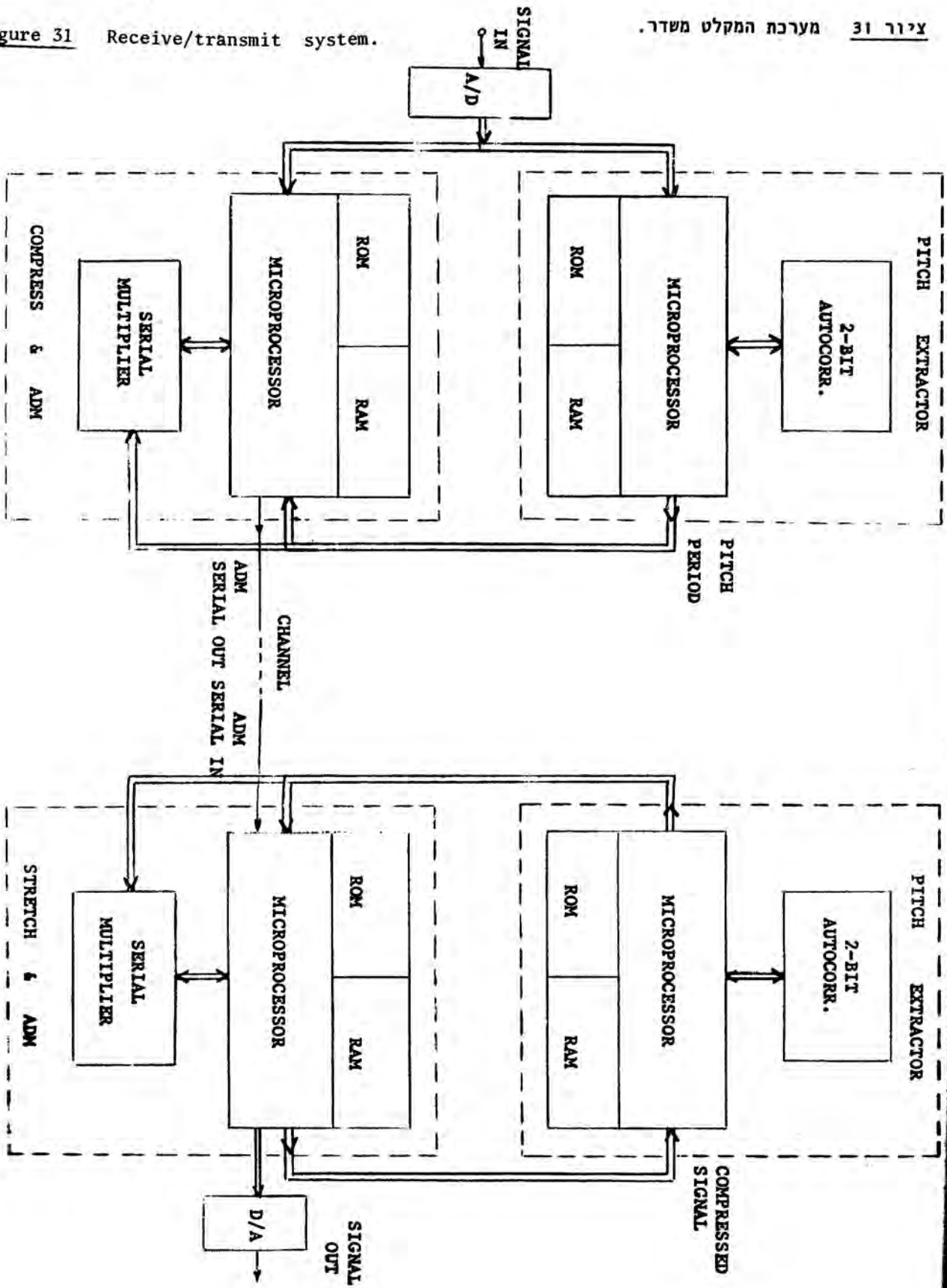
קיימת אפשרות לתקן, לעדכן ואף לשנות את פרמטרי המערכת והפונקציות שלה ללא כל

קושי. למשל כיוון עם $C = 3$ יידרוש רק שינוי בתוכנת המיקרו מחשב שמבצע כיוון

ללא כל שינוי אחר. סכמות מפורטות של המערכת כולה וכן תוכניות האסמבלר של שני

המיקרו מחשבים מצורפות בנספח.

Figure 31 Receive/transmit system.



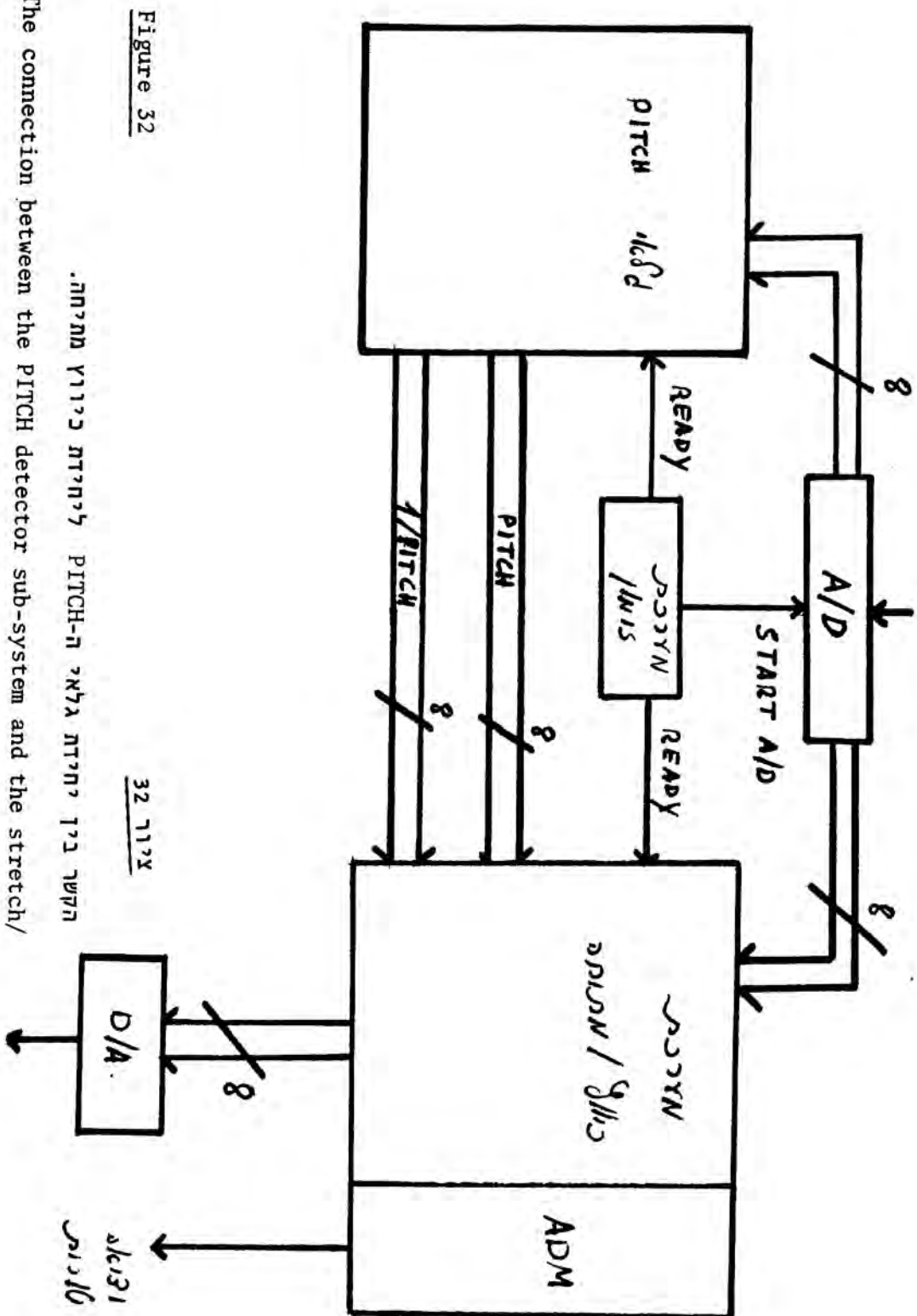


Figure 32

הקשר בין יחידת גלאי PITCH-ה ליחידת כיוון מתחיה.

32 רי"צ

The connection between the PITCH detector sub-system and the stretch/compress sub-system.

6 חמרה

מערכת גלאי ה-PITCH צריכה לבצע את הפונקציות הבאות:

1. דגימת האות
2. ביצוע סינון על האות הדגום
3. חישוב מקסימום בכל קטע לשם מציאת CLIPPING LEVEL (קטע = 100 דגמים)
4. חישוב מספר חציות אפס בקטע
5. חישוב האנרגיה בקטע
6. החלטה על ה- CLIPPING LEVEL לפי שני הקטעים האחרונים
7. ביצוע CLIPPING על הקטע
8. אוטוקורלציה על תוצאות ה- CLIPPING
9. ביצוע TAPERING על תוצאות פונקציית האוטוקורלציה
10. מציאת המקסימום של פונקציית האוטוקורלציה
11. החלטה על אופי הקטע לפי חישוב האנרגיה, חציות האפס ומספר ההתאמות במקסימום פונקציית האוטוקורלציה
12. החלטה על ערך ה- PITCH (PITCH ממוצע במקרה של אות לא קולי)
13. חישוב $1/PITCH$ לשם הוצאה למערכת הכיווץ.

מכיוון שקצב הדגימה במערכת זו הוא 7200 דגמים לשניה הרי עומדים לרשותנו כ-140 מיקרו שניות לביצוע מחזור פעולות כאלו. פעולות 13, 12, 11, 6 הן פעולות המבוצעות פעם אחת בכל קטע ועליהן נדון בהמשך. יתר הפעולות מתבצעות עבור כל דגימה. בשלב ראשון היה צורך להחליט מה ימומש בחמרה ומה בתוכנה. ברור שרצוי להגדיל עד כמה שאפשר את חלקה של התוכנה וע"י כך לחסוך רכיבים ולהגדיל את האפשרות לשנות בקלות את המערכת. אפשר לראות מיד כי השלב שדורש את מירב הפעולות הוא שלב האוטוקורלציה שכן הוא מחייב 200 פעולות אריתמטיות עבור כל דגימה וכן מאה פניות לזכרון. זמן פקודה

ממוצע של ה-8085 הוא כ- 2.5μ ולכן היה ברור כי פעולה זאת יש צורך לבצע בחומרה
 ($140 > 300 \times 2.5$). לאחר בדיקה הוחלט לבצע את יתר הפעולות בתוכנה.
 המערכת צריכה לעבוד בזמן אמיתי ולכן יש צורך לבצע את כל הפעולות בין דגימה
 לדגימה. ניצול זמן C.P.U. מגיע לכדי 98% בקצב של 7.2KHz וזהו ניצול מקסימלי
 של המערכת.

מערכת החמרה מתחלקת למספר חלקים עיקריים.

1. מיקרו מחשב
2. מערכת זימון דגימה
3. מערכת האוטוקורלטור.

6.2.1 מערכת המיקרו מחשב

מערכת המיקרו מחשב כוללת מיקרו מחשב INTEL 8085 זכרון EPROM לאיחסון התוכנית
 בן $2k \times 8 \text{ bit}$. זכרון RAM בגודל 8×256 , הכולל גם 22 קווי I/O, לצורך שמירת
 100 דגימות לביצוע CLIPPING וכן לתאי עזר לחישוב. בנוסף כוללת המערכת חוצצים
 על קווי הכתובת והמידע. בציר 33 מתוארת דיאגרמת בלוקים של המערכת. יציאות
 המערכת הן 2 קבוצות כל 8 קווים של PITCH ו-1/PITCH, כניסות המערכת הן
 8 קווים מה-A/D וקו READY ממערכת הזימון. המערכת עובדת ב-MODE שנקרא
 MEMORY MAPED I/O שבו התקני הקלט/פלט ממופים באזור הזכרון. לוגיקת ה-CHIP SELECT
 ממפה את מרחב הזכרון ומאפשרת שליטה על התקני ה-I/O של המחשב.

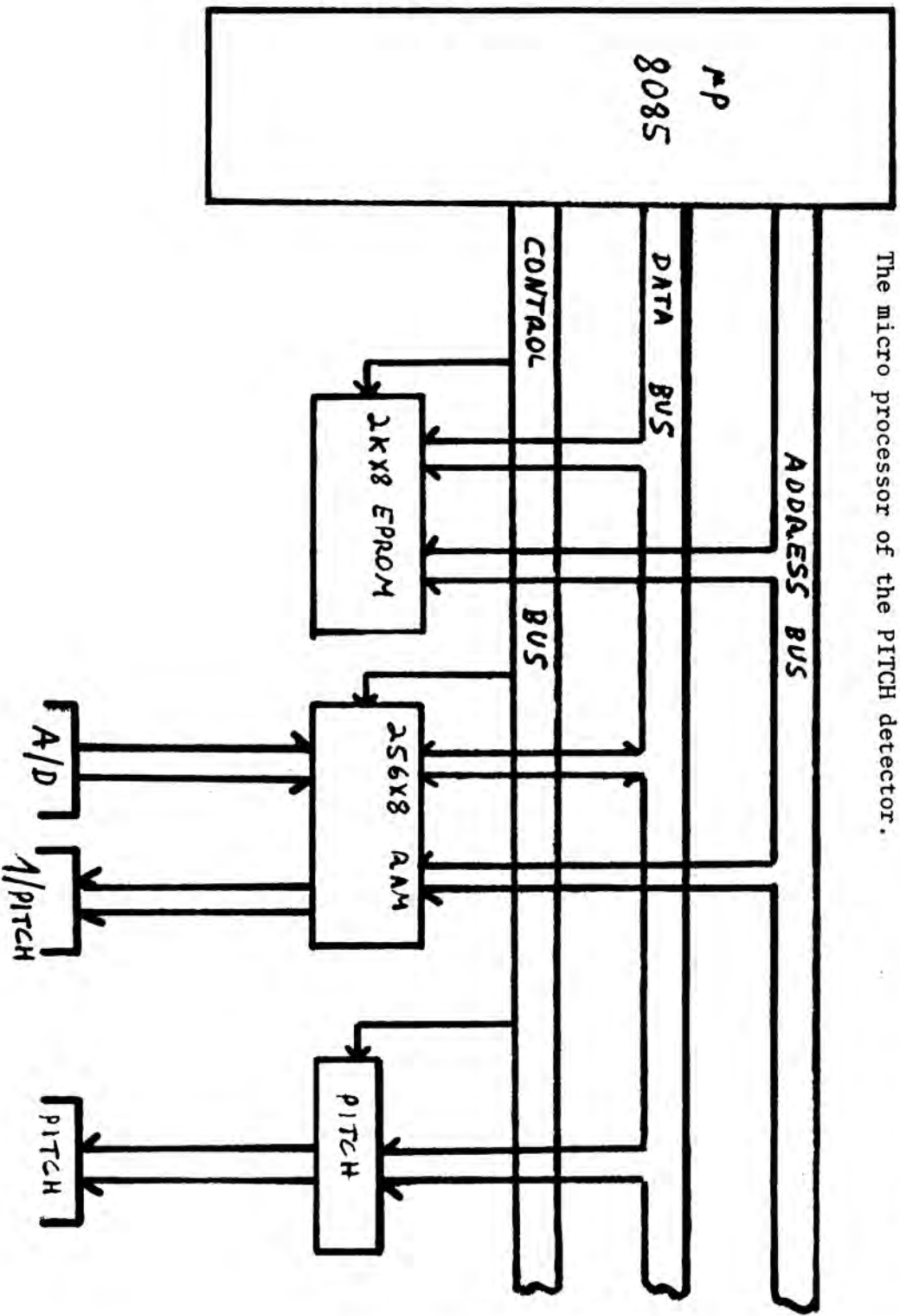
6.2.2 מערכת הזימון

על המערכת להפעיל את ה-A/D ולקרוא ממנו בקצב קבוע. מכיוון שמשך זמן ביצוע
 התוכנית בין דגימה לדגימה אינו קבוע והוא תלוי בדגימה הקודמת, יש צורך לסנכרן
 בין התוכנה לבין ה-A/D. קיימות מספר שיטות לבצע זאת בתוכנה אולם מכיוון
 שדרוש דיוק דגימה וזמן ה-C.P.U. נוצל כמעט עד תום, בוצע הסינכרון בחומרה.

Figure 33

יחידת המיקרו מחשב בגלאי ה-PITCH 33 יור 33

The micro processor of the PITCH detector.

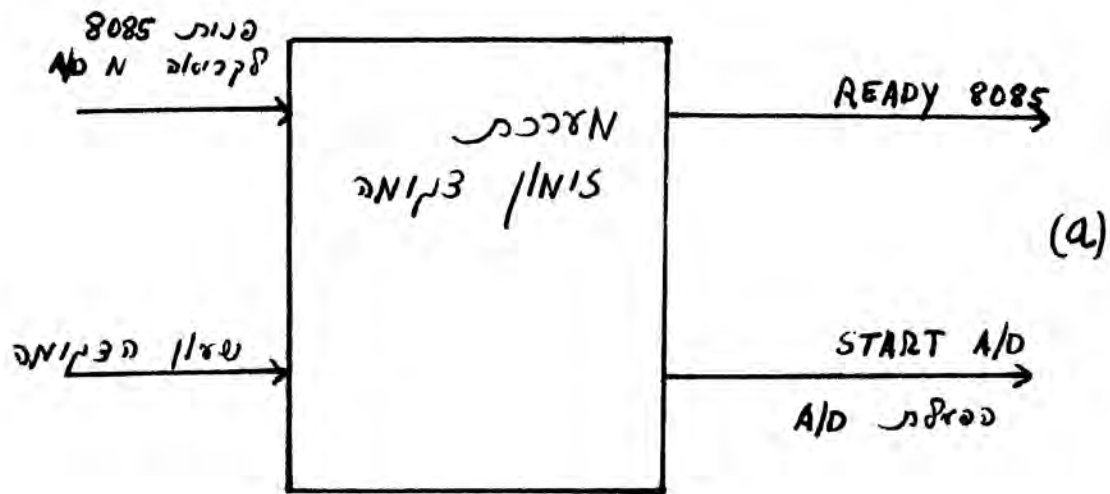


מערכת הזימון מסתנכרת על הפקודה LDA A/D (טען מילה ממתמר אנלוגי לספרתי) בצורה הבאה: ברגע שמזוהה CHIP SELECT ל-A/D מורד קו ה-READY למיקרו מחשב והוא נכנס להמתנה. עם עליית השעון החיצוני המכוון על קצב הדגימה משוחרר ה-READY, המחשב קורא מילה מה-A/D, ה-A/φ מקבל פולס להתמרה נוספת ומחזור נוסף מתחיל. מבנה ודיאגרמת זמנים של מערכת הסינכרון מופיעים בציור 34. הפעלת ה-A/D וכן קריאת דגימה קודמת מתבצעים עם עליית השעון החיצוני, מציור רואים כי משך הזמן שבו יורד READY שונה בכל מחזור והוא תלוי באורך התוכנית בקטע זה.

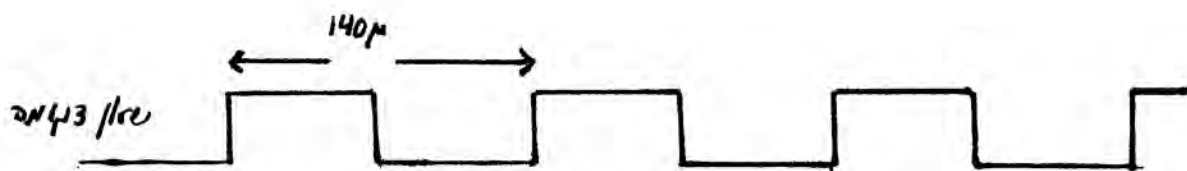
6.2.3 מערכת האוטוקורלטור

מערכת האוטוקורלטור משולבת כהתקן I/O של המיקרו מחשב. היא תוכננה להפעלה ע"י פקודה אחת בלבד של המיקרו מחשב עבור כל מחזור של אוטוקורלציה. בציור 35 מתואר המבנה של האוטוקורלטור ובציור 36 מתוארת מערכת השעונים והבקרה של האוטוקורלטור. המערכת מורכבת מ-5 רגיסטרי הזזה באורך של 100 bit וברוחב מילה של 2 ביט [במקרה של אפנון דלתא מספיק רוחב של ביט אחד בלבד]. המערכת מחולקת לשני חלקים עיקריים החלק המבצע אוטוקורלציה והחלק הנטען בתוצאת ה-CLIPPING. עבור כל דגימה מתבצעת טעינה של האוטוקורלטור בתוצאת CLIPPING של דגימה ומתבצע מחזור של אוטוקורלציה.

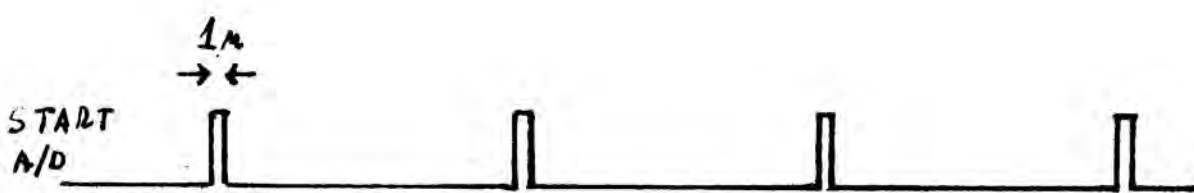
החלוקה הנ"ל היא גמישה ומאפשרת החלפת תפקודים המתבצעת ע"י שליטה על קווי הבקרה של הבוררים $2 + 1$ בכניסות וביציאות רגיסטרי ההזזה. לשם הבהרת התהליך נתאר עתה את שני המצבים של המערכת ואופן ביצוע הקורלציה. כאמור אורך כל רגיסטר הזזה 100 דגימות וזהו רוחב החלון של האוטוקורלציה. נגדיר חלון זה כ- F_n עבור הקבוצה ה-n-ית של 100 דגימות. ציור 37 אי מתאר מצב 1 של האוטוקורלטור בהשמטת הבוררים.



(a)



(b)



ציור 34 מערכת הזימון - גלאי ה- PITCH (a) ודיאגרמת הזמנים של ה- (b).

The timing system of the PITCH detector (a) and it's timing diagrams (b).

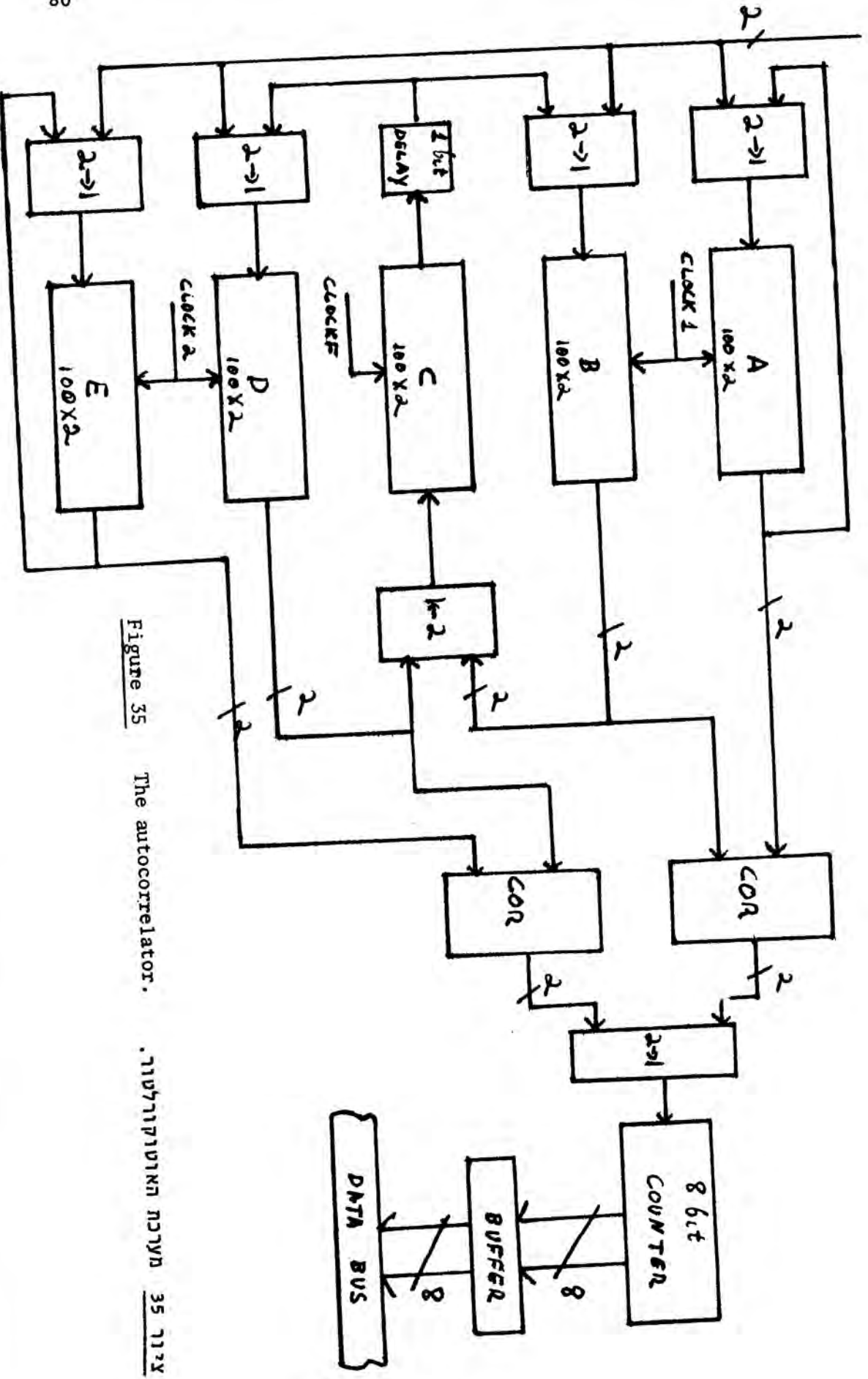


Figure 35 The autocorrelator.

מערכת האוטוקורלטר. 35 ת"צ

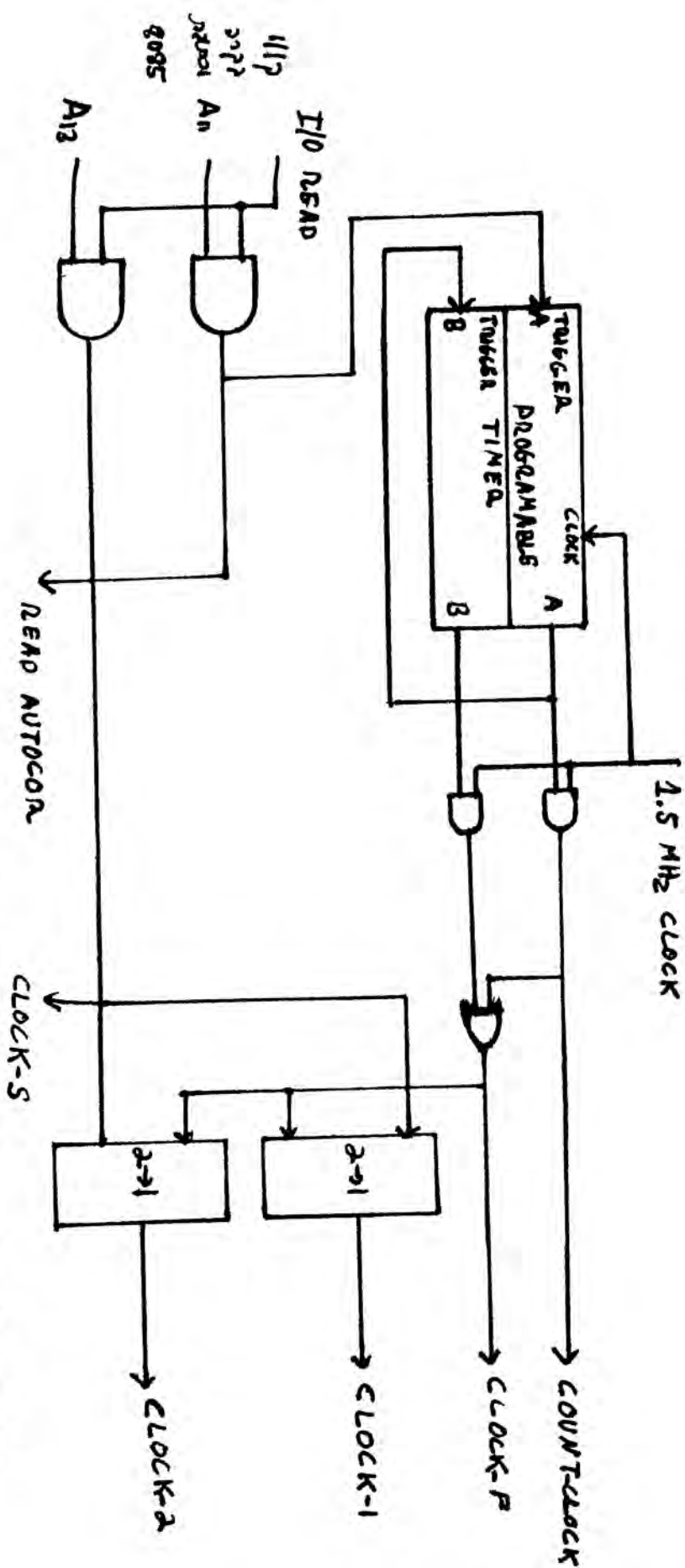


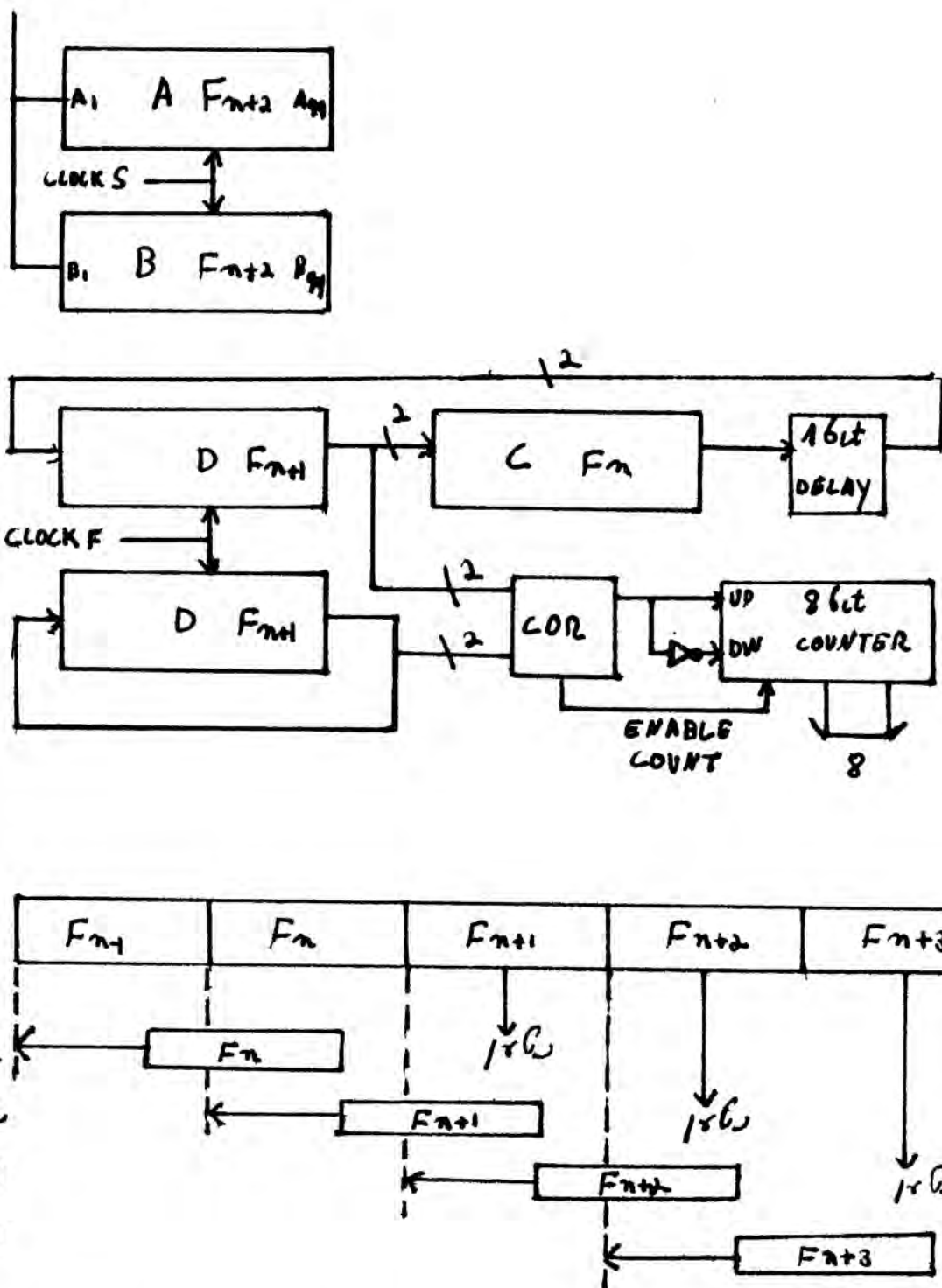
Figure 36 The control logic and timing of the autocorrelator:

מערכות זימון ובקרה של האוטוקורלרטור.

הקטע ה-n טעון ברגיסטרים E ו-D והקטע ה-n-1 טעון ברגיסטר C. עבור כל פעולת אוטוקורלציה מקבלים רכבת של 200 פולסי שיעון ב-CLOCKF. נעשית קורלציה בין הרגיסטרים A ו-B שמכילים שניהם את F_n ולכן נקבל במונה את תוצאת האוטוקורלציה של $R(0)$. יש לציין כי הרגיסטרים מכילים אינפורמציה של (+, 0, -) בלבד ולכן תוצאת ה-COR יכולה להיות (+, 0, -) בלבד גם היא. מספיק לכן מונה ואין צורך במערכת סיכום+אגירה לצורך חישוב האוטוקורלציה. בנוסף יש לציין כי רק ב-100 הפולסים הראשונים של CLOCKF נעשית הקורלציה ולשם כך קיים COUNT CLOCK שחוטט את המונה ב-100 הפולסים האחרונים.

נתאר עתה את מצב הרגיסטרים לאחר חישוב $R(0)$. רגיסטר B אורכו 100 ביט ויציאתו מחוברת לכניסתו ולכן לאחר 200 פולסי שיעון תוכנו לא השתנה. רגיסטרים A, C ואלמנט ההשהיה מחוברים בטור ובצורה ציקלית ואורכם יחד 201 ביט. לאחר 200 פולסי שיעון תוכן תא ההשהיה יגיע ל-C100 תוכן C100 יגיע ל-C99 וכן הלאה. תוכן C1 יועבר ל-A100, A100 ל-A99 וכן הלאה עד ל-A1 שתוכנו יגיע לתא של אלמנט ההשהיה. מה שמתבצע הוא הזזה של F_n, F_{n-1} אחורה יחסית ל- F_n הנמצא ברגיסטר B, דבר זה זהה להזזה של F_n הנמצא ברגיסטר B קדימה יחסית ל- F_{n-1}, F_n הנמצאים ברגיסטרים A, C. בצירור (37) מתואר המצב היחסי של ה-DATA באוטוקורלטור לאחר כל הפעלה ואנו רואים כי המערכת מבצעת בכל מחזור חישוב $R(i)$ ומכינה את ה-DATA ברגיסטרים לחישוב $R(i+1)$ בצורה אוטומטית. במקביל לכל הפעלה של האוטוקורלטור טוענים לרגיסטרים D, E אינפורמציה של תוצאת ביצוע CLIPPING על דגימות הקטע F_{n+1} . נסתכל על מצב המערכת לאחר ביצוע 100 הפעלות. רגיסטרים D, E טוענים בקטע F_{n+1} ברגיסטר B נשאר קטע F_n לעומת זאת לרגיסטר A הועבר קטע F_{n-1} ולרגיסטר C הועבר הקטע F_n . (ההעברה אינה מושלמת וקיימת הזזה של ביט אחד. יש שתי אפשרויות לפתור את הבעיה. אחת לבצע הפעלה נוספת של האוטוקורלטור דבר שאינו אפשרי מבחינת עבודה בזמן אמיתי והשניה בה השתמשנו היא לתת רק 199 פולסי שיעון בהפעלה ה-100. הדבר גורם להזזה כפולה ומקבלים ב-C את תוכן F_n כולו).

CLIP DATA



ציור 37 (א) תאור סכמטי של האוטוקורלטר ללא בוררים במצב ראשון שלו. (ב) תאור אופן ביצוע האוטוקורלציה לאורך מספר מסגרות.

a) Schematic description of the autocorrelator without selectors in it's first state.

b) Description of the autocorrelation of some frames.

במצב זה מבצעים בעזרת הסלקטורים החלפת תפקידים ומקבלים את המצב המתואר בציר 38 א'.

מתבצעת אוטוקורלציה של F_{n+1} על פני F_n, F_{n+1} בעוד רגיסטרים A, B נטענים
 ב- F_{n+2} . התהליך הנ"ל חוזר על עצמו בצורה רציפה שאינה פוגעת בפעולה בזמן אמיתי
 ומלבד מיתוג הבוררים המתבצע ע"י פקודה אחת אינו דורש כל פעולה בין קטע לקטע.
 בציר 38 ב' ניתן תאור סכמתי של פעולת האוטוקורלטור במשך מספר מחזורי מסגרת
 (מחזור מסגרת = 100 דגמים) כל קטע מוזז על שרשר של הקטע עם הקטע הקודם (200
 דגמים) וע"י כך מקבלים אוטוקורלציה ללא ניחות (TAPPERING) וללא ציקליות.
 תיארונו עד עתה את העבודה במסגרות של האוטוקורלטור ונעבור עתה לתאור של הפעלה
 בודדת, כיצד היא מתבצעת ע"י המיקרו מחשב.

בציר מתוארת מערכת הזימון והבקרה של האוטוקורלטור.

הפעלת האוטוקורלטור לפעם הבאה נעשית במקביל לקראת תוצאה הקודמת ע"י הפקודה

8800 LDA (LOAD ACCUMULATOR) . טעינת הקורלטור ב- CLIPPING DATA

מתבצעת ע"י הפקודה LDA A00X כאשר A_0, A_1 מכילים את אינפורמציה ה- CLIP

CLIPPING DATA = -1 - LDA A001

CLIPPING DATA = 0 - LDA A000

CLIPPING DATA = +1 - LDA A002

אפשר גם לבצע את שתי הפעולות יחד, כלומר ע"י פקודה אחת לקרוא תוצאה קודמת של
 ה- A.C., להפעיל אותו לפעם הבאה ולטעון אליו את CLIPPING DATA המתאים. הפקודות

יראו כך

CLIPPING DATA = -1 - LDA A800

CLIPPING DATA = 0 - LDA A801

CLIPPING DATA = +1 - LDA A802

מגוון צורות ההפעלה כל האוטוקורלטור הוא רב ובתיאור התוכנה יוסבר אופן וזמן השימוש
 בכל שלב ושלב.

בציר 39 ניתנה סכמה של יחידת הקורלציה. הקורלציה נעשית בין C'_0, C'_1 לבין

C''_0, C''_1

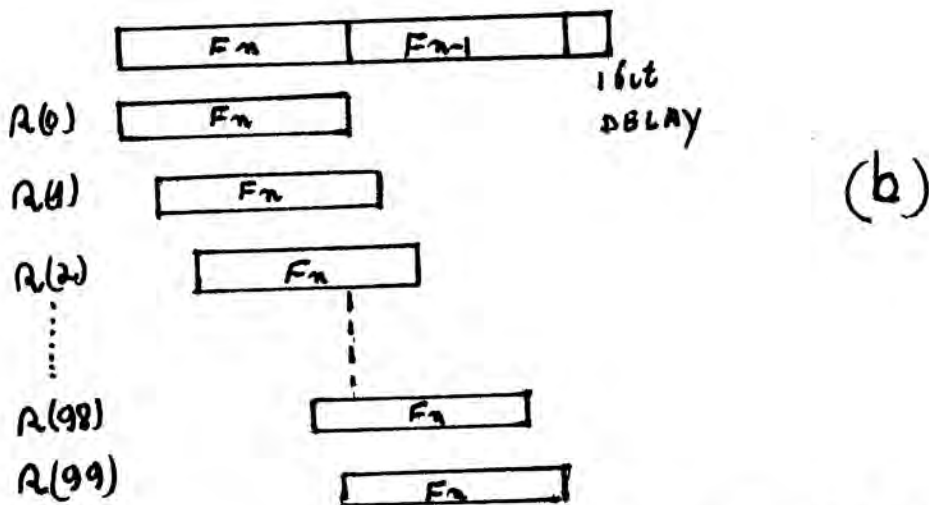
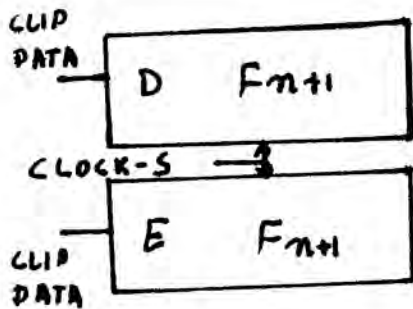
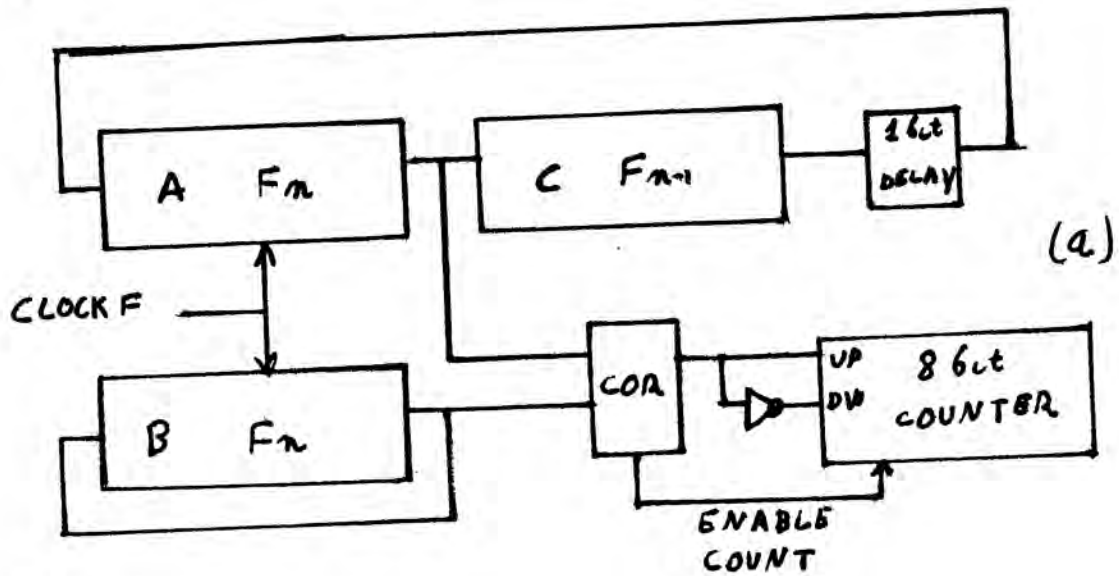
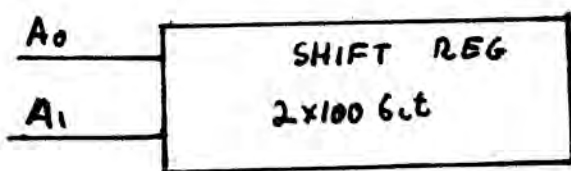
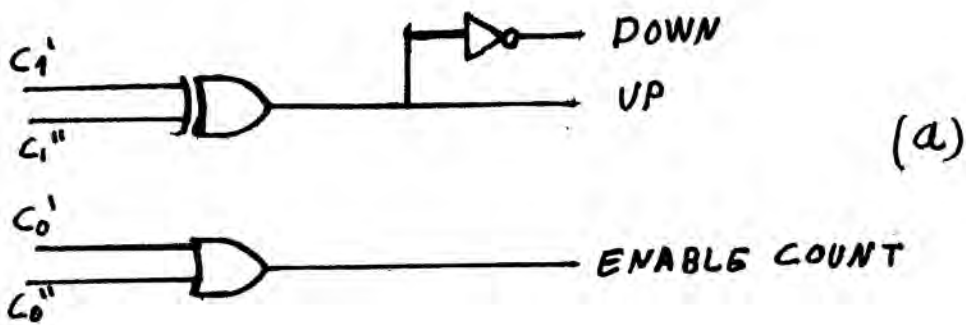


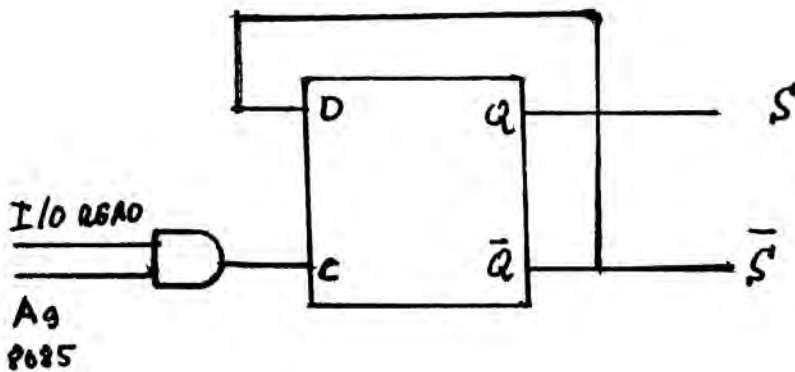
Figure 38 (א) תאור סכמטי של האוטוקורלטר ללא בוורים במצבו השני. (ב) תאור אופן ביצוע האוטוקורלציה במסגרת אחת.

(a) Schematic description of the autocorrelator without selectors in it's second state.

(b) Description of the autocorrelation in one frame.



(b) קלוי בטלגריף
8085 - 8085



(c) קלוי דקנכ
8085 - 8085

ציון 39

(א) מעגל הקולטור, (ב) כניסת - CLIPPING DATA, (ג) בקרת הבוררים.

Figure 39 (a) The correlator circuit, (b) The CLIPPING DATA inputs, (c) The selectors control.

$$C_0, C_1 = \begin{array}{ll} 00 & -1 \\ 10, 11 & 0 \\ 01 & +1 \end{array}$$

במידה ואחד מה- $C_0 = 1$ הרי אחת מהכניסות לאוטוקורלטור היא אפס ולכן המכפלה היא אפס. אם שני ה- $C_0 = 0$ מתקבלת תוצאת \overline{XOR} בין C_1 ל- C_1'' אם הם שווים נקבל '0' ואם הם שונים '1'. מהקורלטור נקבל 3 קווים למונה שבו נאגרת תוצאת הקורלציה, קו UP להגדלת ערך המונה ב-1 קו DOWN להקטנת ערך המונה ב-1. הקו השלישי הוא ENABLE למונה ובמידה ותוצאת הקורלציה היא אפס הוא אינו מאפשר שינוי ערך המונה ללא תלות במצב קווי DOWN, UP.

לקורלטור המבוסס על אפנון דלתא יש רק שני מצבים -1, +1 והוא כולל את החלק של C_1', C_1'' בלבד.

מערכת השעונים של הקורלטור (ציור 36) כוללת TIMER מתוכנתת 8253 מתוצרת חברת INTEL. הוא כולל 3 מערכות זמן מתוכנמות. לצורך האוטוקורלטור השתמשנו ב-2 מערכות והשלישית שימשה לצורך המכפל במיקרו מחשב של מערכת הכיווץ. 2 המערכות תוכנתו לאופן עבודה בו הן מופעלות ע"י אות חיצוני ומעלות את קו היציאה שלהן למשך זמן הזה ל- TX_n כאשר $T = \frac{1}{f}$ של השעון החיצוני שמחובר למערכת שהוא 1.5MHz (חלוקת שעון ה-8085 ב-2) ו- n הוא מספר הניתן לתיכנות.

קו היציאה של מערכות הזמן הוכפל בשעון החיצוני וכתוצאה מכך התקבלה רכבת פולסים אשר ניתנה לתיכנות. כאמור כדי להפעיל את האוטוקורלטור המערכת זקוקה לרכבת פולסים באורך של 200 כאשר במאה הראשונים מתבצעת האוטוקורלציה ובמאה הבאים מכינים האוטוקורלטור להפעלה הבאה. כל אחת ממערכות הזמן מתוכנתת כך שתוציא רכבת פולסים באורך של 100. המיקרו מחשב מפעיל את מערכת A אשר סיום הפעולה שלה מפעיל את מערכת B. ע"י OR של שתיים נקבל רכבת פולסים באורך 200. ביציאת A נקבל רכבת באורך 100 אשר מסנכרנת את המונה של תוצאת האוטוקורלציה. עם עליית השעון נחסם המונה מתבצעת הזזה ברגיסטרי ההזזה ויחידת הקורלציה מחליטה על תוצאת הקורלציה הנוכחית. עם ירידת השעון המונה מוגדל, מוקטן, או נשאר ללא

שינוי עייפ תוצאת הקורלציה. ניתוב השעונים לרגיסטרים המתאימים נעשה עיי בוררים 1 + 2.

אחת למחזור של 100 דגימות יש לשנות את מצב הבוררים ולהפוך תפקידים בין רגיסטרי ההזזה. הזוג שביצע אוטוקורלציה הופך להיות הזוג שנטען באינפורמציה חדשה ולהיפך. הדבר מתבצע עיי הפקודה LDA 8200 את הופכת את מצב הפלי-פ-לופ § (צירור 39). בכל הפעלה יש לבצע ניקוי של המונה וזה מתבצע עיי טעינה של 80H (128 דצימלי למונה). תנאי התחלה זה למונה תוצאת האוטוקורלציה נוח ומתקבלת תוצאה ב-BINARY OFFSET שהיא בתחום 228-28 עשרוני. התוצאה מתקבלת בצורה רציפה ונוחה להשוואה לשם מציאת מקסימום. בטעינת אפס היינו מקבלים תוצאה ב-2'C שאינה נוחה להשוואה. לסיכום: האוטוקורלטור תלוי על המיקרו מחשב כהתקן I/O, דורש רק פקודה אחת להפעלה עבור כל מחזור אוטוקורלציה ומאפשר עם זאת מגוון של פעולות שונות עיי המיקרו מחשב כתלות בפקודה שניתנת לו.

6.3 תוכנה - PITCH DETECTOR

הבעיה העיקרית שניצבת בפני תוכנת הגלאי היא עמידה בזמנים. עליה לממש את הרשימה שניתנת בתחילת הפרק על החומרה ולסיים מחזור כזה של פעולות בין דגימה לדגימה. לרשות התוכנה עומדים כ-415 מחזורי שעון של ה-8085 שהם כ-50-60 פקודות אסמבלר בין דגימה לדגימה. על התוכנה לבצע 2 סוגים של פעולות.

א. פעולות בין דגימה לדגימה

1. דגימת האות.
2. ביצוע L.P.F. על האות הדגום.
3. בדיקת גודל האות לאחר L.P.F. לצורך חישוב CLIPPING LEVEL.
4. בדיקה ועידכון של מספר חציות אפס. (
5. עידכון ערך האנרגיה בקטע עם הדגימה החדשה. (

L.P.F. ללא (

6. ביצוע CLIPPING על דגימה של קטע קודם.
7. הפעלת האוטוקורלטור וטעינת התוצאה הקודמת.
8. ביצוע TAPPERING על תוצאת האוטוקורלציה.
9. בדיקה ועידכון מקסימום פונקצית האוטוקורלציה לפי התוצאה החדשה.
10. בדיקת סיום הקטע הנוכחי.

ב. פעולות בין קטע לקטע (אחת ל-100 דגמים)

1. החלטה על אופי הקטע לפי האנרגיה חציות האפס ומספר ההתאמות במקסימום פונקצית האוטוקורלציה. (האופי יכול להיות קולי או לא קולי).
2. החלטה על CLIPPING LEVEL חדש לפי שני הקטעים האחרונים.
3. חישוב ערך ה- PITCH לפי מקסימום פונקצית הקורלציה ואופי הקטע.
4. חישוב $1/PITCH$ שהוא גודל הצעד בחלון המשולש בדיוק של 12bit.
5. החלפת תפקידים באוטוקורלטור.

אילו היינו מכניסים את הפעולות בין קטע לקטע, ובעיקר את פעולת החילוק ($1/PITCH$) בין הדגימה האחרונה של קטע ישן לדגימה החדשה של קטע חדש אין ספק כי לא היינו מתקרבים אפילו לפעולה בזמן אמיתי.

פתרון לבעיה זו התקבל ע"י ניצול מידע על משך ה- PITCH וע"י הזנחת סבירות. מכיוון שידוע כי משך ה- PITCH המינימלי למבוגרים (יותר נכון לנשים) הוא כ-3-4 מילי שניות הרי ברור כי בתחום 0-3 מילי שניות אין צורך לבצע הפעלות חוזרות של האוטוקורלטור, לקרוא תוצאות לבצע TAPPERING על התוצאות ולחפש מקסימום. בנוסף הונחה ההנחה הסבירה כי ידיעת האנרגיה ומספר חציות האפס ב-80% מאורך הקטע מאפשרים לקבוע את אופי הקטע.

המסקנה ממה שכתוב למעלה היא שבתחום הדגימות בקטע שהוא בין 0-3 מילי שניות (20 דגימות ראשונות בכל קטע) אין צורך לבצע את השלבים 4,5,7,8,9. במשך זמן זה בוצעו כל הפעולות שצריך לבצע בין קטע לקטע. הפעולות הנ"ל צריכות להתבצע במקביל לפעולות שעדיין יש לעשות בין דגימה לדגימה. קיימים שתי פתרונות לבעיה זו.

א. עבודה עם REAL TIME INTERRUPT המוציא אותנו מהתוכנית שאנו מבצעים בין קטע לקטע לתוכנית המבצעת את הדגימה עם כל הפעולות הכרוכות בה ולאחר מכן לשארית הזמן שנותרה עד לדגימה הבאה חוזרים לתוכנית בין קטע לקטע.

שיטה זו מומשה בגלאי שהתבסס על אפנון דלתא, היא מפשטת את התוכנה בין קטע לקטע ומאפשרת ביצוע בנוחיות של פעולות שונות על תוצאת ה-PITCH כמו מיצוע של מספר קטעים אחרונים בשיטות שונות ועוד.

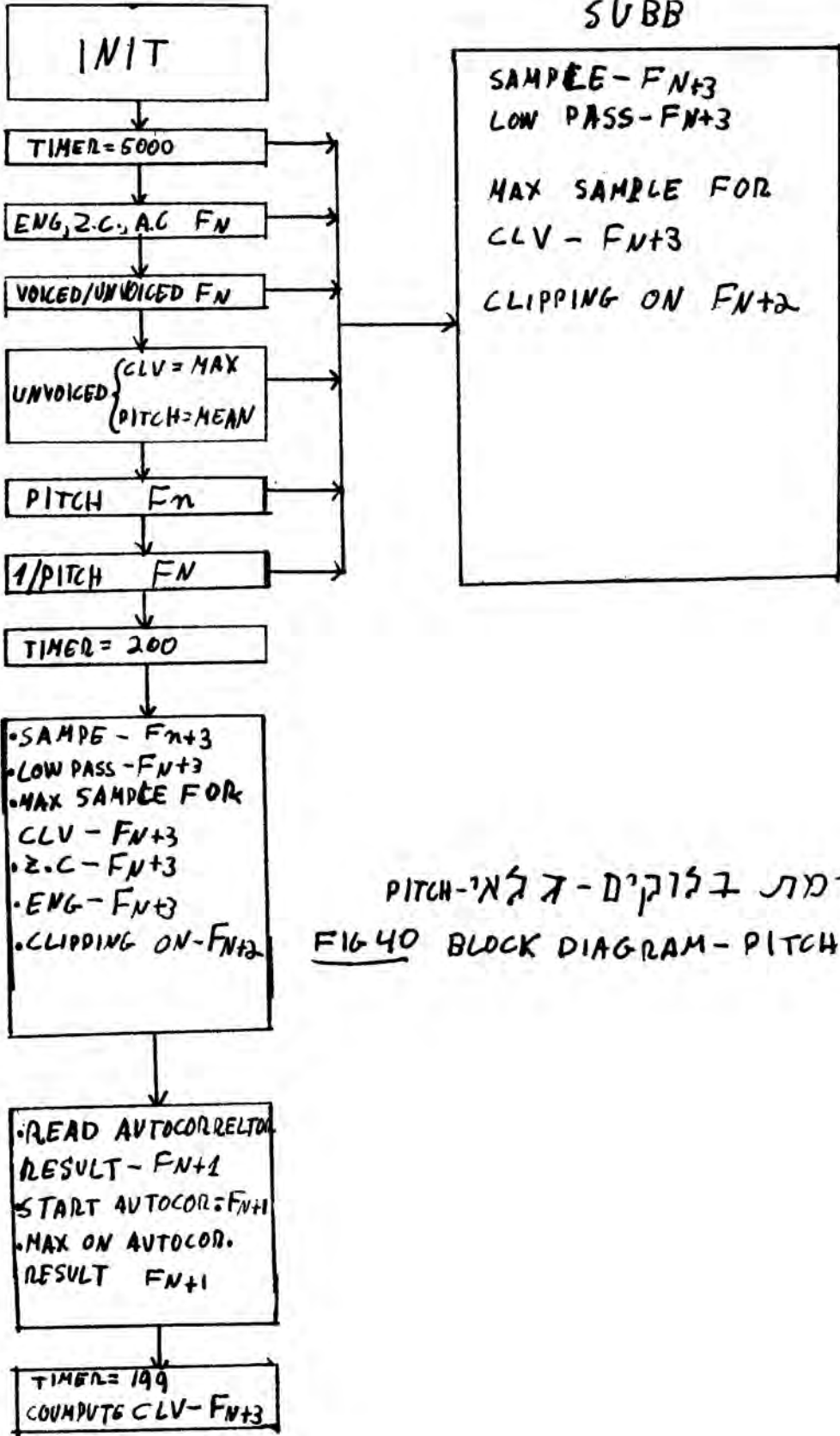
שיטה שניה היא סנכרון בתוכנה הדורש שילוב קריאות לסברוטינה המבצעת דגימה בתוך התוכנית בין קטע לקטע. (סינכרון זמן הדגימה המדוייק נדון בחאור החומרה) שיטה זו מאפשרת ניצול פחות טוב של זמן המחשב העודף ב-20 המחזורים הראשונים אולם מפשטת את החמרה. מכיוון שבמערכת שלנו דרושה הוצאת מידיות של ערך ה-PITCH ללא אפשרות של השהיות לצורך מיצוע (דבר שידרוש השהיות נוספות במערכת הכיווץ) הרי שיטה זו נבחרה ומומשה בגלאי ה-PITCH המבוסס על CENTER CLIPPING. מכיוון שלרשותינו מערכת זמן מתוכנתת באוטוקורלטור ואין אנו מעונינים בתוצאות במשך 20 המחזורים הראשונים, המערכת תוכנתה להפעלה רצופה של 20 מחזורים (20x200 פולסי שעון) ויש צורך בהפעלה אחת בלבד במשך 20 המחזורים הראשונים.

בשאר המחזורים מתבצעות כל הפעולות לפי הרשימה. המחזור ה-100 גם הוא שונה משאר המחזורים. במחזור זה (כאמור בפרק החמרה) מתוכנתת מערכת הזמן של האוטוקורלטור ל-199 בלבד וכן מחושב ה-CLIPPING LEVEL לקטע הבא. בין המחזור ה-100 לראשון מוחלף מצב האוטוקורלטור ואז מתחילה תוכנית עיבוד הנתונים של הקטע הקודם במקביל ל-20 הדגימות הראשונות.

מספר אמצעים שונים ננקטו על מנת שמשך זמן ביצוע התוכנית יוקטן. חלקים מהתוכנית המתחסיים לדגימה חיובית או שלילית נכתבו פעמיים ולפי הדגימה עוברים בקטע המתאים. דבר זה חוסך פקודות של בדיקה ושינוי בהתאם לדגימה. דבר דומה נעשה בהקשר לתוצאת ה-CLIP (+1, 0, -1) על מנת לחסוך בדיקות ופקודות קפיצה.

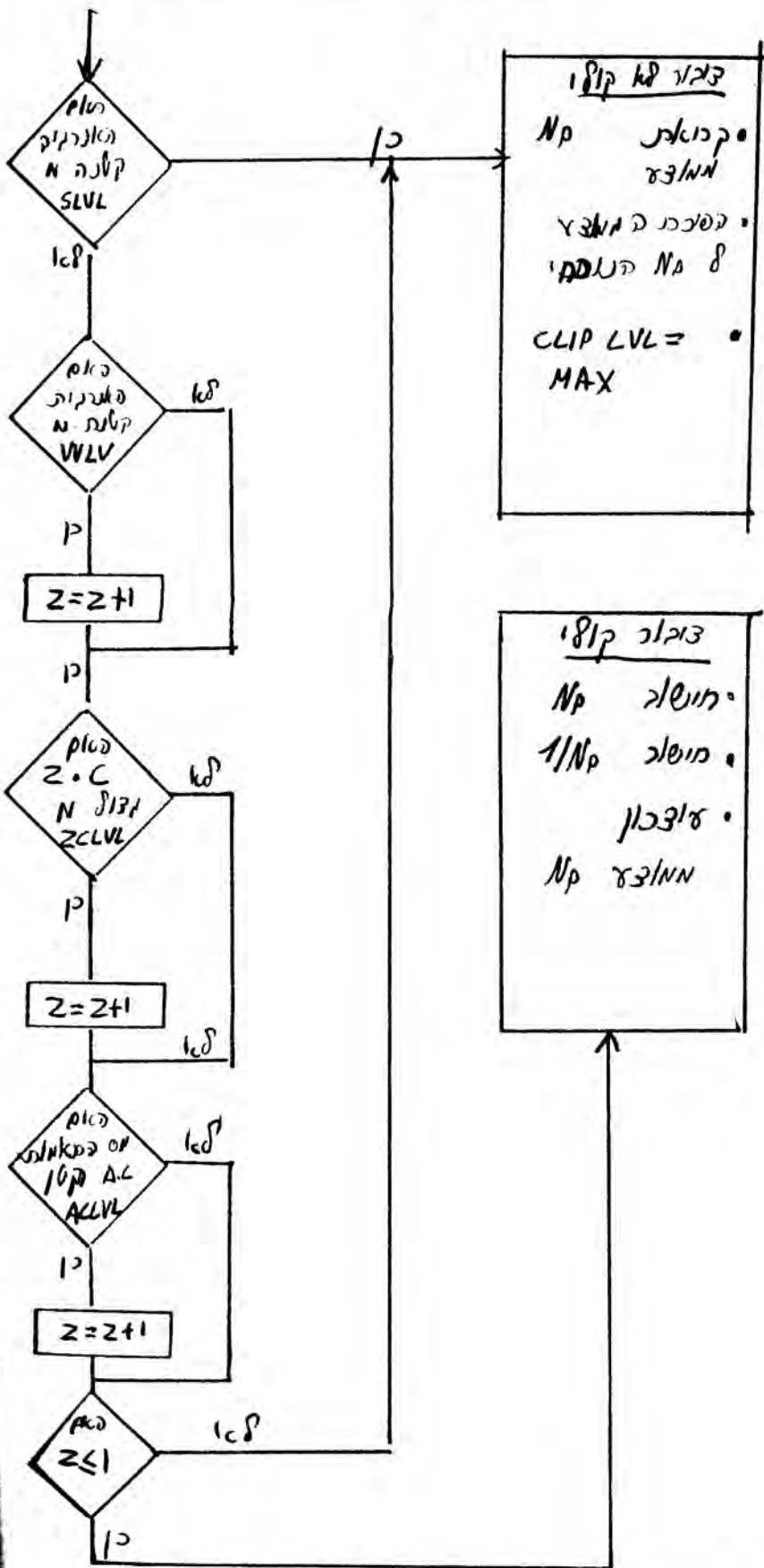
נעשה שימוש רב ככל האפשר ברגיסטרי המיקרו מחשב הן כתאים לאיחסון מידע והן כמצביעים
 לזכרון. תאי העבודה בזכרון סודרו לפי סדר הפניות לזכרון בתוכנית והפניה אליהם
 נעשת באופן עקיף דרך רגיסטרי H,L של ה-8085. פניות כאלו חוסכות כ-40%
 זמן C.P.U. יחסית לפניה ישירה (7 מחזורי שעות לעומת 13). בעזרת השיטות הנ"ל וכן
 ע"י כתיבה של התוכנית מספר רב של פעמים במספר וריאציות עד למציאת הטובה ביותר ניתן
 היה לעמוד בזמנים ואף להשאיר מספר אחוזים רזרבה לשלב הבדיקה של התוכנית (שאכן
 נוצלו כמעט עד תום). בציר 40 מתוארת דיאגרמת בלוקים כללית כל התוכנית ובצירים
 41, 42 ניתנים תרשימי זרימה של הקטעים השונים בתוכנית. בנספח מצורף הדפס של התוכנית-PIT.
 הפעלת המערכת בצורה עצמאית.

מערכת גלאי ה-PITCH הופעלה על אות דיבור בקצב של 7.2KHz. בצירים (43,44)
 ניתנת תוצאת פונקצית האוטוקורלציה במחזורים על אות הדיבור. רואים כי מחזור הפונקציה
 כמחזור ה-PITCH וניתן להוציא ממנה את האינפורמציה הדרושה.
 בציר 45 נתון קטע דיבור באורך של כ-2 שניות ויציאת גלאי ה-PITCH במקביל אליו.
 רואים כי העקיבה אחר ה-PITCH טובה. הנדנודים ביציאת ה-PITCH נובעים מרעשים
 של האדמה הדיגיטלית של המכשיר אשר גורמים לרעידות ביציאת ה-D/A שאליו מועברת
 תוצאת ה-PITCH. תוצאת ה-PITCH הועברה דרך D/A ולאחר יתן ל-A/D
 של מחשב NOVA-2 שם נדגמו והתהליך הזה גרם להיווצרות הרעש שהוא בתדר עידכון
 ה-PITCH.

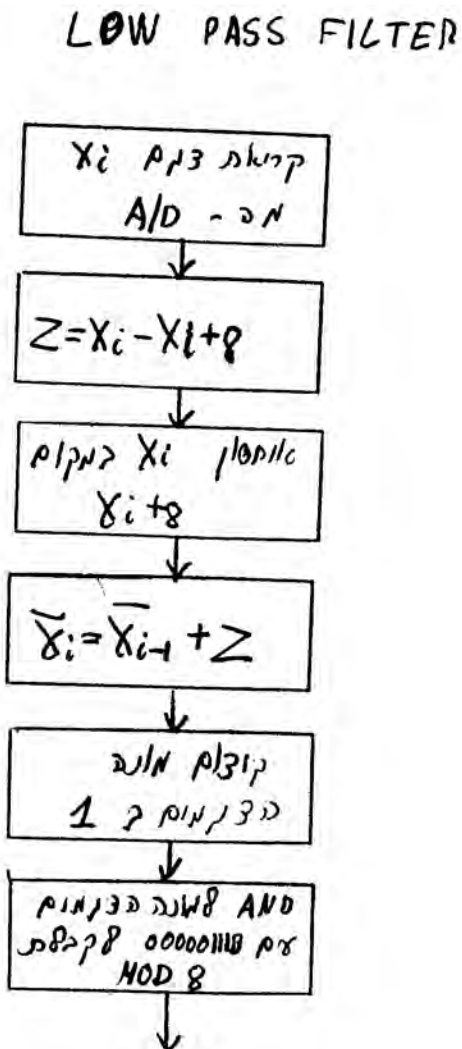
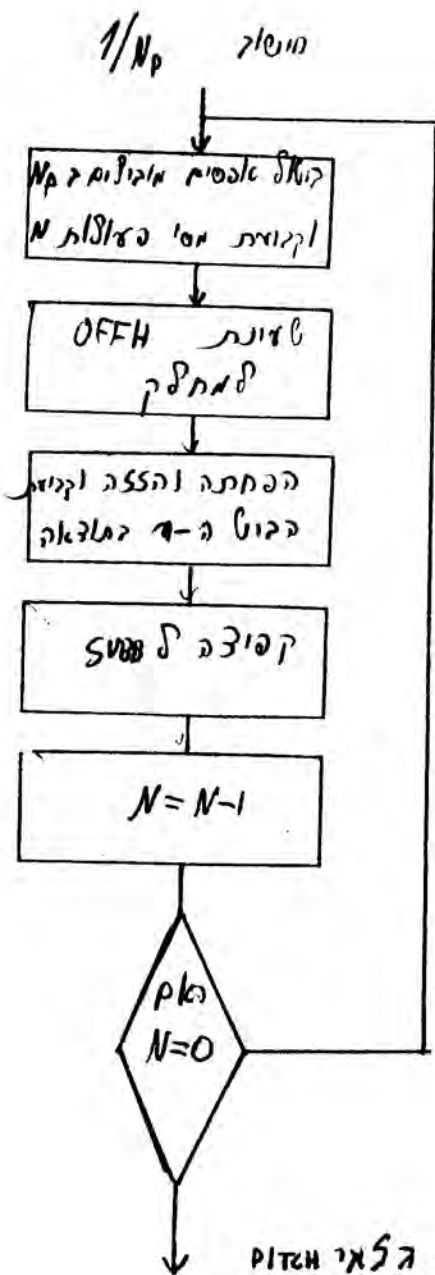


פיתוח תוכנת 40 דיאגרמת בלוקים - לאי-פיתוח
 FIG 40 BLOCK DIAGRAM - PITCH DETECTOR SOFTWARE

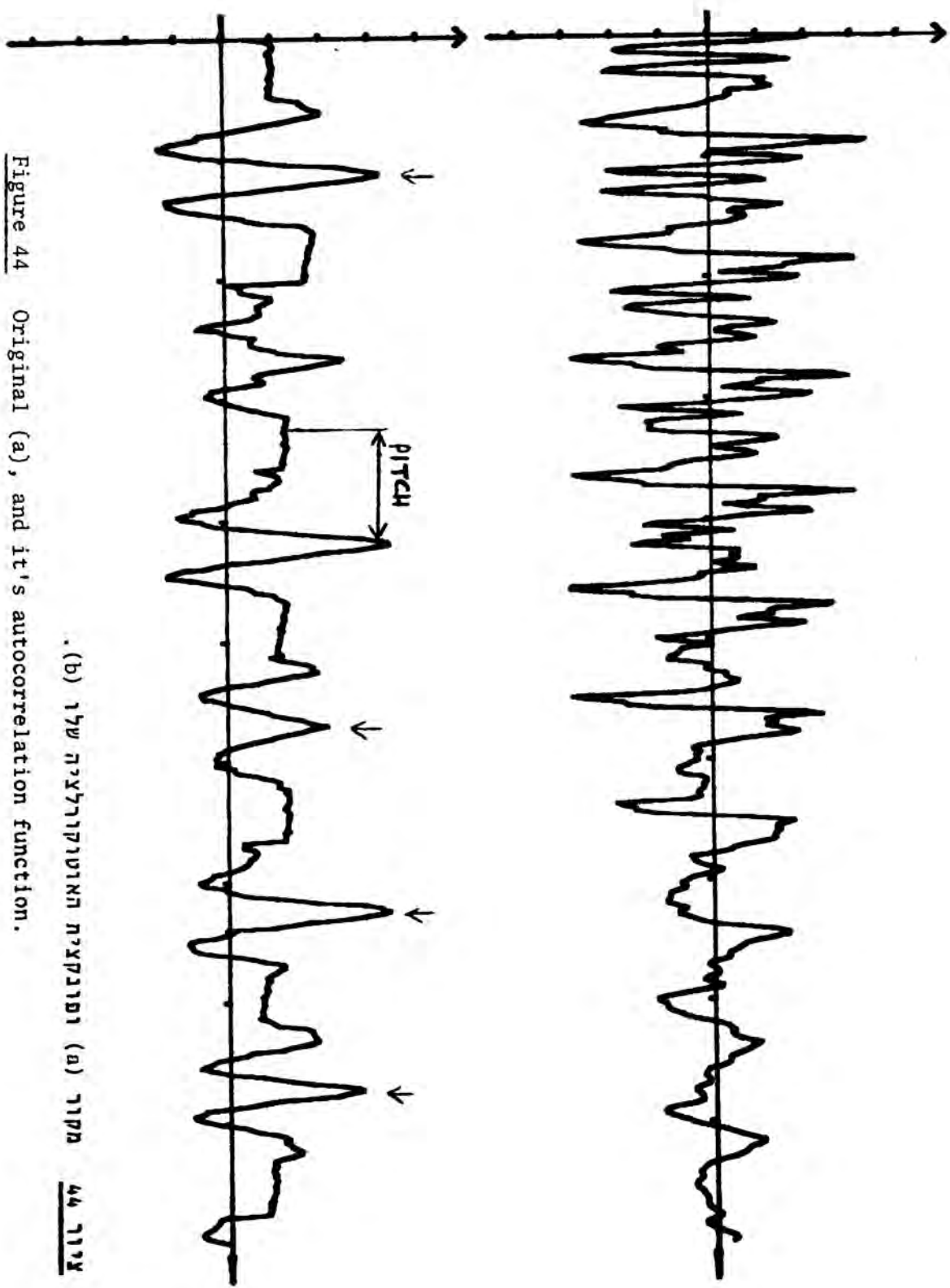
התהליך של חישוב האנליזה



ציור 41 תרשים זרימה לתוכנת גלאי פיתץ
 FIG 41 FLOW CHART - PITCH DETECTOR SOFTWARE



צירוף 42 תרשימי צרימה תוכנת גלאי אדוק
 FIG 42 FLOW CHARTS-PITCH DETECTOR SOFTWARE



מקור (א) ופונקציית האוטוקורלציה שלו (ב).

Figure 44 Original (a), and it's autocorrelation function.

44 תר

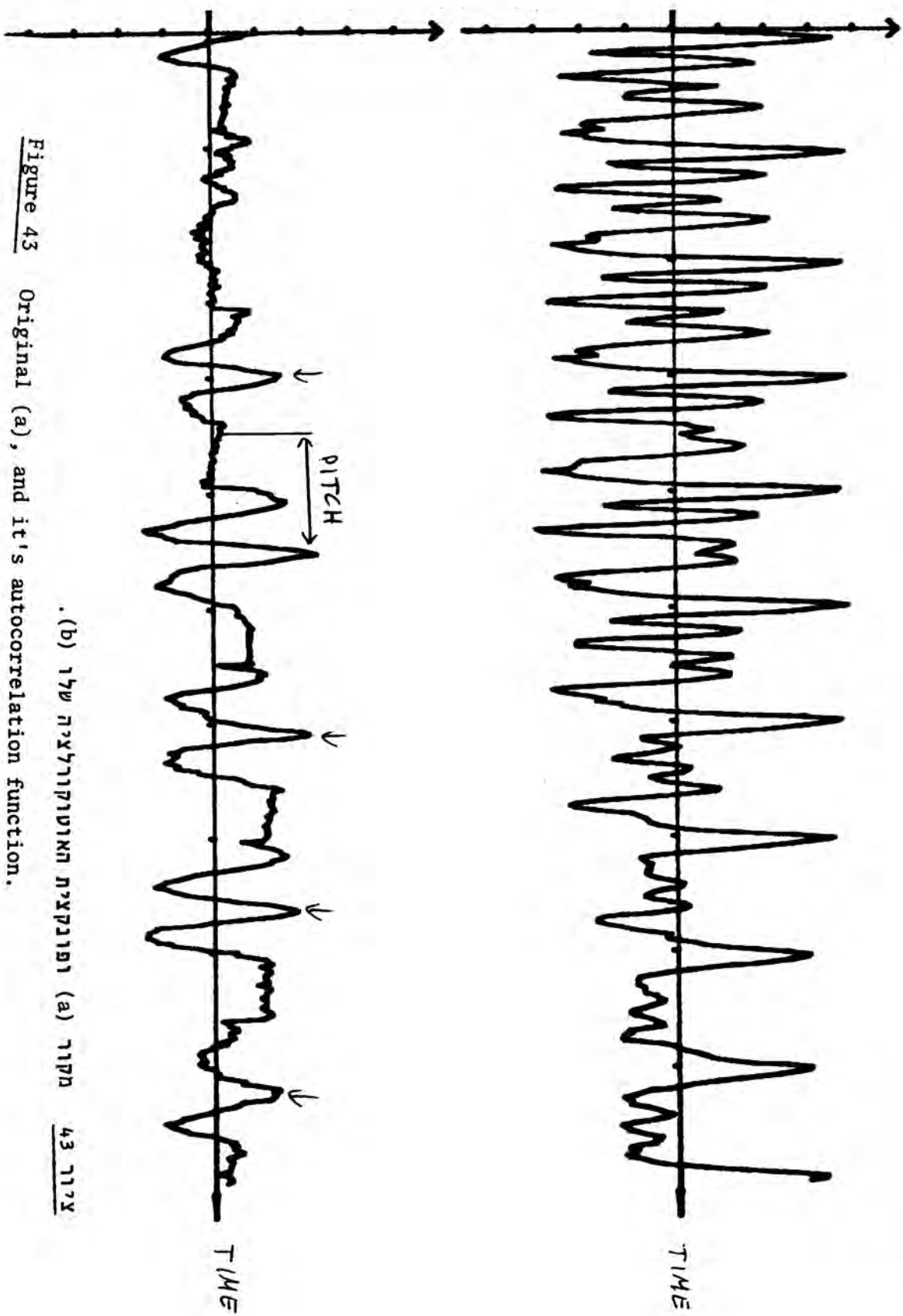


Figure 43 Original (a), and it's autocorrelation function.

תקור (a) ופונקציית האוטוקורלציה שלו (b).

ציור 43

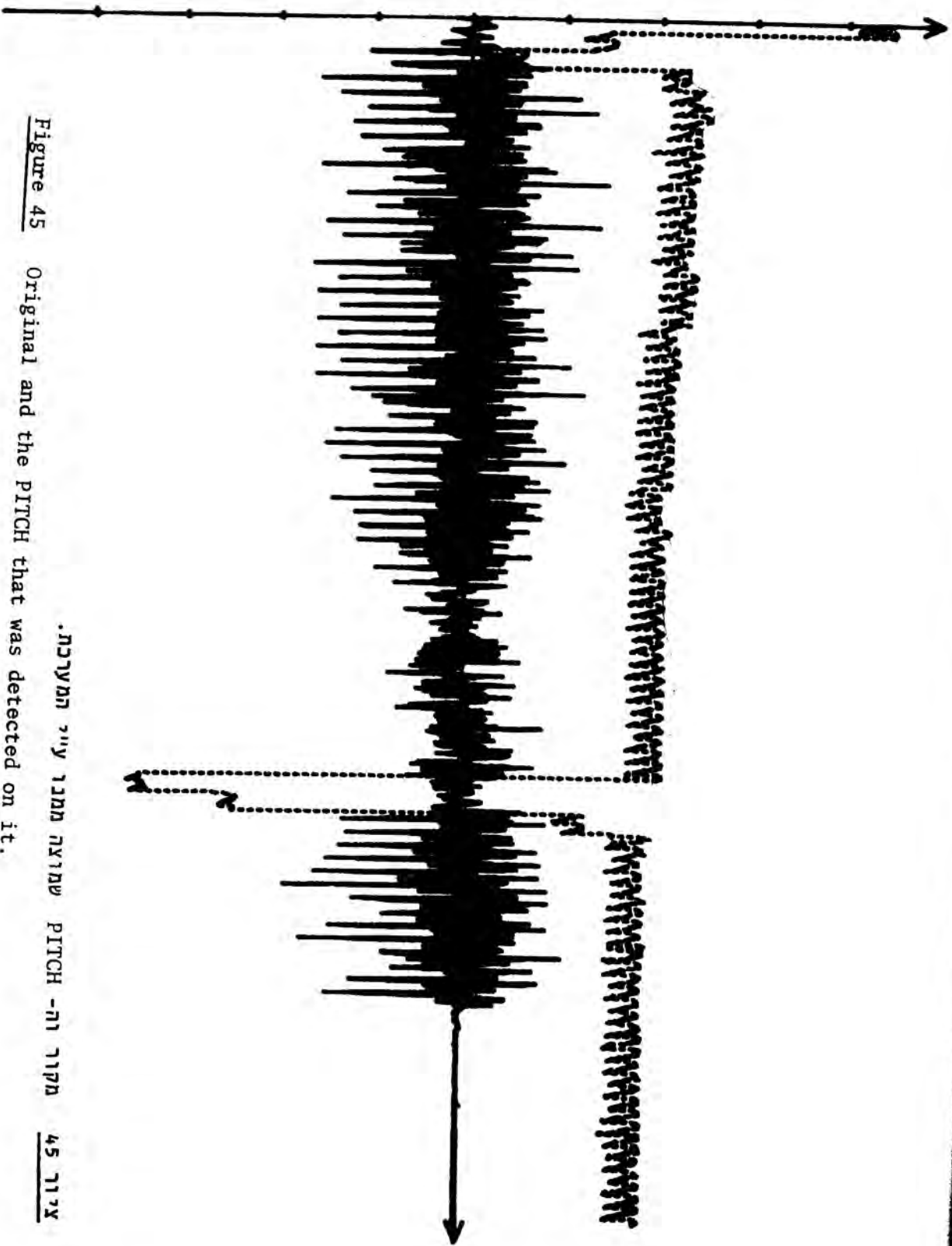


Figure 45

Original and the PITCH that was detected on it.

.תמערבתי

שמוצה ממנו ע"י התמערבתי

מקור זה - PITCH

45

פרק 7 : מערכת כיווץ/מתיחה + ADM

7.1 הקדמה

במערכת זאת ממומש האלגוריתם לכיווץ/מתיחה שמתואר בפרק 2. המערכת עובדת בארבעה אופנים שונים:

1. כיווץ

2. מתיחה

3. כיווץ + ADM

4. DADM + מתיחה

המערכת מקבלת מגלאי ה-PITCH את הערכים של ה-PITCH, המבוטא כמספר הדגמים במחזור PITCH, ו- $1/PITCH$ בדיוק של 12 סיביות. נתאר את אופן הפעולה בארבעת האופנים.

1. לבצוע כיווץ המערכת דוגמת אות דיבור בקצב f_s מבצעת עליו כיווץ ביחס של 2:1. ביציאת המערכת נקבל דגמים בקצב של $f_s/2$. רוחב הסרט של האות ביציאת המערכת הוא רק חצי מרוחב הסרט המקורי. f_s יכול לנוע בין 8KHz ל-14.4KHz. מערכת הכיווץ יכולה לעבוד אף יותר מהר אולם הקצב המקסימלי של מלאי ה-PITCH הוא 7.2KHz (המקביל ל-14.4KHz במערכת הכיווץ) על המערכת להיות מסונכרת עמו.

2. במתיחה המערכת דוגמת אות דיבור בקצב f_s ומבצעים מתיחה ביחס 2. ביציאת המערכת נקבל דגמים בקצב של $2 \times f_s$. רוחב הסרט של האות ביציאת המערכת הוא כפול מרוחב הסרט המקורי. f_s יכול לנוע בין 4KHz ל-12KHz. יש לציין כי פעולת המתיחה דורשת מספר חישובים כפול ממערכת הכיווץ עבור כל דגם כניסה.

3. כיווץ + ADM

המערכת דוגמת אות הדיבור ב-14.4KHz ומבצעת עליו כיווץ ביחס 2:1. על תוצאות הכיווץ הכוללת 7200 דגמים לשניה מבוצע ADM וביציאת המערכת נקבל זרם של סיביות

בקצב של 7.2K סיביות לשניה.

.4 D-ADM + מתיחה

המערכת מקבלת בכניסה זרם סיביות בקצב של 7200 סיביות לשניה מבצעת עליהם D-ADM וכתוצאה מכך נוצרים 7200 דגמים לשניה. לאחר מכן מתבצעת פעולת מתיחה וביציאת המערכת נקבל 14,400 דגמים לשניה ברוחב סרט של הדגמים המקוריים. תאור המערכת יתחלק ל-3 חלקים עיקריים. תאור החומרה של המערכת המטפל בכיווץ/מתיחה בלבד, תאור התוכנה של כיווץ מתיחה, ותאור התוכנה של ה-ADM בשיטת אשר מימושו הוא בתוכנה בלבד.

7.2 תאור החומרה

החומרה מתחלקת למספר חלקים עיקריים:

1. יחידת המיקרו-מחשב
2. יחידת זימון ודגימה הכוללת בתוכה A/D (מתמר מאות אנלוגי למספר בינארי) ו-D/A (מתמר של מספר לאות אנלוגי).
3. יחידת המכפל הכוללת בתוכה את מימוש החלון המשולש.

7.2.1 המיקרו מחשב

יחידת המיקרו-מחשב כוללת מיקרו מחשב מסוג INTEL-8085 זכרון תוכנית ב- EPROM בגודל 1Kx8, זכרון RAM 1Kx8 לדגמים וזכרון RAM 256x8 + 22 קווי קלט פלט (8156) המשמש לתאי עבודה לתוכנית ולהתקשרות להתקנים חיצוניים. בצירוף 46 מתוארת יחידת המיקרו מחשב.

זכרון הדגמים משמש ליצירת ההשהיה הדרושה כדי להתאים את יציאת גלאי ה- PITCH המוטה

ב- 45msec למערכת כיווץ/מתיחה. זכרון התוכנית משמש לתאי עבודה לתוכנית ולחיבור

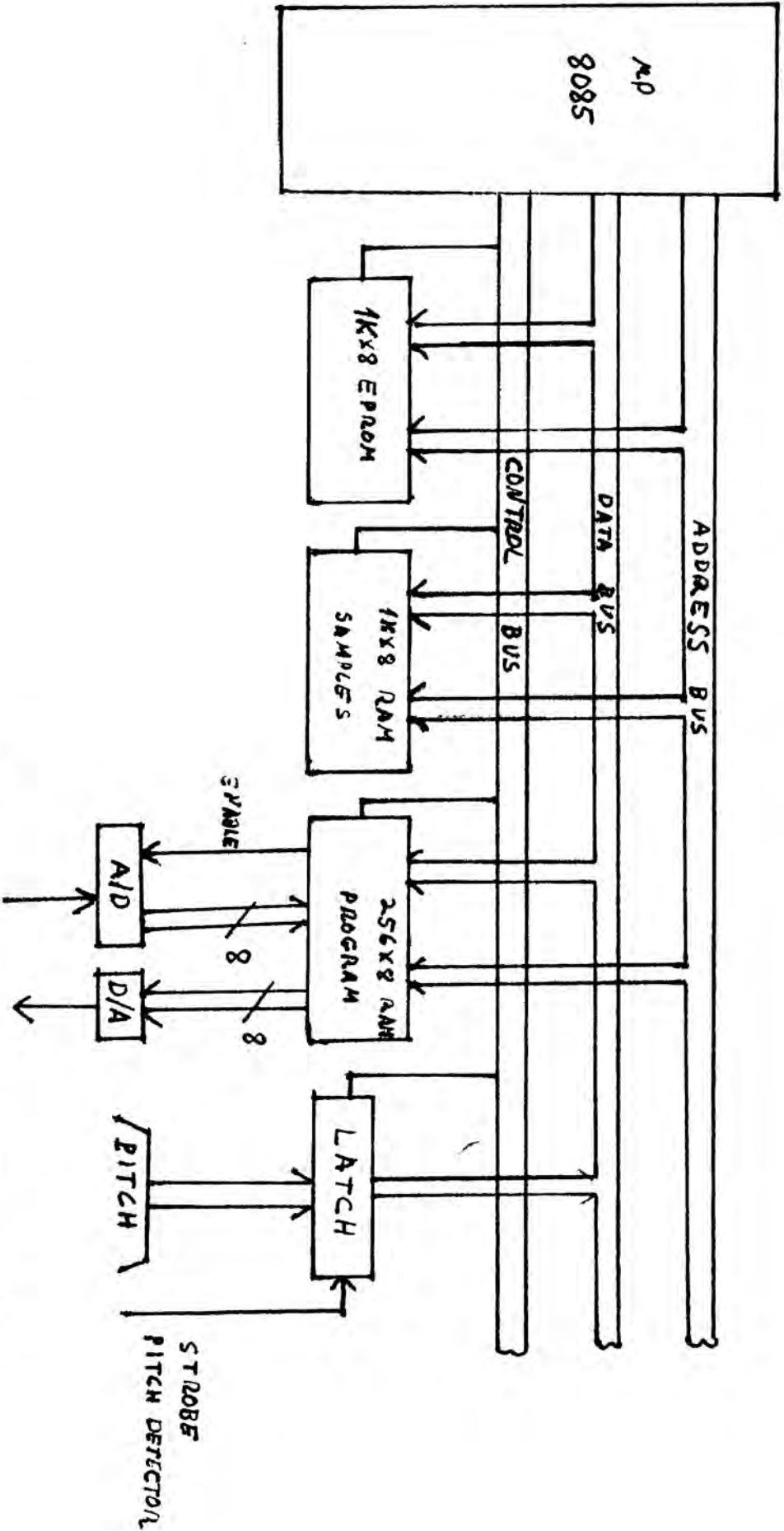


Figure 46 The micro processor of the size compress system.

הקידוד המיקרו מחשב במערכת הכיזוי/מחיימת. 46 ת"צ

ל- A/D ו- D/A. קווי הקלט/פלט של הזכרון מתוכנתים ואפשר לקבוע בתוכנה האם הם יהיו קווי קלט או קווי פלט. תכונה זו מנוצלת כדי להתאים את המערכת להיות גם משדר וגם מקלט. במצב של משדר שמונת קווי הקלט/פלט של ה- A/D מתוכנתים כקווי קלט וקו קלט/פלט נוסף של ה- RAM המחובר ל- ENABLE של ה- A/D מועלה ל- י' לוגי ומעלה את קווי ה- A/D על קווי הקלט/פלט. אל אותם קווים מתחברים גם קווי הקלט של גלאי ה- PITCH המקבלים מכאן את הדגמים מתוכם הגלאי ממצה את ה- PITCH.

במצב של מקלט היחידה קולטת זרם של ביטים מבצעת עליהם ADM ומתוכם יש למצות את ה- PITCH. במוד כזה מורד קו ה- ENABLE של ה- A/D ל- י' לוגי ושמונת קווי הקלט/פלט של ה- A/D מתוכנתים להיות קווי פלט. מכיוון שעל אותם קווים יושבים קווי הקלט של גלאי ה- PITCH הרי בצורה זו מעבירה מערכת כיווץ/מתיחה + ADM את הדגמים למערכת מיצוי ה- PITCH. כל השינויים האלו מתבצעים בתוכנה בלבד. בנוסף תלוי על BUS המחשב LATCH בן 8 סיביות שלתוכו מעדכן גלאי ה- PITCH את ערך ה- PITCH הנוכחי.

זכרון הדגימות בנוי כך ש-1024 תאי הזכרון שלו ממופים במרחב הזכרון ב-4096 כתובות והגישה ל-1024 הכתובות בתחום הזה היא ציקלית, קרי לאחר גישה לתא 1024 ניגשים שוב לתא ראשון. הצורך בדבר זה יובהר בתאור התוכנה אך יצויין רק כי צורת מיפוי זאת מונעת בדיקת גלישה של מצביעים לזכרון בכל פעם בו משנים את ערכם.

7.2.2 יחידת זימון הדגימה

יחידת זימון הדגימה זהה לזו של גלאי ה- PITCH ומתאמת את ריצת תוכנית המיקרו מחשב עם שעון הדגימה ע"י הורדת הקו READY של ה-8085 עם הפניה לקריאת תוצאת ההמרה של ה- A/D. מערכת זו זהה למערכת זימון הדגימה של גלאי ה- PITCH ושם היא מתוארת בפרוטרוט. שעון הדגימה של מערכת דגימה זו הוא כפול בתדר מזו של מערכת גלאי ה- PITCH ולכן אין אפשרות להשתמש במערכת אחת לשני המחשבים. המערכת מייצרת גם דפק בתדר קבוע להפעלת ה- A/D. בצירוף 47 מתוארת מערכת זו המבוססת על כך שהמחשב במצב המתנה עד לעלית שעון הדגימה הזמן בו מתחיל המחשב לרוץ על התוכנית. במחזור מכונה הראשון הוא

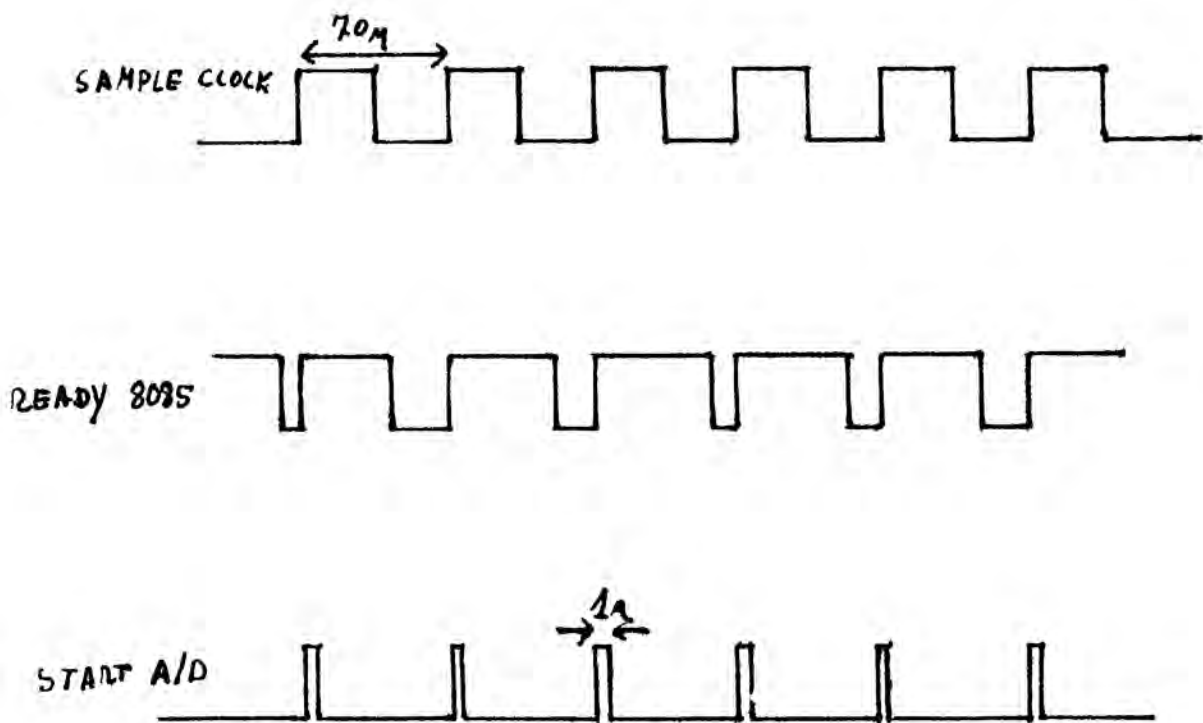
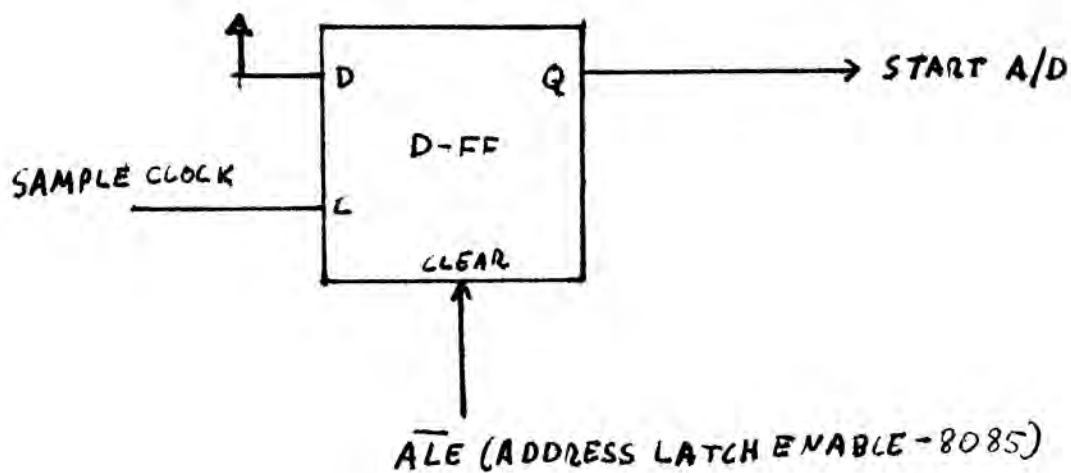


Figure 47 דיאגרמת הזימון למיקרו מחשב ולמתמר אנלוגי דיגיטלי. ציור 47

The timing diagram for the microprocessor and the analog to digital convertor.

מוציא דפק ALE (אות לדגימת קווי כתובת) עלית השעון מדליקה את הפלי-פ-פלו פ-1-ALE מכבה אותו לאחר מיקרו שניה ומתקבל דפק בקצב קבוע להפעלה ה-A/D .

בציור 4 מתוארת דיאגרמת זמנים של המערכת בו רואים שאף על פי שזמן הריצה של המיקרו מחשב בין דגימה לדגימה שונה המערכת מסנכרנת המיקרו מחשב עם השעון החיצוני. יש לציין כי פעולה זו אינה דורשת כל פקודת תוכנה ואינה גוזלת זמן חישוב. פעולת הסינכרון בין שעון חיצוני למערכת המיקרו מחשב חשובה בעיקר למערכת המשמשת כמקלט. במצב זה המערכת תקבל את השעון ממעגלי סינכרון סיבית המזהים את תדר שעון המשדר לפי האינפורמציה הנקלטת בערוץ.

7.2.3 יחידת המכפל

יחידת המכפל היא יחידת החומרה העיקרית במערכת. היחידה מתוארת בציור 48. היחידה כוללת מכפל טורי מקבילי מסוג AMD 25LS14 אשר מקבל 8 סיביות בטור ותוך 17 מחזורי שעון מתקבלת תוצאת הכפל במשלים לשנים. שעון המיקרו מחשב מחובר למכפל ואת 17 דפקי השעון יוצרת היחידה השלישית ב-TIMER המתוכנת (8253) הנמצא בגלאי ה-PITCH. מכיוון שפקודת הקריאה מהמכפל לוקחת 10 מחזורי שעון שבאחרון שבהם נקראת התוצאה הרי מבחינת המיקרו מחשב תוצאת הכפל מוכנה פקודה לאחר ביצוע הכפל ואין המכפל מעכב את פעולת המיקרו מחשב כלל. הפעולה האריתמטית בתהליך הכיווץ/מתיחה היא

$$(73) \quad X_{OUT}(N) = X_i(N) + [X_{i+N_p}(N) - X_i(N)] \quad N/N_p$$

- כאשר
- X_i, X_{i+N_p} דגמי כניסה
 - X_{OUT} דגם מכווץ/מתוח
 - N_p מספר דגמי כניסה במחזור PITCH
 - N הפעולה ה-N במחזור זה $N = 0, 1, 2, \dots, N_p - 1$

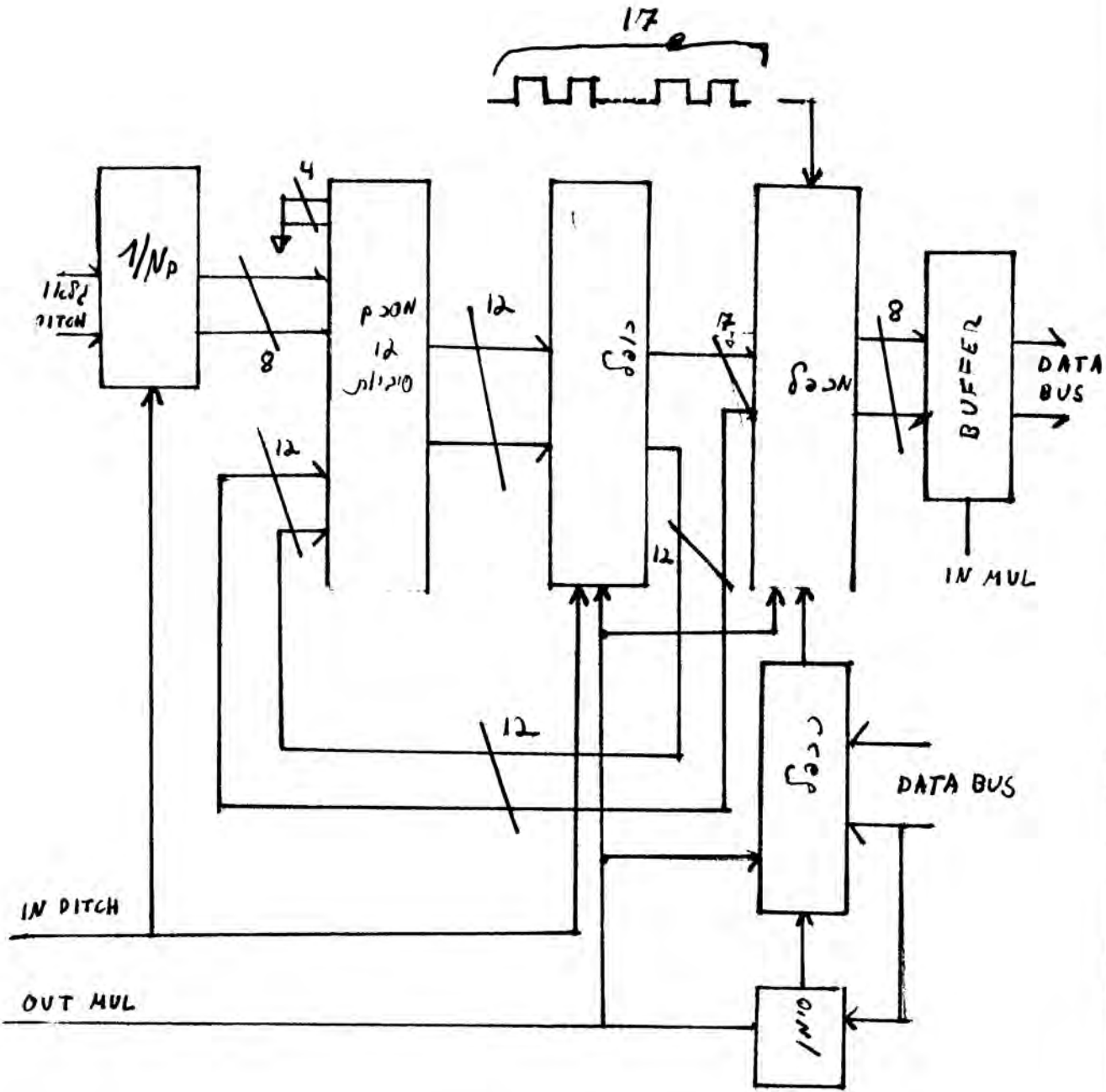


Figure 48 היחידה ליצירת החלון המשולש והמכפל הטורי מקבילי. ציור 48

The sub system of the triangular window and the multiplier.

אנו מקבלים מגלאי ה- PITCH את N_p המשמש כמונה כמה פעמים לבצע את הפעולה ואת $1/N_p$ שהוא התוספת הבסיסית בפונקציית החלון. יחידת החלון המשולש דוגמת את $1/N_p$ בזמן שהמיקרו מחשב קורא ערך PITCH חדש (אותו דפק בקרה). בכל מחזור מתבצע חוג סיכום המסכם את התוצאה הקודמת (בשלב קריאת ה- PITCH מאפסים ערך זה) עם $1/N_p$. תוצאת הסיכום מוכנסת למכפל ומוכפלת ב- $[X_{i+N_p}(N) - X_i(N)]$ שמחשב המיקרו מחשב ובצורה זו מבוצע האלגוריתם. יש לציין שמכיוון שתחום N_p הוא בין 50 ל-200 הרי דיוק של 8 סיביות בחישוב $1/N_p$ אינו מספיק.

נניח כי $N_p = 180$ נקבל ש- $1/N_p$ בדיוק של 8 סיביות הוא 1.42 (256/200) ולכן $1/N_p = 1$ יועבר למערכת כיווץ מתיחה. הדבר יגרום לכך ש- $1/N_p = 0.7$ N_p ולכן נקבל ניחות בקטע ואי רציפות במעבר מקטע לקטע. לכן גלאי ה- PITCH מחשב את $1/N_p$ בדיוק של 12 סיביות. מכיוון ש- N_p מינימלי הוא 50 נקבל ש- $1/N_p$ בדיוק של 12 סיביות הוא 82 ולכן מספיק להעביר למערכת הכיווץ רק את שמונה הסיביות הפחות משמעותיות, ומערכת הכיווץ מוסיפה את ארבעת האפסים המובילים.

ע"י הפקודה IN PITCH קורא המיקרו מחשב את ערך N_p מגלאי ה- PITCH. בעזרת אותו אות בקרה שמעלה את ערך N_p על ה- DATA BUS נדגם $1/N_p$ לתוך אוגר 8 סיביות. עם אותו אות בקרה גם מבוצע איפוס לאוגר של הכופל. ע"י הפקודה OUT MUL מוציא המיקרו מחשב את ערך הנכפל וטוען אותו לרגיסטר הזזה. מכיוון שאנו מבצעים כפל בשיטת המשלים ל-2 יש להכניס למכפל את סימן הנכפל החל מהדפק ה-8 עד ה-17 (בשבעה הראשונים מוכנסים למכפל סיביות ערך הנכפל) לכן התוסף לרגיסטר הזזה FLIP-FLOP הנטען לסימן ומחזיקו בכניסה הטורית של רגיסטר ההזזה. עם עליית אות הבקרה של ה**טעינה** נפתח אוגר הכופל והוא נטען בערכו ועוד $1/N_p$. מכיוון שהוא מסוג EDGE TRIGGER הפעולה מתבצעת פעם אחת בלבד. עם ירידת אות הבקרה הנ"ל מופעל המכפל ולאחר 17 פולסי שרון ע"י הפקודה IN MUL המיקרו מחשב קורא את תוצאת הכפל. המערכת **בדומה** לאוטוקורלטור בגלאי ה- PITCH תוכננה כהתקן המיקרו מחשב המופעל **במינימום פקודות** כאשר כל פקודה מבצעת מספר פעולות במקביל. יש לזכור שבמקרה של מתיחה **קצב הפעלה**

המערכת הזו הוא 14,400 פעמים בשניה כלומר פעם ב- 70μsec ובקצב זה הפחתה של מספר פקודות מיקרו מחשב עקרונית (המחשב מבצע גם ADM בשלב זה).

7.3 תוכנה מערכת כיווץ/מתיחה

התוכנה הבסיסית של מערכת כיווץ/מתיחה היא פשוטה, אם נניח כי רגיסטרים H,L מצביעים על $X_i(N)$ רגיסטרים P,E על $X_{i+N_p}(N)$ הרי התוכנית הבאה תבצע את האלגוריתם

LDAX	D	$A_{cc} \leftarrow (X_{i+N_p})$
SUB	M	$A_{cc} \leftarrow (X_{i+N_p} - X_i)$
OUT	MUL	MULTIPLICAND $\leftarrow (X_{i+N_p} - X_i)$
INX	D	$X_{i+N_p} = X_{i+N_p} + 1$
NOP		
IN	NUL	$(X_{i+N_p} - X_i) N/N_p \rightarrow A_{cc}$
ADD	M	$X_{i+N_p} + (X_{i+N_p} - X_i) N/N_p \rightarrow A_{cc}$
INX H		$X_i = X_{i+1}$

בזה תמה התוכנית והמערכת מוכנה להפעלה נוספת. הבעיות העיקריות של התוכנה נובעות משלבי לואי לתהליך הראשי. יש לבצע בדיקה ל- X_i ו- X_{i+N_p} אם אין גלישה מה-RAM ולעדכנם. בשלב החלפת PITCH בין מחזור למחזור יש לקרוא ערך PITCH חדש לעדכן את המונים. לשלב ההחלפה יש יתרון אחד מבחינת זמן אמיתי, אין צורך לבצע פעולת כפל מכיוון ש- $N = 0$ בביטוי N/N_p . כן באה החומרה לעזרת התוכנה. כזכור מתאור החומרה, RAM הדגימות היא ציקלי בחום של 4096 כתובות. לכן, אפשר לבדוק גלישת מבצעי הכתובת רק בזמן מעבר בין מחזור למחזור וזהו זמן עידכון PITCH. N_p המקסימלי הוא 200 ולכן מובטח

בדיקת גלישה כל 200 דגמים בעוד שהציקליות מאפשרת גלישה של 3072 דגמים. בנוסף יש לזכור כי בפעולה בזמן אמיתי קצב הפעלת המערכת יקבע לפי הפעולה הארוכה בין שתי דגימות. מכיוון שבהחלפת PITCH בין מחזור למחזור אין אנו מבצעים הפעלת מכפל ואנו מעדכנים את X_i לפי X_{i+N_p} וערך ה-PITCH החדש, יש צורך לבדוק גלישה של X_{i+N_p} בלבד. בשיטה זו מגיעים לאיזון הפעולות בין הדגימות והמערכת יכולה לפעול בקצב מהיר. על המערכת בנוסף לדגום מה-A/D דגמים ולאחסנם בזכרון הדגמים בהזדה של 600 דגמים מהמקום בו מתבצעת פעולת הכיווץ. בציוור 49 מתואר תרשים זרימה של התוכנה.

7.4 תוכנת ה-ADM CVSD

אלגוריתם ה-CVSD ממומש בתוכנה בלבד. לפני שניגש לתאור התוכנה נחזור על תאור האלגוריתם

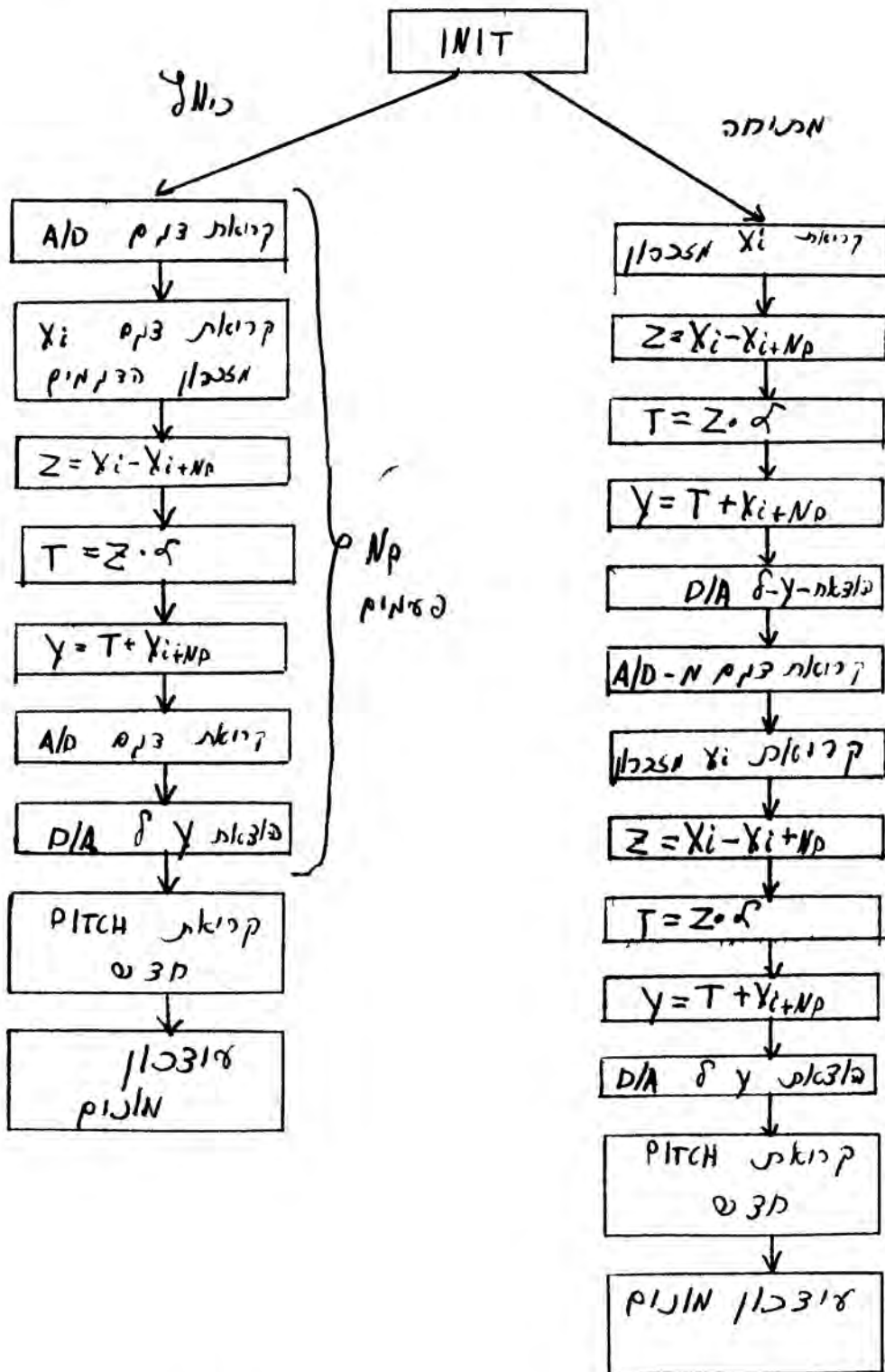
$$74) \quad e_n = \text{SIGN}(X_n - \bar{X}_{n-1})$$

$$D_n = \begin{matrix} 1 \\ -1 \end{matrix} \quad e_n = e_{n-1} = e_{n-2}$$

בכל מקרה אחר

$$75) \quad \Delta_n = \beta \Delta_{n-1} + \left(\frac{1+D_n}{2}\right) \Delta_0$$

$$76) \quad \bar{X}_n = \bar{X}_{n-1} + \Delta_n$$



צ 777 49 תרשים זרימה - מערכת כולל/מחזור

FIG 49 FLOW CHART - STRETCH/COMPRESS SYSTEM

כאשר

$$\begin{aligned}
 & \bar{X}_n - \text{האות המשוחזר} \\
 & X_n - \text{האות הדגום} \\
 & e_n - \text{הסיבית המשודרת השווה גם לסימן ההפרש בין האות הדגום לאות המשוחזר} \\
 & \Delta_0 - \text{תוספת קבועה} \\
 & \Delta_n - \text{התוספת ה- } n \\
 & \beta - \text{מקדם קבוע } 0 < \beta < 1 \text{ ובדרך כלל קרוב ל-1}
 \end{aligned}$$

במימוש האלגוריתם התעוררו מספר בעיות:

$$\begin{aligned}
 & 1. \text{ ביצוע הכפל } \beta \Delta_n \\
 & \text{בעיה זו נפתרה ע"י בחירת בתור } 1 - \frac{1}{2^K} \\
 & \text{ואז}
 \end{aligned}$$

$$\beta \Delta_n = \Delta_n - \frac{\Delta_n}{2^K}$$

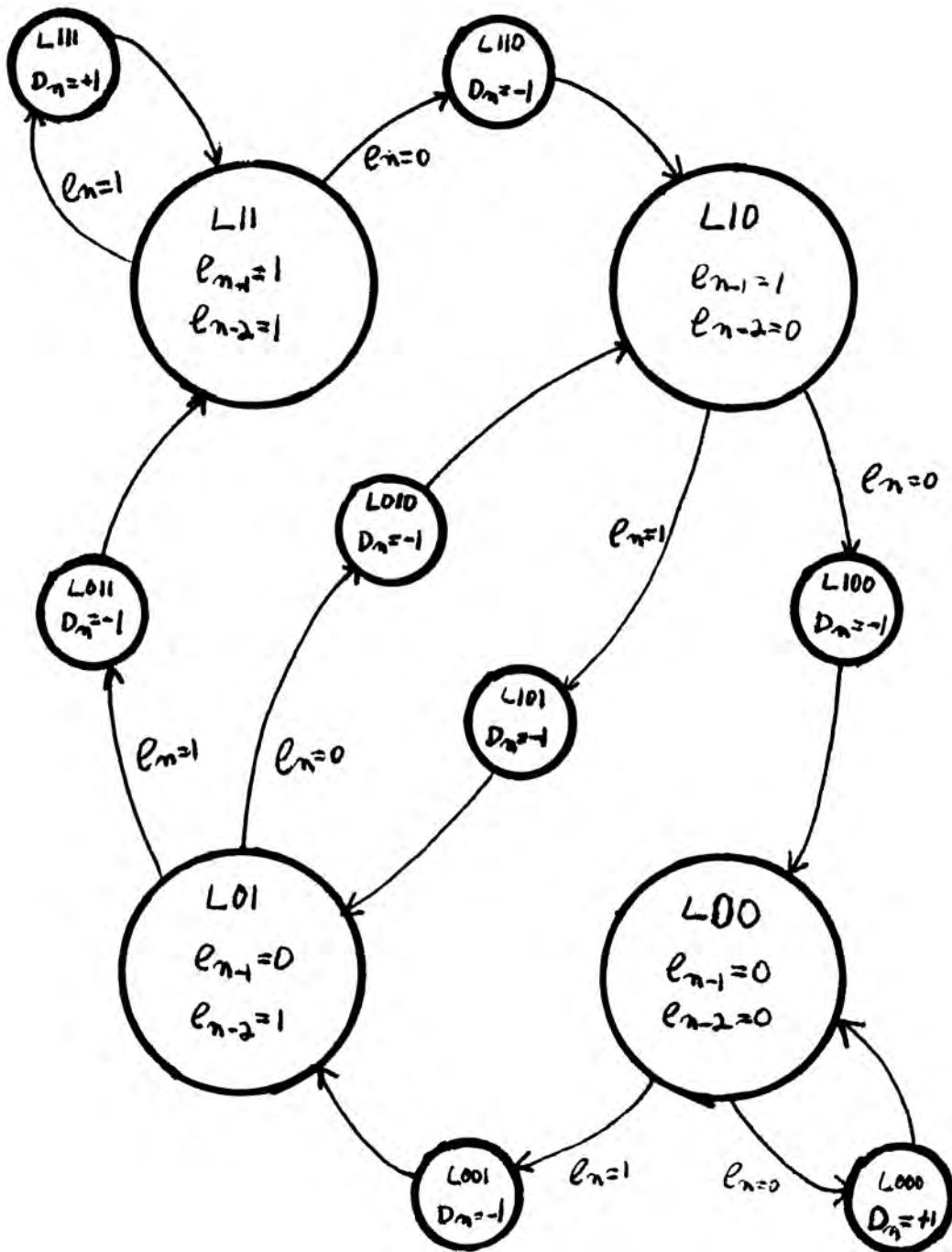
ובמקום כפל התקבלה פעולת הזזה וחסור.

בנוסף הוברר מתוצאות הסימולציות שעשינו כי ערך β הנותן תוצאות טובות הוא לערך $1 - \frac{1}{250}$ ולכן נבחר $K = 8$ ו- $\beta = 1 - \frac{1}{256}$.

מאחר והאות המשוחזר מוחזק במחשב בדיוק של 16 סיביות כפי שיוסבר בהמשך הרי כל פעולת $\beta \Delta_n$ הפכה להפחתת שמונה הסיביות היותר משמעותיות של Δ_{n-1} משמונה הסיביות הפחות משמעותיות של Δ_0 תוך העברת גלישה ליותר משמעותיות.

2. ערך Δ_0 המתאים מהסימולציות היה מספר אלפיות מהתחום הדינאמי ולכן היה צורך להחזיק את האות המשוחזר בדיוק של שש עשרה סיביות וזה גרר שכל החישובים יעשו על זוג רגיסטרים. כמובן שכל ההשוואות בין הדגם בכניסה לאות המשוחזר נעשו לפי שמונה הסיביות היותר משמעותיות.

3. בעיה שלישית היא חישוב D_n או באופן יותר מעשי האם יש צורך לחבר את Δ_0 לאות המשוחזר או לא (כש- $D_n = 1$ מחברים $(\frac{1 + D_n}{2} = 1)$ וכש- $D_n = -1$ $(1 + \frac{D_n}{2} = 0)$. פעולה זו דורשת מאיתנו לזכור את e_n, e_{n-1}, e_{n-2} להשוות ביניהם ולפי התוצאה להחליט על D_n . פעולה זו היא איטית מאוד ולא היינו יכולים לעמוד בקצב הנדרש מאתנו. מכיוון שהתוכנית הבסיסית היא קצרה ואין בעיה של מקום בזכרון הפתרון היה ליצור בתוכנה מצבים שונים ועיי המצבים האלו לחשב את D_n כך לפי e_n ואז לעבור למצב הבא. דבר זה גרם להערכת התוכנית פי שלוש לערך אולם קיצר את זמן החישוב בכ- 50%. בציוור 50 מתוארת דואגרמת המצבים. מדיאגרמת המצבים רואים כי יש 4 מצבים עיקריים L00, L01, L10, L11. האינדקס של L מציין את e_{n-1}, e_{n-2} בהתאמה. לפי המצב בו נמצאים למשל L11 ולפי ה- e_n למשל 0 מחליטים על D_n שבדוגמה שלנו יהיה ו- (L110) ועוברים למצב L10. מבחינת זמן חישוב הדבר אומר לחשב את ההפרש $(X_n - \bar{X}_{n-1})$ ולפי הסימן שלו לקפוץ לקטע בו מתבצע האלגוריתם עם D_n הנכון לפי המצב ו- e_n ובסוף קטע זה יש לבצע קפיצה בלתי מותנית למצב הבא. התוכנית לכן הופכת מהירה מאוד זמן מחזור לביצוע CVSD הוא 42μ בלבד. דבר המאפשר לבצע אלגוריתם זה בלבד בקצב של 24KHz או לשלב אותו בתוכנת המתיחה כיווץ בקצב של 14.4KHz. בציוור 50 מתואר תרשים זרימה של התוכנה.

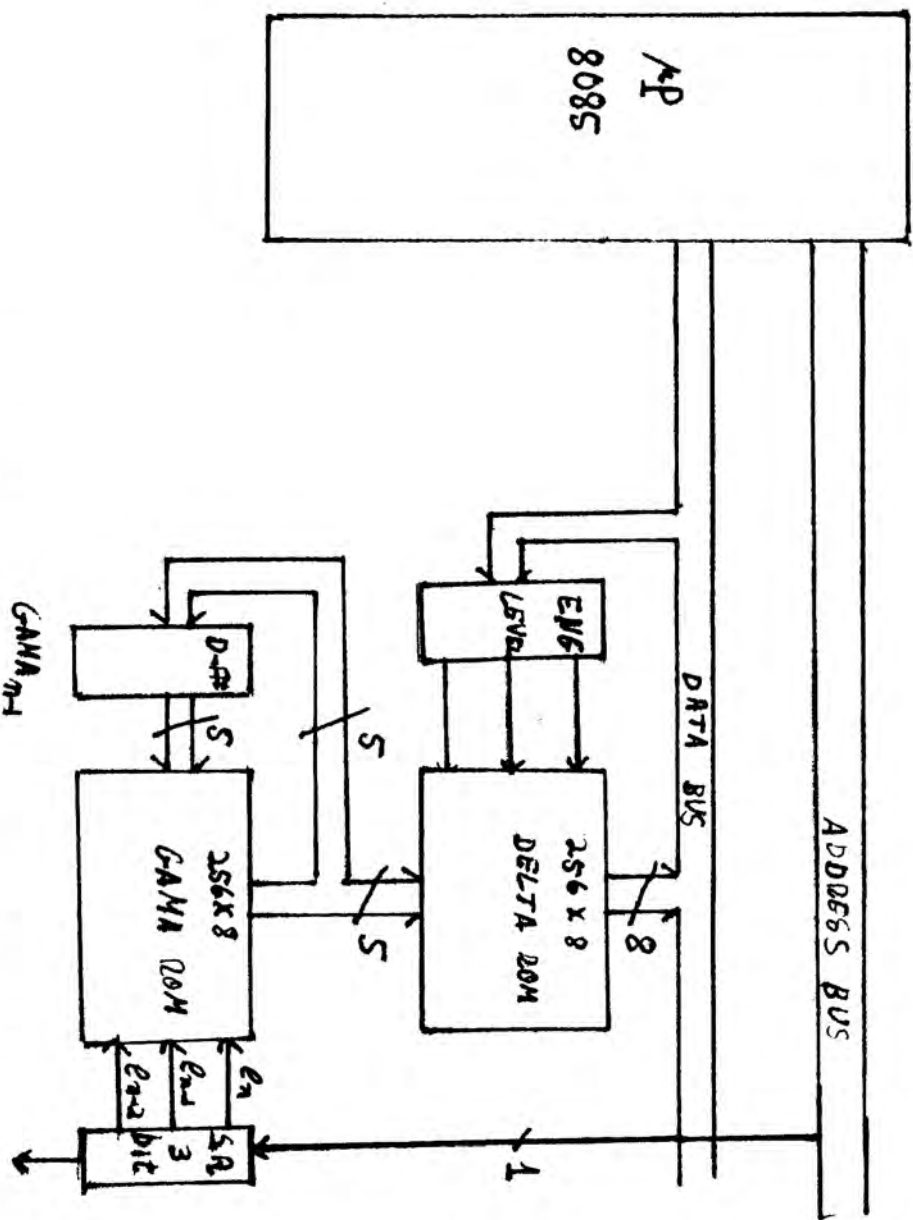


ציר 50 דיאגרמה מצבים של תהליך ה-CVSD.

Figure 49 States diagram of the CVSD algorithm.

7.5 מימוש ADM היברידי

ה-ADM ההיברידי מומש במערכת הכוללת מיקרו מחשב A/D D/A ו-2 טבלאות ב-ROM. מבנה ה-ADM ניתן בצירור 51 בשני ROMS DELTA ROM ו-1 GAMA-ROM נמצאות הטבלאות של Δ_n כפונקציה של G_n ורמת האנרגיה (לפי נוסחה 70) ושל G_n כפונקציה של G_{n-1} ו- e_{n-2}, e_{n-1}, e_n . האנרגיה חולקה ל-8 רמות והיא מחושבת ע"י המיקרו מחשב. על המיקרו מחשב לחשב את e_n ולטעון אותו לרגיסטר ההזדה לחשב את האנרגיה הנוכחית (לפי נוסחה 57). G_n מחושבת אוטומטית ע"י החמרה ועל המיקרו מחשב לקרוא את התוצאה Δ_n ולחברה או לחסרה ל- \bar{X}_{n-1} (לפי e_n) ולקבל את \bar{X}_n . התהליך הנייל הוא פשוט ויכול להתבצע בקצב גבוה.

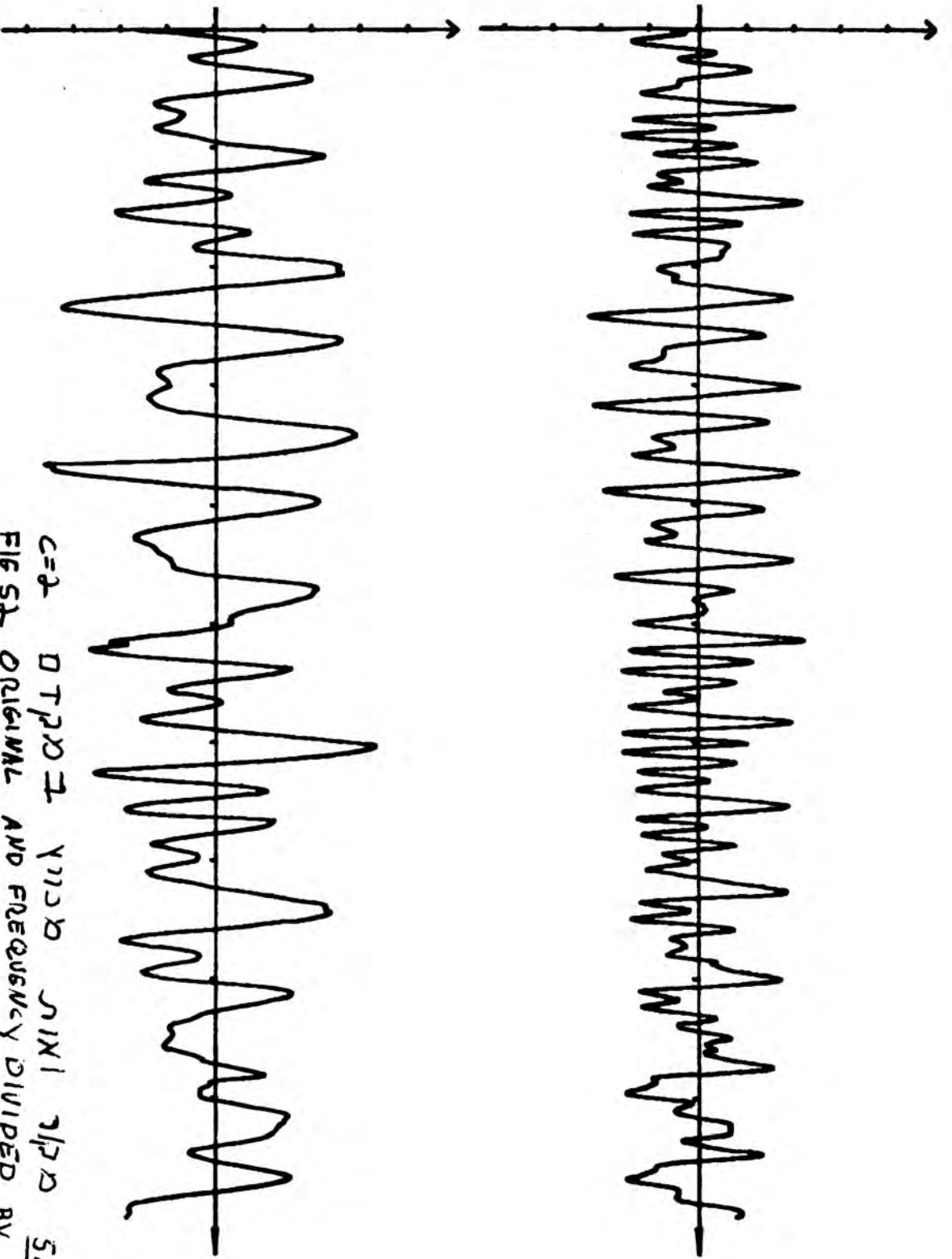


הערכת מערכת ADM היברידית
FIG 51 HYBRID ADM SYSTEM

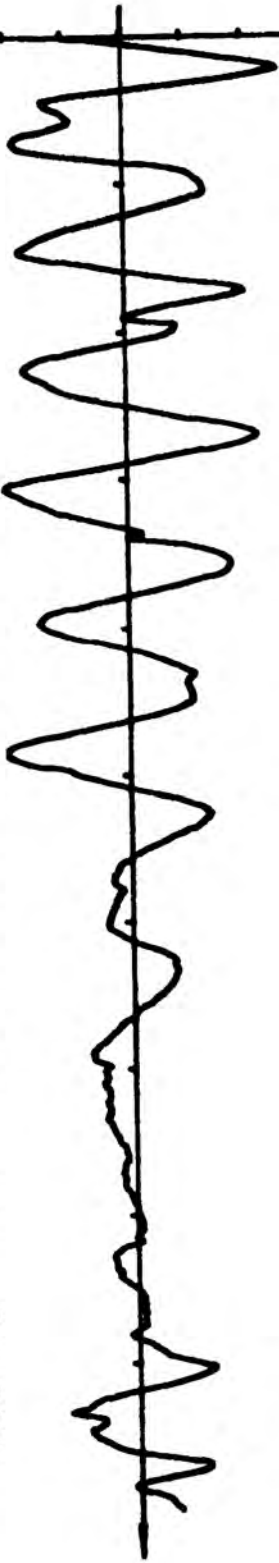
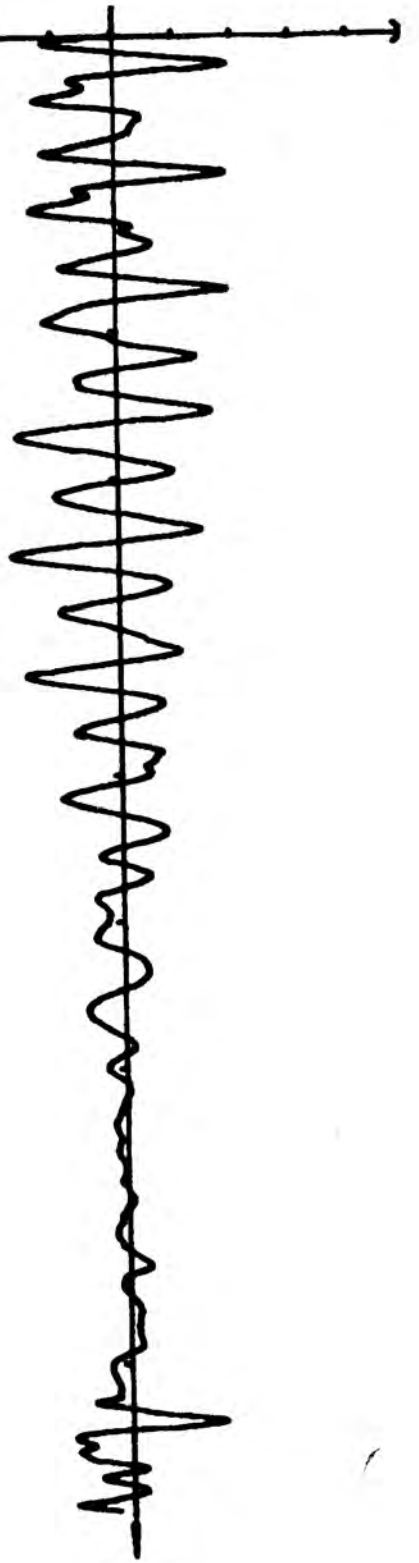
פרק 8 : תוצאות ומסקנות

המערכת הופעלה על אותות דיבור ובציורים 52, 53, 54, 55 ניתנות תוצאות שהוקלטו מן המערכת. הבעיות שהתעוררו עם הפעלת המערכת היו בעיקר בעיות של רעש שנלווה לאות. חלק מהרעש נבע מבעיות של הפרעות ספרתיות על האות האנלוגי. גם לאחר ניפוי ההפרעות נותר רעש חזק והפרעות רבות. הוברר כי העובדה שעבדנו בדיוק של 8 סיביות ודגמנו אות במלוא התחום הדינמי גררה שגיאות של גלישה מ-8 הסיביות. בפעולות החיבור והחיסור שדורש האלגוריתם. נעשה שינוי בתוכנה לפיו הדגמים מחולקים ב-2 לפני ביצוע האלגוריתם ומוכפלים ב-2 ולאחר סיום פעולת הכיוון או מתיחה לשם קבלת מלוא התחום הדינמי ביציאה.

פעולה זו ביטלה לחלוטין את הרעש החזק אולם בגלל הקטנת הדיוק וביצוע פעולה כפל, פעולת חיסור ופעולת חיבור על כל דגם, דבר שגרם לאי דיוק בחישוב היציאה, הופיע רעש רקע הרבה יותר חלש ביציאה שאין להמנע ממנו. מהאזנה לקטעים שכווצו ונמתחו התקבלה איכות דיבור סבירה והיתה עקיבה טובה אחרי ה- PITCH דבר שגרר מעט עיוותים ב- PITCH של האות המכווץ ובזיהוי הדובר. המסקנות המתבקשות הן לעבור למיקרו מחשבים של 16 סיביות ולבצע את כל החישובים בדיוק גבוה יותר. המיקרו מחשבים בני 16 סיביות המופיעים כיום בשוק כוללים גם פקודת כפל מהירה. ניתן לכן להקטין את רמות החומרה של מערכת הכיוון מתיחה תוך הגדלת הדיוק וטיב התוצאות ע"י מימוש המערכת בעזרת מיקרו מחשב מסוג זה. במערכת גלאי ה- PITCH ניתן להוסיף אלגוריתם למיצוע תוצאת ה- PITCH על מספר מחזורים. דבר זה יסנן את רוב השגיאות בגלאי ה- PITCH ויבטיח עקיבה טובה אחרי ה- PITCH. את תוכנת ה- ADM ניתן להחליף במערכת או תוכנה מתוחכמת יותר אשר תאפשר הקטנה גדולה יותר בקצב השידור. יש לציין כי ניתן להוסיף את מערכת כיוון/מתיחה לכל מערכת קיימת וע"י כך להקטין את קצב השידור של אותה מערכת ב-2.



$c=2$ DTPDZ YNDZ SIXI JPDZ SA 7MS
 FIG. 52 ORIGINAL AND FREQUENCY DIVIDED BY
 FACTOR $c=2$



C=2 DT 2 Y 1120 J 1X 1 7170 53 711X
Fig. 53 ORIGINAL AND ENERGY DIVIDED BY
 FACTOR C=2

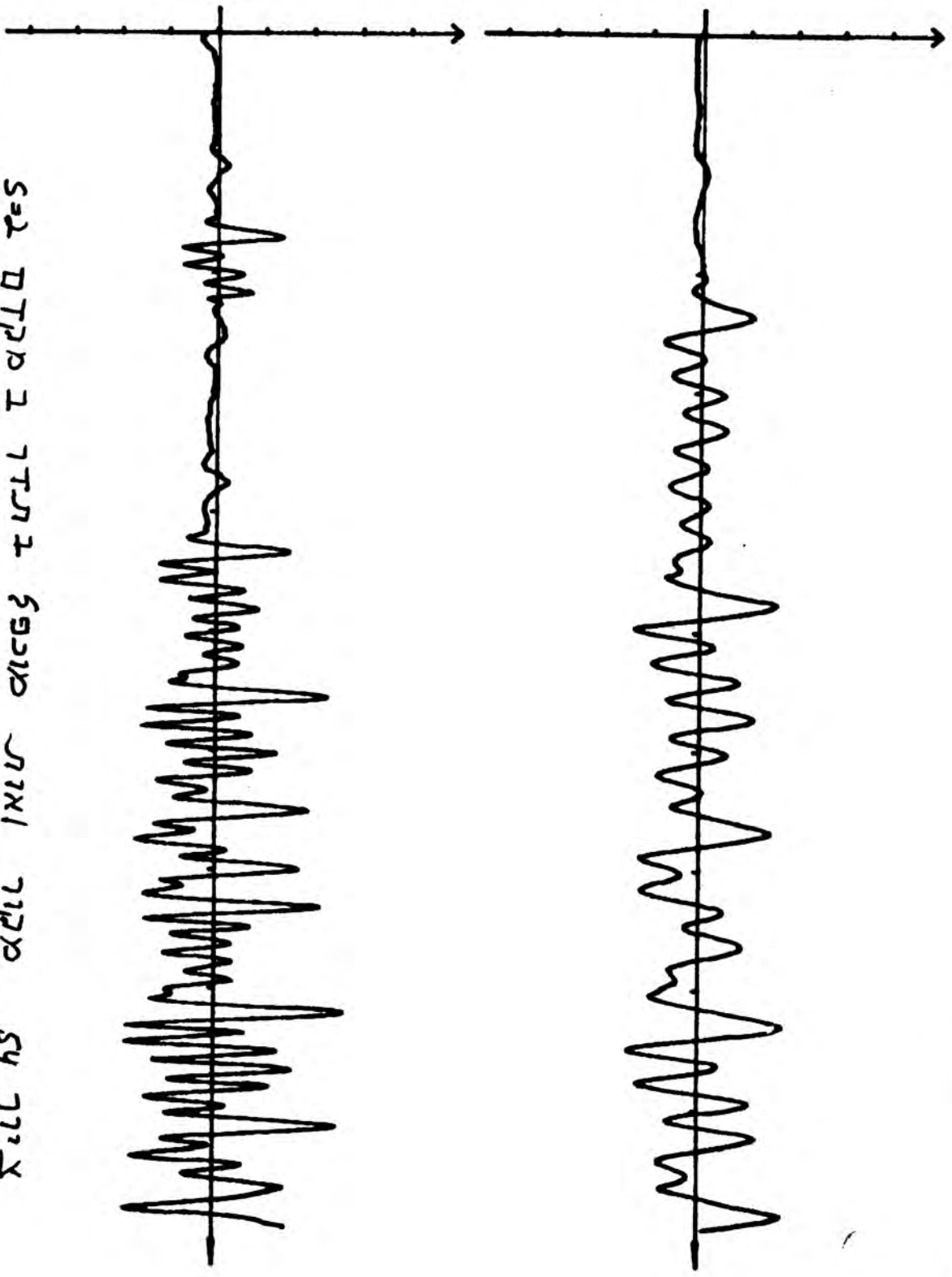
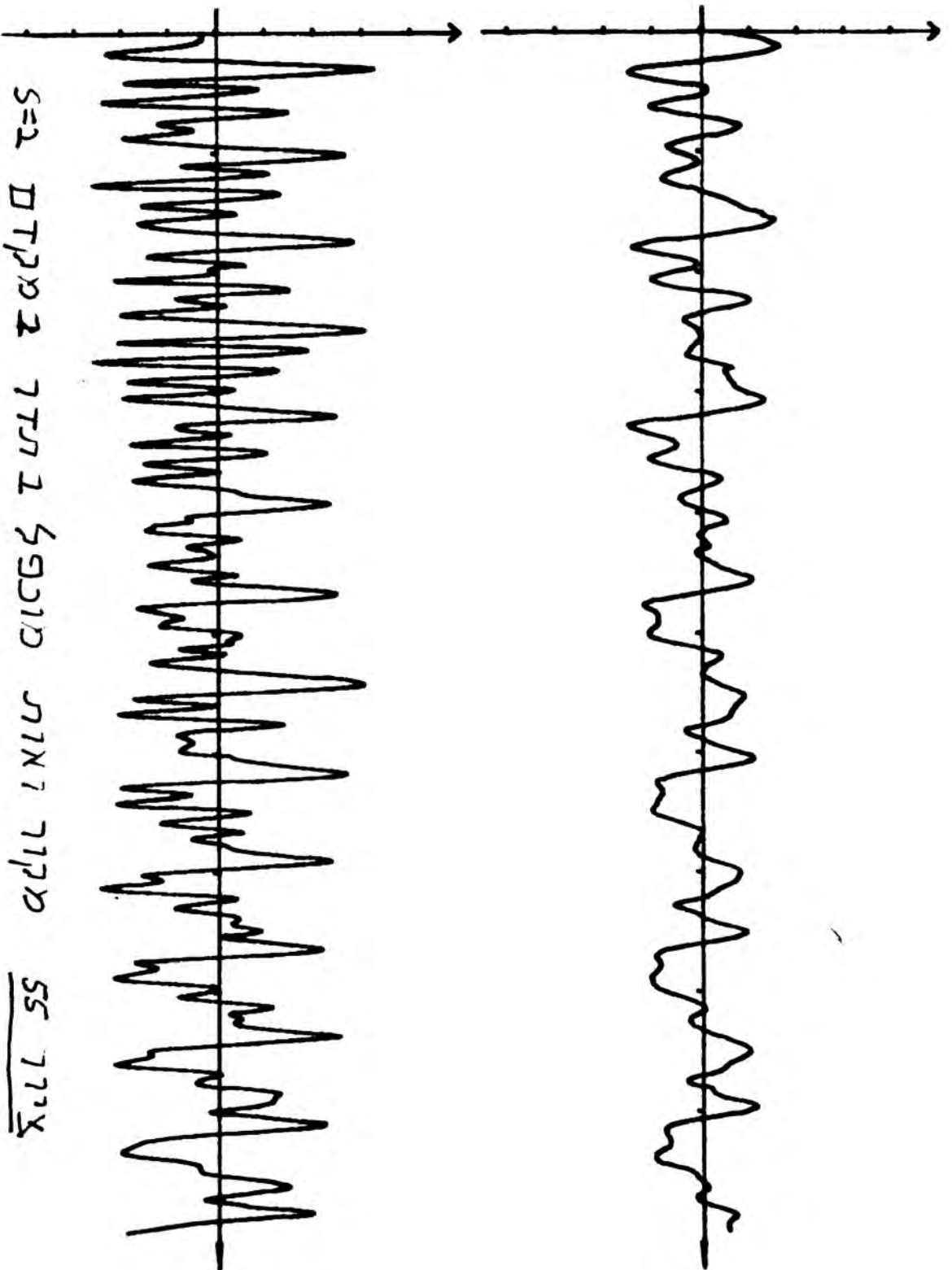


FIG 54 ORIGINAL AND FREQUENCY MULTIPLICATION WITH FACTOR $S=2$



$S=2$ $\text{DT} \times 2$ $\text{TT} \times 2$ $\text{SS} \times 2$ $\text{JTX} \times 2$ $\text{TK} \times 2$ $\text{SS} \times 2$

FIG 55 ORIGINAL AND FREQUENCY MULTIPLICATION WITH
 FACTOR $S=2$

נספח א' : תוכניות סימולציה על מחשב

נספח זה כולל הדפסי תוכניות המחשב ששימשו לסימולציות של המערכות השונות שתוארו בעבודה זו. כל התוכניות כתובות בשפת Fortran 5 והורצו במחשב NOVA 2 הקיים במעבדה לעבוד אותות בטכניון.

PTCLP - 1

```

C
C          ***** PTCLP *****
C***** PROGRAM FOR PITCH TRACKING USING CENTER CLIPPING *****
C***** AND AUTOCORRELATION *****
C
C
C
C
C
C          LOGICAL IL
C          INTEGER WIND1, LAST(16), DECFAC, LWIND, PT(2000)
C          INTEGER WIND, NDELTA, N2, NOFRAM, SUM, NPITCH
C          INTEGER INPUT1(275), INPUT2(275), NA1(128), NA2(128)
C          ACCEPT " # OF ZERO CROSSING : ", NGZC
C          ACCEPT "LOW WIND LIMIT ", LWIND
C          LWIND=LWIND/2
C          ACCEPT "SIL. LEVEL ", NEGL
C          ACCEPT "CLIPPING LEVEL ", CLPLVL
C          ACCEPT "# OF BLK ", NOFRAM
C          NOFRAM=NOFRAM-2
C          DECFAC=2
C          ACCEPT "LENGTH OF LPF WIND", WIND
C          IL=.FALSE.
C          ACCEPT "COR. OUT & LPF FILE OUT YES(1) NO(0) ", L1
C          IF(L1.EQ.0) GOTO 200
C          IL=.NOT.IL
C          CALL INOUT(3, "OUT LPF FILE ", 1)
C          CALL INOUT(2, "OUT COR. -FILE", 1)
200  CALL INOUT(1, "OUT PITCH-FILE", 1)
C          CALL INOUT(0, "INPUT FILE ", 0)
C          WIND1=WIND+256
C          NSIGNO=2
C          NTG=(NGAMA+NGAML)/2
C          IND=WIND+1
C          SUM=0
C          I9=0
C          N2=0
C          DO 33 J=1, 16
C          LAST(J)=0
33  CONTINUE
C          DO 120 I=1, 275
C          INPUT2(I)=0
120  CONTINUE
C          DO 110 I=1, 128
C          NA1(I)=0
110  CONTINUE
C
C          ***** MAIN LOOP *****
C
C          DO 1 I=0, NOFRAM, 2
C
C          ***** READING BLOCK 0 *****
C
C          CALL RDBLK(0, I, INPUT1(IND), 1, IER)
C
C          ***** CLIPPING AND AUTOCORRELATION *****

```

PTCLP - 2

```

C
CALL CLIP(NA2, INPUT2, WIND, SUM, NCLVL, LAST, I9, IL, NZC, NEG, CLPLVL)
CALL AUTOCOR(NA2, NA1, NPITCH, M, MX, LWIND, IL)
NPITCH=NPITCH*2
C
C***** SILENCE DECISION *****
C
IF (NEG. GT. NEGL) GOTO 6
K1=-10
32 PT(I+1)=K1
GOTO 14
C
C***** V/UV DECISION *****
C
6 IF (NZC. LT. NGZC) GOTO 4
K1=0
GOTO 32
4 PT(I+1)=NPITCH
X TYPE I, " PITCH=", NPITCH
14 NPITCH1=NPITCH
X WRITE(12, 90) I, NPITCH
J=I+1
C
C***** READING BLOCK 1 *****
C
CALL RDBLK(0, J, INPUT2(IND), 1, IER)
C
C***** CLIPPING AND AUTOCORRELATION *****
C
CALL CLIP(NA1, INPUT1, WIND, SUM, NCLVL, LAST, I9, IL, NZC, NEG, CLPLVL)
CALL AUTOCOR(NA1, NA2, NPITCH, M, MX, LWIND, IL)
NPITCH=NPITCH*2
C
C***** SILENCE DECISION *****
C
IF (NEG. GT. NEGL) GOTO 7
K1=-10
22 PT(J+1)=K1
NPITCH=80
GOTO 15
C
C***** V/UV DECISION *****
C
7 IF (NZC. LT. NGZC) GOTO 5
K1=0
GOTO 22
5 PT(J+1)=NPITCH
15 NPITCH1=NPITCH
X WRITE(12, 90) J, NPITCH
X TYPE J, " PITCH=", NPITCH
1 CONTINUE
NO=NOFRAM+2
C
C***** WRITING THE PITCH *****
C
WRITE BINARY(1)(PT(I), I=1, NO)

```

SCLIP - 1

```

C
C***** CLIP *****
C
C
SUBROUTINE CLIP(NAD, INPUT, WIND, SUM, NCLVL, LAST, I9, IL, NZC, NG, CLPLVL
LOGICAL IL
INTEGER NAD(128), NG, WIND, INPUT(275), SUM, N2, LAST(16)
INTEGER NDA, ND, DECFAC
NZC=0
DO 14 I1=1, WIND
INPUT(I1)=LAST(I1)
LAST(I1)=INPUT(I1+256)
14 CONTINUE
NG=IABS(INPUT(WIND+1)/8)
INPUT(1)=SUM-INPUT(1)/WIND+INPUT(WIND+1)/WIND
MAXI=IABS(INPUT(1))
C
C***** FILTERING BY AVERGING OVER WIND POINTS *****
C***** FIND # OF ZERO CROSSING *****
C***** FIND THE MAXIMOM IN THE BLOCK *****
C
DO 15 I1=2, 256
NZ1=0
IF (INPUT(I1+WIND). GT. 0) NZ1=1
NZ2=0
IF (INPUT(I1+1+WIND). GT. 0) NZ2=1
IF (NZ1. NE. NZ2) NZC=NZC+1
NG=NG+IABS(INPUT(WIND+I1)/8)
INPUT(I1)=INPUT(I1-1)-INPUT(I1)/WIND+INPUT(I1+WIND)/WIND
IF(IABS(INPUT(I1)). GT. MAXI) MAXI=IABS(INPUT(I1))
15 CONTINUE
SUM=INPUT(256)
C
C***** DECISION ON CLIPPING LEVEL *****
C
IF(NZC. GT. 70. OR. NG. LT. 800) MAXI=MAXI+2000
NLVL=INT(NCLVL*CLPLVL)
IF(MAXI. LT. NCLVL) NLVL=INT(MAXI*CLPLVL)
NCLVL=MAXI
X TYPE "CLVL=", CLPLVL, " LVL=", NLVL, " ZC=", NZC
C
C***** CLIPPING THE BLOCK *****
C
DO 116 I=1, 256
IF(INPUT(I)) 217, 217, 218
217 INPUT(I)=INPUT(I)+NLVL
IF(INPUT(I). GT. 0) INPUT(I)=0
GOTO 116
218 INPUT(I)=INPUT(I)-NLVL
IF(INPUT(I). LT. 0) INPUT(I)=0
116 CONTINUE
IF(IL) CALL WRBLK(3, I9, INPUT(1), 1, IER)
I9=I9+1
DO 16 I=1, 128
NAD(I)=0
IF(INPUT(2*I). GT. 0) NAD(I)=1
IF(-INPUT(2*I). GT. 0) NAD(I)=-1
16 CONTINUE
RETURN
END

```

SAUTCLP - 1

```

C
C***** AUTOCORRELATION *****
C
      SUBROUTINE AUTOCOR(NAD1,NAD2,NPITCH,M,MX,LWIND,IL)
      LOGICAL IL
      INTEGER NAD1(128),NAD2(128),NPITCH,INDEX(8),MAX(0:9),MINDEX(3)
      * ,MMAX(3),MX,ZERO(128),NAD(256)
      DO 140 I=1,128
      NAD(I)=NAD2(I)
      NAD(128+I)=NAD1(I)
      ZERO(I)=0
140    CONTINUE
      IF(IL) WRITE BINARY(2) (ZERO(I),I=1,LWIND)
C
C***** AUTOCORRELATION *****
C
      DO 28 I1=0,7
      MAX(I1+1)=0
      INDEX(I1+1)=I1*8+LWIND
      DO 28 I2=1,8
      NOUT=0
      I3=LWIND+I1*8+I2
      DO 25 I4=0,127
      NOUT=NOUT+NAD(256-I4-I3)*NAD(256-I4)
25    CONTINUE
      IF(IL) WRITE BINARY(2) NOUT
      IF (MAX(I1+1).GT.NOUT) GOTO 28
      MAX(I1+1)=NOUT
      INDEX(I1+1)=I3
28    CONTINUE
C
C***** TAPPER THE AUTOCOR. FUNCTION ****
C
      DO 150 I=1,8
      MAX(I)=MAX(I)-I/2+1
150    CONTINUE
      DO 33 I1=1,3
      I3=1
C
C***** FINDING THE MAX. OF AUTOCOR. *****
C
T      DO 32 I2=1,8
      IF (MAX(I3).GE.MAX(I2)) GOTO 32
      I3=I2
32    CONTINUE
      MMAX(I1)=MAX(I3)
      MINDEX(I1)=INDEX(I3)
X      TYPE I1,"MAX",MINDEX(I1),MMAX(I1)
X      WRITE(12,80)I1,MINDEX(I1),MMAX(I1)
      MAX(I3)=0
      MAX(I3-1)=0
      MAX(I3+1)=0
33    CONTINUE
      MX=MMAX(1)
      LWIND1=64-LWIND
      IF(IL) WRITE BINARY(2) (ZERO(I),I=1,LWIND1)

      GOTO 34
34    NPITCH=MINDEX(1)
X      TYPE "MAXCOR=",MMAX(1)
      RETURN
X80    FORMAT(I5,"MAX=",I5,I5)
      END

```

PADMCVSD - 1

```

C          ***** PADMCVSD *****
C          ***** CVSD---ADM *****
C
C
C
C
      INTEGER B(3), ER, ERL(2), RL(2)
      DIMENSION D(2)
      INTEGER XA(256), XB(256), Y(256)
      LOGICAL IXA, IXB
      INTEGER DEVICEI(5), DEVICEO(5), CHANNEL
      DATA TAU1, TAU2, TAU3/.02, 1.5E-3, 6.8E-3/
      CALL INOUT(3, "IN FILES FILE ", 0)
      CALL INOUT(4, "OUT FILES FILE", 0)
      CALL INOUT(2, "OUT-MSE FILE: ", 1)
      ACCEPT "# OF INPUT BLOCKS =?", NIB
      ACCEPT "NORMALIZED BASIC STEP = ? ", DELTA
      ACCEPT "NORMALIZED MAX. STEP SIZE = ? ", DMAX
      ACCEPT "NORMALIZED GAIN = ? ", GA
      DMAX=DMAX*2047.
      ACCEPT "BIT RATE (KBIT/SEC) = ? ", BR
      ACCEPT "STD. (0) OR DELAYED DECISION (1) ADM ? ", NDL
      T=1./(BR*1000.)
      BETA=1.-T/TAU1
      ALPHA=1./(1.+T/TAU2+T/(2.*TAU3))
      DELTA=DELTA*6.*(T/TAU1)*2047./5.
      GAMA=GA*ALPHA*T/TAU2
      TYPE "ALPHA=", ALPHA, " BETA=", BETA, " GAMA= ", GAMA, " DELTA= ", DELTA
      DO 111 I=1,5
      DEVICEI(I)=32*257
      DEVICEO(I)=32*257
111      CONTINUE
1000     FORMAT(5A2)
800     READ(3,1000,ERR=802) DEVICEI
802     IF(DEVICEI(1).EQ.48*257) GOTO 801
      CHANNEL=0
      CALL FOPEN(CHANNEL, DEVICEI)
      READ(4,1000,ERR=805) DEVICEO
805     DELETE DEVICEO
      CALL FOPEN(1, DEVICEO)
      ER=0
      ES=0.
      KB=0
      IB=1
      R=0.
      DN=-1.
      IB=1
      DELA=0.
      DO 30 I=1,3
30      B(I)=1
      CALL RDBLK(0,0, XA, 1, IER)
      IXB=.TRUE.
      IXA=.FALSE.
50      CONTINUE
      IF(IXB) CALL RDBLK(0, IB, XB, 1, IER)
      IF(IXA) CALL RDBLK(0, IB, XA, 1, IER)

```

```

IXA=. NOT. IXA
IXB=. NOT. IXB
IF( IER. EQ. 9) IB=NIB
IB=IB+1
IF(NDL. EQ. 1) GOTO 140
DO 140 I=1,256
DELA=BETA*DELA+DELTA*(1. +DN)/2.
IF(DELA. GT. DMAX) DELA=DMAX
R=ALPHA*R+B(3)*DELA*GAMA
Y(I)=INT(R)
IF(IXA) ER=XA(I)-Y(I)
IF(IXB) ER=XB(I)-Y(I)
ERF=FLOAT(ER)
ES=ES+ERF*ERF
B(3)=ISIGN(1, ER)
DN=-1.
IF(B(3). EQ. B(2). AND. B(3). EQ. B(1)) DN=1.
B(1)=B(2)
B(2)=B(3)
140 CONTINUE
IF(NDL. EQ. 0) GOTO 150
DO 148 I=1,256
DO 142 IL=1,2
M=2*IL-3
DN=-1.
IF(M. EQ. B(2). AND. M. EQ. B(1)) DN=1.
D(IL)=BETA*DELA+DELTA*(1. +DN)/2.
IF(D(IL). GT. DMAX) D(IL)=DMAX
RL(IL)=INT(ALPHA*R+M*D(IL)*GAMA)
IF(IXA) ERL(IL)=XA(I)-RL(IL)
IF(IXB) ERL(IL)=XB(I)-RL(IL)
142 CONTINUE
IF(IABS(ERL(1)). GT. IABS(ERL(2))) GOTO 146
Y(I)=RL(1)
R=FLOAT(RL(1))
DELA=D(1)
ERLF=FLOAT(ERL(1))
ES=ES+ERLF*ERLF
B(3)=-1
GOTO 147
146 Y(I)=RL(2)
R=FLOAT(RL(2))
DELA=D(2)
ERLF=FLOAT(ERL(2))
ES=ES+ERLF*ERLF
B(3)=1
147 B(1)=B(2)
B(2)=B(3)
148 CONTINUE
150 CALL WRBLK(1,KB, Y, 1, IER)
KB=KB+1
IF( IB. GT. NIB) GOTO 200
GOTO 50
200 ES=ES/(FLOAT( IB-1)*256. )
TYPE "MSE = ", ES
WRITE BINARY(2) ES
CALL FCLOS(1)
CALL FCLOS(0)
K111=0
GOTO 800
801 CALL RESET
STOP
END

```

PADM - 1

```

C      ***** PADM *****
C
C
C
C
REAL TAW,FC,BETA
INTEGER ER,R,DELMIN,DELMAX,DEL,B(5),XA(256),XB(256)
INTEGER Y(256)
LOGICAL IXA,IXB
CALL INOUT (0,"INPUT FILE :",0)
CALL INOUT (1,"OUT-REC. FILE:",1)
ACCEPT "# OF INPUT BLOCKS =?",NIB
ACCEPT "NORMALIZED MIN. & MAX. STEP SIZE =?",DMIN,DMAX
ACCEPT "# OF MEMORY BITS = ? (3 OR 5) ",NTF
ACCEPT "ENERGY WEIGHTING FACTOR = ? ",ALPHA
ACCEPT "TAW(MSEC),FC(KHZ) ",TAW,FC
BETA=1. -1./(TAW*FC)
ACCEPT "STD. MULTIPLIERS ? (Y=1,NO=0) ",NSM
IF(NSM.EQ.0) GOTO 20
AS=1.5
BD=.5
CBD=1.
OSC=.5
GOTO 22
20 ACCEPT "STEP MULTIPLIERS = ? (AS,BD,CBD,OSC) ",AS,BD,CBD,OSC
22 CONTINUE
ER=0
ES=0
KB=0
GM=DMIN
IB=1
R=0
FE=0
DO 30 I=1,5
30 B(I)=1
DELMIN = INT(2047.*DMIN)
DMAXM=2047.*DMAX
DELMAX=INT(DMAXM)
CALL RDBLK(0,0,XA,1,IER,IBLK)
IXB=.TRUE.
IXA=.FALSE.
50 CONTINUE
IF(IXB) CALL RDBLK(0,IB,XB,1,IER,IBLK)
IF(IXA) CALL RDBLK(0,IB,XA,1,IER,IBLK)
IXA=.NOT. IXA
IXB=.NOT. IXB
IF(IER.EQ.9. OR. IBLK.EQ.9) IB=NIB
IB=IB+1
I=1
120 DG=GM*2047.
IF(DG.GT.DMAXM) DG=DMAXM
DEL=INT(DG*(SQRT(FE)*ALPHA+1.))
IF(DEL.GT.DELMAX) DEL=DELMAX
R=R+B(5)*DEL
Y(I)=R
IF(IXA) ER=XA(I)-R

```


PADM - 2

```

IF(IXB) ER=XB(I)-R
ES=ES+FLOAT(ER)*FLOAT(ER)
B(5)=ISIGN(1,ER)
IF(B(5).EQ.B(4).AND.B(4).EQ.B(3)) GM=GM*AS
IF(B(5).NE.B(4).AND.B(4).EQ.B(3)) GM=GM*BD
IF(B(5).EQ.B(4).AND.B(4).NE.B(3)) GM=GM*CBD
IF(B(5).EQ.B(3).AND.B(5).NE.B(4)) GM=GM*OSC
NB52=0
IF(NTF.EQ.3) GOTO 130
IF(B(5).NE.B(4).AND.B(4).EQ.B(3).AND.B(3).EQ.B(2)) NB52=1
IF(NB52.EQ.1.AND.B(2).EQ.B(1)) GM=GM*CBD/BD
130 CONTINUE
IF(GM.LT.DMIN) GM=DMIN
DO 135 J=1,4
135 B(J)=B(J+1)
FE=FE*BETA+ABS(FLOAT(Y(I)))
IF(I.EQ.256) GOTO 150
I=I+1
GOTO 120
150 CALL WRBLK(1,KB,Y,1,IER,IBLK)
KB=KB+1
IF(IB.GT.NIB) GOTO 200
GOTO 50
200 ES=ES/(FLOAT(IB-1)*256.)
TYPE "MSE = ",ES
CALL RESET
STOP
END

```

נספח ב' : תוכניות אסמבלר למערכת

נספח זה כולל הדפסי תוכניות האסמבלר של המיקרו מחשבים במערכת הממומשת.

PITDETECT.LST - 1

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 1

STMNT	LOC	OBJECT CODE	SOURCE STATEMENT
1	0070		CLVUV EQU 70H
2	000A		CLPMIN EQU 10
3	0080		ENGLL EQU 80H
4	0001		ENGL EQU 1
5	000A		ZCL EQU 10
6	0005		ACL EQU 5
7	0001		OPIT EQU 9101H
8	0000		PITCH EQU 8800H
9	0000		C8156 EQU 9100H
10	0003		COUNT EQU 8403H
11	0000		CNT0 EQU 8400H
12	0001		CNT1 EQU 8401H
13	0002		CNT2 EQU 8402H
14	0011		HRAM EQU 11H
15	0002		A2D EQU 0D102H
16	0000		SAC EQU 8800H
17	0001		SAC00 EQU 0A001H
18	0000		SACM1 EQU 0A000H
19	0002		SACP1 EQU 0A002H
20	0001		AC00 EQU 0A801H
21	0002		ACP1 EQU 0A802H
22	0000		ACM1 EQU 0A800H
23	0048		CNT0C EQU 048H
24	004A		CNT1C EQU 4AH
25	0049		CNT2C EQU 49H
26	0046		AC98T EQU 46H
27	0000		STATUS EQU 8200H
28	0026		AC100 EQU 26H
29	0088		MUL17 EQU 88H
30	000F		A2DMK EQU 0FH
31	1100		RAM EQU 1100H
32			;
33			;
34			;
35			;
36			;
37			;
38	0000	21 00 11	LXI H, 1100H
39	0003	AF	LCC: XRA A
40	0004	77	MOV M, A
41	0005	2C	INR L
42	0006	C2 03 00	JNZ LCC
43	0009	3E 01	MVI A, 01
44	000B	32 00 00	STA C8156
45	000E	26 11	MVI H, HRAM
46	0010	3E 40	MVI A, 40H
47	0012	32 71 11	STA PITV
48	0015	32 72 11	STA PITM
49	0018	32 75 11	STA LTAD+1
50	001B	1E 01	MVI E, 1
51	001D	3E 48	MVI A, CNT0C

PITDETECT. LST - 2

52	001F	32	03	00	STA	COUNT
53	0022	3E	4A		MVI	A, CNT1C
54	0024	32	03	00	STA	COUNT
55	0027	3E	49		MVI	A, CNT2C

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 2

STMNT	LOC	OBJECT CODE	SOURCE STATEMENT
56	0029	32 03 00	STA COUNT
57	002C	3E 26	MVI A, AC100
58	002E	32 00 00	STA CNT0
59	0031	32 01 00	STA CNT1
60	0034	3E 88	MVI A, MUL17
61	0036	32 02 00	STA CNT2
62	0039	16 F7	MVI D, 0F7H
63	003B	31 F0 11	LXI SP, 11F0H
64	003E	3E 4C	AGAIN: MVI A, 4CH
65	0040	32 03 00	STA COUNT
66	0043	3E 24	MVI A, 24H
67	0045	32 00 00	STA CNT0
68	0048	3E C8	MVI A, 0C8H
69	004A	32 00 00	STA CNT0
70	004D	3A 00 00	LDA SAC
71	0050	CD B5 01	CALL SUBB
72	0053	21 78 11	LXI H, LTAD+4
73	0056	7E	MOV A, M
74	0057	FE 00	CPI 0
75	0059	C2 65 00	JNZ ENCON
76	005C	2D	DCR L
77	005D	7E	MOV A, M
78	005E	FE 80	CPI ENGLL
79	0060	DA 83 00	JC UVOICE
80	0063	2C	INR L
81	0064	7E	MOV A, M
82	0065	FE 01	ENCON: CPI ENGL
83	0067	3E 00	MVI A, 0
84	0069	CE 00	ACI 0
85	006B	47	MOV B, A
86	006C	2D	DCR L
87	006D	2D	DCR L
88	006E	7E	MOV A, M
89	006F	FE 0A	CPI ZCL
90	0071	78	MOV A, B
91	0072	3F	CMC
92	0073	CE 00	ACI 0
93	0075	47	MOV B, A
94	0076	2D	DCR L
95	0077	2D	DCR L
96	0078	7E	MOV A, M
97	0079	FE 05	CPI ACL
98	007B	78	MOV A, B
99	007C	CE 00	ACI 0
100	007E	FE 02	CPI 2
101	0080	DA 9E 00	JC UVOICE: VOICE
102	0083	3A 72 11	LDA PITM

PITDETECT. LST - 3

103	0086	32	71	11	STA	PITV
104	0089	3E	70		MVI	A, CLVUV
105	008B	32	73	11	STA	LASTC
106	008E	3E	7F		MVI	A, 7FH
107	0090	32	7A	11	STA	117AH
108	0093	3E	81		MVI	A, 81H
109	0095	32	7B	11	STA	117BH
110	0098	CD	B5	01	CALL	SUBB

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 3

STMT	LOC	OBJECT CODE	SOURCE STATEMENT
111	009B	C3 B2 00	JMP GO
112	009E	3F	VOICE: CMC
113	009F	2C	INR L
114	00A0	7E	MOV A, M
115	00A1	17	RAL
116	00A2	32 71 11	STA PITV
117	00A5	1F	RAR
118	00A6	47	MOV B, A
119	00A7	3A 72 11	LDA PITM
120	00AA	1F	RAR
121	00AB	80	ADD B
122	00AC	32 72 11	STA PITM
123	00AF	CD B5 01	CALL SUBB
124	00B2	0E 17	GO: MVI C, 23
125	00B4	06 05	MVI B, 5
126	00B6	3A 71 11	LDA PITV
127	00B9	67	MOV H, A
128	00BA	07	ZERO: RLC
129	00BB	DA C9 00	JC PROG
130	00BE	67	MOV H, A
131	00BF	04	INR B
132	00C0	C3 BA 00	JMP ZERO
133	00C3	94	YSUB: SUB H
134	00C4	37	STC
135	00C5	4F	MOV C, A
136	00C6	C3 E8 00	JMP NOSUB+1
137	00C9	79	PROG: MOV A, C
138	00CA	90	SUB B
139	00CB	3D	DCR A
140	00CC	E5	PUSH H
141	00CD	C5	PUSH B
142	00CE	21 6E 11	LXI H, CNT
143	00D1	77	MOV M, A
144	00D2	CD B5 01	CALL SUBB
145	00D5	C1	POP B
146	00D6	E1	POP H
147	00D7	0E FF	MVI C, OFFH
148	00D9	2E 00	MVI L, 0
149	00DB	37	LOP: STC
150	00DC	79	MOV A, C
151	00DD	17	RAL
152	00DE	4F	MOV C, A
153	00DF	DA C3 00	JC YSUB

PITDETECT. LST - 4

154	00E2	94				SUB	H
155	00E3	DA	E7	00		JC	NOSUB
156	00E6	4F				MOV	C, A
157	00E7	3F			NOSUB:	CMC	
158	00E8	7D				MOV	A, L
159	00E9	17				RAL	
160	00EA	6F				MOV	L, A
161	00EB	E5				PUSH	H
162	00EC	C5				PUSH	B
163	00ED	26	11			MVI	H, HRAM
164	00EF	CD	B5	01		CALL	SUBB
165	00F2	C1				POP	B

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 4

STMT	LOC	OBJECT	CODE		SOURCE	STATEMENT
166	00F3	E1			POP	H
167	00F4	05			DCR	B
168	00F5	C2	DB	00	JNZ	LOP
169	00F8	7D			MOV	A, L
170	00F9	32	70	11	STA	PITDV
171	00FC	26	11		MVI	H, HRAM
172	00FE	3A	6E	11	LDA	CNT
173	0101	4F			MOV	C, A
174	0102	21	74	11	LXI	H, LTAD
175	0105	AF			XRA	A
176	0106	77			MOV	M, A
177	0107	2C			INR	L
178	0108	77			MOV	M, A
179	0109	2C			INR	L
180	010A	21	00	00	LXI	H, 0
181	010D	22	74	11	SHLD	LTAD
182	0110	22	76	11	SHLD	LTAD+2
183	0113	22	77	11	SHLD	LTAD+3
184	0116	26	11		MVI	H, HRAM
185	0118	C3	4D	01	JMP	WT25
186	011B				DS	50
187	014D	CD	B5	01	WT25:	CALL SUBB
188	0150	0D			DCR	C
189	0151	C2	4D	01	JNZ	WT25
190	0154	0E	19		MVI	C, 25
191	0156	3E	07		MVI	A, 7
192	0158	3D			W1:	DCR A
193	0159	C2	58	01	JNZ	W1
194	015C	3E	4C		MVI	A, 4CH
195	015E	32	03	00	STA	COUNT
196	0161	3E	26		MVI	A, AC100
197	0163	32	00	00	STA	CNT0
198	0166	3E	00		MVI	A, 0
199	0168	32	00	00	STA	CNT0
200	016B	3A	00	00	LDA	SAC
201	016E	3A	02	00	LDA	A2D
202	0171	EE	0F		XRI	A2DMK
203	0173	47			MOV	B, A
204	0174	3E	03		MVI	A, 3

PITDETECT. LST - 5

205	0176	3D			W2:	DCR	A
206	0177	C2	76	01		JNZ	W2
207	017A			20		DB	20H
208	017B	FE	00			CPI	0
209	017D	FA	93	01		JM	STR
210	0180	3A	71	11	COMP:	LDA	PITV
211	0183	00				NOP	
212	0184	00				NOP	
213	0185	32	00	00		STA	PITCH
214	0188	AF				XRA	A
215	0189	3A	70	11		LDA	PITDV
216	018C	1F				RAR	
217	018D	32	01	00		STA	OPIT
218	0190	C3	A5	01		JMP	LLL
219	0193	3A	71	11	STR:	LDA	PITV
220	0196	00				NOP	

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 5

STMNT	LOC	OBJECT CODE			SOURCE STATEMENT	
221	0197	00			NOP	
222	0198	32	00	00	STA	PITCH
223	019B	3A	70	11	LDA	PITDV
224	019E	1F			RAR	
225	019F	1F			RAR	
226	01A0	E6	3F		ANI	3FH
227	01A2	32	01	00	STA	OPIT
228	01A5	78			MOV	A, B
229	01A6	CD	BB	01	CALL	SUBB+6
230	01A9	C3	1C	02	JMP	LOOP
231	01AC	C3	1C	02	JMP	LOOP
232					;	
233					;	
234					;	
235					;	
236	01AF	6B			SP1: MOV	L, E
237	01B0	1C			INR	E
238	01B1	77			MOV	M, A
239	01B2	C3	EC	01	JMP	CON1
240	01B5	3A	02	00	SUBB: LDA	A2D
241	01B8	EE	0F		XRI	A2DMK
242	01BA	47			MOV	B, A
243	01BB	6B			MOV	L, E
244	01BC	7E			MOV	A, M
245	01BD	FE	00		CPI	0
246	01BF	F2	CE	01	JP	SPOS
247	01C2	2E	7B		MVI	L, TAD1+2
248	01C4	BE			CMP	M
249	01C5	D2	DA	01	JNC	SCPO
250	01C8	3A	00	00	LDA	SACM1
251	01CB	C3	DD	01	JMP	GO1
252	01CE	3E	7A		SPOS: MVI	A, TAD1+1
253	01D0	BE			CMP	M
254	01D1	DA	DA	01	JC	SCPO
255	01D4	3A	02	00	LDA	SACP1

PITDETECT. LST - 6

256	01D7	C3	DD	01		JMP	G01		
257	01DA	3A	01	00	SCPO:	LDA	SAC00		
258					G01:				
259	01DD	6B				MOV	L, E	;	**
260	01DE	2D				DCR	L	;	**
261	01DF	7E				MOV	A, M	;	**
262	01E0	6A				MOV	L, D	;	**
263	01E1	96				SUB	M	;	**
264	01E2	80				ADD	B	;	**
265	01E3	70				MOV	M, B	;	
266	01E4	F2	AF	01		JP	SP1	;	**
267	01E7	6B			SM1:	MOV	L, E	;	**
268	01E8	1C				INR	E	;	**
269	01E9	77				MOV	M, A	;	**
270	01EA	2F				CMA		;	**
271	01EB	3C				INR	A	;	**
272	01EC	2E	79		CON1:	MVI	L, TAD1	;	**
273	01EE	BE				CMP	M	;	**
274	01EF	DA	F3	01		JC	SM2	;	**
275	01F2	77				MOV	M, A	;	**

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE

6

STMNT	LOC	OBJECT	CODE	SOURCE	STATEMENT	
276	01F3	14		SM2:	INR	D ; **
277	01F4	7A			MOV	A, D ; **
278	01F5	E6	F7		ANI	11110111B ;
279	01F7	57			MOV	D, A ; **
280	01F8	C9			RET	
281	01F9	2E	7A	CP11:	MVI	L, TAD1+1
282	01FB	BE			CMP	M
283	01FC	DA	42 02		JC	CP00
284	01FF	3A	02 00		LDA	A2D ; **
285	0202	EE	0F		XRI	A2DMK
286	0204	32	6F 11		STA	A2DD
287	0207	79			MOV	A, C
288	0208	0F			RRC	
289	0209	0F			RRC	
290	020A	0F			RRC	
291	020B	0F			RRC	
292	020C	E6	0F		ANI	0FH
293	020E	6F			MOV	L, A
294	020F	3A	02 00		LDA	ACP1
295	0212	95			SUB	L
296	0213	2E	74		MVI	L, TAD
297	0215	C3	58 02		JMP	CON ; **
298	0218	2C		TL1:	INR	L
299	0219	C3	5F 02		JMP	L1
300	021C	6B		LOOP:	MOV	L, E
301	021D	7E			MOV	A, M
302	021E	FE	00		CPI	0
303	0220	F2	F9 01		JP	CP11 ; <<<== **
304	0223	2E	7B		MVI	L, TAD1+2
305	0225	BE			CMP	M ; **
306	0226	D2	42 02		JNC	CP00 ; CLIP **

PITDETECT. LST - 7

```

307
308
309
310      0229      3A  02  00      CPM1:  LDA      A2D
311      022C      EE  0F                    XRI      A2DMK
312      022E      32  6F  11          STA      A2DD      ; FROM **
313      0231      79                    MOV      A, C      ; ADC  **
314      0232      0F                    RRC
315      0233      0F                    RRC
316      0234      0F                    RRC
317      0235      0F                    RRC
318      0236      E6  0F                    ANI      0FH
319      0238      6F                    MOV      L, A
320      0239      3A  00  00          LDA      ACM1      ; <<<=== **
321      023C      95                    SUB      L
322      023D      2E  74                    MVI     L, TAD
323      023F      C3  58  02          JMP      CON
324      0242      3A  02  00          CPO0:  LDA      A2D
325      0245      EE  0F                    XRI      A2DMK
326      0247      32  6F  11          STA      A2DD
327      024A      79                    MOV      A, C
328      024B      0F                    RRC
329      024C      0F                    RRC
330      024D      0F                    RRC

```

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 7

STMNT	LOC	OBJECT CODE	SOURCE STATEMENT
331	024E	0F	RRC
332	024F	E6 0F	ANI 0FH
333	0251	6F	MOV L, A
334	0252	3A 01 00	LDA AC00
335	0255	95	SUB L
336	0256	2E 74	MVI L, TAD
337			;
338	0258	BE	CON: CMP M
339	0259	DA 18 02	JC TL1
340	025C	77	MOV M, A
341	025D	2C	INR L
342	025E	71	MOV M, C
343	025F	2C	L1: INR L
344	0260	3A 6F 11	LDA A2DD
345	0263	A8	XRA B
346	0264	F2 68 02	JP LC
347	0267	34	INR M
348	0268	A8	LC: XRA B
349	0269	47	MOV B, A
350	026A	F2 6F 02	JP EP
351	026D	2F	CMA
352	026E	3C	INR A
353	026F	2C	EP: INR L
354	0270	86	ADD M
355	0271	77	MOV M, A
356	0272	2C	INR L
357	0273	D2 77 02	JNC L3

PITDETECT. LST - 9

409	02C2	3A	73	11	LDA	LASTC
410	02C5	47			MOV	B, A
411	02C6	3A	79	11	LDA	CLIP
412	02C9	32	73	11	STA	LASTC
413	02CC	B8			CMP	B
414	02CD	DA	D1	02	JC	LL1
415	02D0	78			MOV	A, B
416	02D1	47			LL1: MOV	B, A
417	02D2	0F			RRC	
418	02D3	0F			RRC	
419	02D4	E6	3F		ANI	3FH ; 0.25 X LEVEL
420	02D6	47			MOV	B, A
421	02D7	0F			RRC	
422	02D8	E6	1F		ANI	1FH ; 0.125 X LEVEL
423	02DA	80			ADD	B
424	02DB	32	7A	11	STA	CLPLVL ; 0.25+0.125=0.
425	02DE	AF			XRA	A
426	02DF	32	79	11	STA	CLIP
427	02E2	CD	B5	01	CALL	SUBB
428	02E5	3A	7A	11	LDA	CLPLVL
429	02E8	FE	0A		CPI	CLPMIN
430	02EA	D2	F2	02	JNC	CLPCON
431	02ED	3E	0A		MVI	A, CLPMIN
432	02EF	32	7A	11	STA	CLPLVL
433	02F2	2F			CLPCON: CMA	
434	02F3	3C			INR	A
435	02F4	32	7B	11	STA	CLPLVL+1
436	02F7	6B			MOV	L, E
437	02F8	2D			DCR	L
438	02F9	7E			MOV	A, M
439	02FA	1E	01		MVI	E, 1
440	02FC	6B			MOV	L, E

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 9

SIMNT	LOC	OBJECT CODE	SOURCE STATEMENT
441	02FD	2D	DCR L
442	02FE	77	MOV M, A
443	02FF	3A 00 00	LDA STATUS
444	0302	0E 01	MVI C, 1
445	0304	C3 3E 00	JMP AGAIN
446	116E		ORG RAM+110
447	116E		CNT: DS 1
448	116F		A2DD: DS 1
449	1170		PITDV: DS 1
450	1171		PITV: DS 1
451	1172		PITM: DS 1
452	1173		LASTC: DS 1
453	1174		LTAD: DS 5
454	1179		LTAD1: DS 1
455	0074		TAD EQU LTAD-1100H
456	0079		TAD1 EQU LTAD1-1100H
457	1179		CLIP EQU LTAD1
458	117A		CLPLVL: DS 1
459			END

ASMCOMPSTR. LST - 1

***** PROGRAM FOR COMPRESS & STRECH ***

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 1

STMNT	LOC	OBJECT CODE	SOURCE STATEMENT
1			; COMPRESS STRECH PROGRAM
2			;
3			;
4			;
5	0000	3E 0C	MVI A, 0CH
6	0002		DB 30H ; SIM
7	0003	FB	WAIT: EI
8	0004	C3 03 00	JMP WAIT
9	002C		ORG 2CH
10	002C	C3 00 01	JMP 100H
11	0034		ORG 34H
12	0034	C3 00 02	JMP 200H
13	0100		ORG 100H
14	0100	31 F0 24	LXI SP, RAM56+0F0H
15	0103	3E 0E	MVI A, 0EH
16	0105	D3 20	OUT 20H
17	0107	3E FF	MVI A, 0FFH
18	0109	D3 23	OUT 23H
19	010B	21 80 46	LXI H, RAM+280H
20	010E	22 02 24	SHLD HPNT
21	0111	11 00 44	LXI D, RAM1
22	0114	21 00 44	LXI H, RAM1
23	0117	0E 02	MVI C, 2
24	0119	00	NOP
25	011A	00	NOP
26	011B	00	NOP
27	011C	00	NOP
28	011D	1A	LOOP: LDAX D
29	011E	96	SUB M
30	011F	D3 10	OUT MULO
31	0121	13	INX D
32	0122	00	NOP
33	0123	00	NOP
34	0124	DB 08	IN MULI
35	0126	86	ADD M
36	0127	23	INX H
37	0128	17	RAL
38	0129	C6 80	ADI 80H
39	012B	D3 22	OUT D2A
40	012D	E5	PUSH H
41	012E	2A 02 24	LHLD HPNT
42	0131	7C	MOV A, H
43	0132	E6 47	ANI MASK
44	0134	67	MOV H, A
45	0135	DB A1	LOOP2: IN A2D
46	0137	EE 7F	XRI 7FH
47	0139	F2 3D 01	JP SP4
48	013C	3F	CMC

ASMCOMPSTR. LST - 2

49	013D	1F			SP4:	RAR	
50	013E	77				MOV	M, A
51	013F	23				INX	H
52	0140	22	02	24		SHLD	HPNT
53	0143	E1				POP	H
54	0144	1A				LDAX	D
55	0145	96				SUB	M

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 2

STMNT	LOC	OBJECT CODE		SOURCE STATEMENT
56	0146	D3 10		OUT MULO
57	0148	13		INX D
58	0149	00		NOP
59	014A	00		NOP
60	014B	DB 08		IN MULI
61	014D	86		ADD M
62	014E	23		INX H
63	014F	17		RAL
64	0150	C6 80		ADI 80H
65	0152	D3 22		OUT D2A
66	0154	00		NOP
67	0155	0D		DCR C
68	0156	C2 1D 01		JNZ LOOP
69	0159	7A		MOV A, D
70	015A	E6 47		ANI MASK
71	015C	F6 44		ORI MASK1
72	015E	57		MOV D, A
73	015F	EB		XCHG
74	0160	DB 10		IN PITCH
75	0162	4F		MOV C, A
76	0163	7D		MOV A, L
77	0164	91		SUB C
78	0165	5F		MOV E, A
79	0166	54		MOV D, H
80	0167	D2 6B 01		JNC L
81	016A	15		DCR D
82	016B	7E		L: MOV A, M
83	016C	23		INX H
84	016D	13		INX D
85	016E	17		RAL
86	016F	C6 80		ADI 80H
87	0171	D3 22		OUT D2A
88	0173	E5		PUSH H
89	0174	2A 02 24		LHLD HPNT
90	0177	7C		MOV A, H
91	0178	E6 47		ANI MASK
92	017A	67		MOV H, A
93	017B	C3 35 01		JMP LOOP2
94	4400			RAM1 EQU 4400H
95	2402			HPNT EQU 2402H
96	2400			RAM56 EQU 2400H
97	00A1			A2D EQU 0A1H
98	0022			D2A EQU 22H
99	0008			MULI EQU 08H

ASMCMPSTR. LST - 3

100	0010				MULO	EQU	10H
101	0010				PITCH	EQU	10H
102	4400				RAM	EQU	4400H
103	0044				MASK1	EQU	44H
104	0047				MASK	EQU	47H
105	0200					ORG	200H
106	0200	31	F0	24		LXI	SP, RAM56+0F0H
107	0203	3E	0E			MVI	A, 0EH
108	0205	D3	20			OUT	20H
109	0207	3E	FF			MVI	A, 0FFH
110	0209	D3	23			OUT	23H

INTEL 8080 MICRO PROCESSOR ASSEMBLER PAGE 3

STMNT	LOC	OBJECT CODE	SOURCE STATEMENT
111	020B	21 00 44	LXI H, RAM
112	020E	E5	PUSH H
113	020F	21 80 46	LXI H, RAM1+280H
114	0212	11 80 46	LXI D, RAM1+280H
115	0215	0E 02	MVI C, 2
116	0217	00	NOP
117	0218	00	NOP
118	0219	00	NOP
119	021A	00	NOP
120	021B	1A	COOP: LDAX D
121	021C	96	SUB M
122	021D	D3 10	OUT MULO
123	021F	22 02 24	SHLD HPNT
124	0222	E1	POP H
125	0223	DB A1	IN A2D
126	0225	EE 7F	XRI 7FH
127	0227	F2 2B 02	JP SP1
128	022A	3F	CMC
129	022B	1F	SP1: RAR
130	022C	77	MOV M, A
131	022D	23	INX H
132	022E	E5	PUSH H
133	022F	2A 02 24	LHLD HPNT
134	0232	13	INX D
135	0233	DB 08	IN MULI
136	0235	86	ADD M
137	0236	23	INX H
138	0237	17	RAL
139	0238	C6 80	ADI 80H
140	023A	D3 22	OUT D2A
141	023C	22 02 24	COOP2: SHLD HPNT
142	023F	E1	POP H
143	0240	DB A1	IN A2D
144	0242	EE 7F	XRI 7FH
145	0244	F2 48 02	JP SP2
146	0247	3F	CMC
147	0248	1F	SP2: RAR
148	0249	77	MOV M, A
149	024A	23	INX H
150	024B	E5	PUSH H

ASMCMPSTR. LST - 4

151	024C	2A	02	24	LHLD	HPNT
152	024F	0D			DCR	C
153	0250	C2	1B	02	JNZ	COOP
154	0253	EB			XCHG	
155	0254	7C			MOV	A, H
156	0255	E6	47		ANI	MASK
157	0257	F6	44		ORI	MASK1
158	0259	67			MOV	H, A
159	025A	DB	10		IN	PITCH
160	025C	4F			MOV	C, A
161	025D	85			ADD	L
162	025E	5F			MOV	E, A
163	025F	3E	00		MVI	A, 0
164	0261	8C			ADC	H
165	0262	57			MOV	D, A

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE

4

STMNT	LOC	OBJECT CODE	SOURCE STATEMENT
166	0263	22 02 24	SHLD HPNT
167	0266	E1	POP H
168	0267	DB A1	IN A2D
169	0269	EE 7F	XRI 7FH
170	026B	F2 6F 02	JP SP3
171	026E	3F	CMC
172	026F	1F	RAR
173	0270	77	MOV M, A
174	0271	23	INX H
175	0272	7C	MOV A, H
176	0273	E6 47	ANI MASK
177	0275	67	MOV H, A
178	0276	E5	PUSH H
179	0277	2A 02 24	LHLD HPNT
180	027A	7E	MOV A, M
181	027B	23	INX H
182	027C	13	INX D
183	027D	17	RAL
184	027E	C6 80	ADI 80H
185	0280	D3 22	OUT D2A
186	0282	C3 3C 02	JMP COOP2
187			END

SYMBOL ADDRESS

PAGE

5

A2D	00A1
COOP	021B
COOP2	023C
D2A	0022
HPNT	2402
L	016B
LOOP	011D
LOOP2	0135
MASK	0047
MASK1	0044
MUL I	0008

ASMCVSD - 1

***** ASM CVSD-ADM PROGRAM ***

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 1

STMT	LOC	OBJECT CODE	SOURCE STATEMENT
1	0000	3E A4	MVI A, TIMEL
2	0002	32 04 00	STA 09004H
3	0005	3E C1	MVI A, TIMEH
4	0007	32 05 00	STA 09005H
5	000A	3E C2	MVI A, MODE
6	000C	32 00 00	STA 09000H
7	000F	0E 08	MVI C, DEL0
8	0011	59	MOV E, C
9	0012	16 00	MVI D, 0
10	0014	C3 00 01	JMP 100H
11	0100		ORG 100H
12	0100	3A 01 00	L00 LDA A2D *
13	0103	D6 01	SUI DEL08
14	0105	92	SUB D
15	0106	DA 27 01	JC L000
16	0109	7B	L001 MOV A, E
17	010A	07	RLC
18	010B	07	RLC
19	010C	07	RLC
20	010D	E6 08	ANI 8
21	010F	47	MOV B, A
22	0110	7B	MOV A, E
23	0111	90	SUB B
24	0112	5F	MOV E, A
25	0113	1F	RAR
26	0114	1F	RAR
27	0115	1F	RAR
28	0116	E6 1F	ANI 1FH
29	0118	47	MOV B, A
30	0119	7A	MOV A, D
31	011A	80	ADD B
32	011B	D2 20 01	JNC NOVR1
33	011E	3E FF	MVI A, MAX
34	0120	57	NOVR1 MOV D, A
35	0121	32 02 00	STA D2A *
36	0124	C3 8A 01	JMP L01
37	0127	7B	L000 MOV A, E
38	0128	07	RLC
39	0129	07	RLC
40	012A	07	RLC
41	012B	E6 08	ANI 8
42	012D	47	MOV B, A
43	012E	7B	MOV A, E
44	012F	90	SUB B
45	0130	81	ADD C
46	0131	1F	RAR
47	0132	1F	RAR
48	0133	1F	RAR

ASMCVSD - 2

49	0134	E6	1F			ANI	1FH
50	0136	47				MOV	B, A
51	0137	57				MOV	D, A
52	0138	90				SUB	B
53	0139	D2	3E	01		JNC	NOVR2
54	013C	3E	00			MVI	A, MIN
55	013E	57			NOVR2	MOV	D, A

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 2

STMT	LOC	OBJECT	CODE		SOURCE	STATEMENT	
56	013F	32	02	00		STA	D2A *
57	0142	C3	00	01		JMP	L00
58	0145	3A	01	00	L11	LDA	A2D *
59	0148	C6	01			ADI	DEL08
60	014A	92				SUB	D
61	014B	DA	6D	01		JC	L110
62	014E	7B			L111	MOV	A, E
63	014F	07				RLC	
64	0150	07				RLC	
65	0151	07				RLC	
66	0152	E6	08			ANI	8
67	0154	47				MOV	B, A
68	0155	7B				MOV	A, E
69	0156	90				SUB	B
70	0157	81				ADD	C
71	0158	5F				MOV	E, A
72	0159	1F				RAR	
73	015A	1F				RAR	
74	015B	1F				RAR	
75	015C	E6	1F			ANI	1FH
76	015E	47				MOV	B, A
77	015F	7A				MOV	A, D
78	0160	80				ADD	B
79	0161	D2	66	01		JNC	NOVR3
80	0164	3E	FF			MVI	A, MAX
81	0166	57			NOVR3	MOV	D, A
82	0167	32	02	00		STA	D2A *
83	016A	C3	45	01		JMP	L11
84	016D	7B			L110	MOV	A, E
85	016E	07				RLC	
86	016F	07				RLC	
87	0170	07				RLC	
88	0171	E6	08			ANI	8
89	0173	47				MOV	B, A
90	0174	7B				MOV	A, E
91	0175	90				SUB	B
92	0176	1F				RAR	
93	0177	1F				RAR	
94	0178	1F				RAR	
95	0179	E6	1F			ANI	1FH
96	017B	47				MOV	B, A
97	017C	57				MOV	D, A
98	017D	90				SUB	B
99	017E	D2	83	01		JNC	NOVR4

ASMCVSD - 3

100	0181	3E	00			MVI	A, MIN	
101	0183	57			NOVR4	MOV	D, A	
102	0184	32	02	00		STA	D2A	*
103	0187	C3	CC	01		JMP	L10	
104	018A	3A	01	00	L01	LDA	A2D	*
105	018D	92				SUB	D	
106	018E	DA	AF	01		JC	L010	
107	0191	7B			L011	MOV	A, E	
108	0192	07				RLC		
109	0193	07				RLC		
110	0194	07				RLC		

INTEL 8080 MICRO PROCESSOR ASSEMBLER PAGE 3

STMT	LOC	OBJECT CODE	SOURCE STATEMENT
111	0195	E6 08	ANI 8
112	0197	47	MOV B, A
113	0198	7B	MOV A, E
114	0199	90	SUB B
115	019A	5F	MOV E, A
116	019B	1F	RAR
117	019C	1F	RAR
118	019D	1F	RAR
119	019E	E6 1F	ANI 1FH
120	01A0	47	MOV B, A
121	01A1	7A	MOV A, D
122	01A2	80	ADD B
123	01A3	D2 AB 01	JNC NOVR5
124	01A6	3E FF	MVI A, MAX
125	01A8	57	MOV D, A
126	01A9	32 02 00	STA D2A *
127	01AC	C3 45 01	JMP L11
128	01AF	7B	MOV A, E
129	01B0	07	RLC
130	01B1	07	RLC
131	01B2	07	RLC
132	01B3	E6 08	ANI 8
133	01B5	47	MOV B, A
134	01B6	7B	MOV A, E
135	01B7	90	SUB B
136	01B8	1F	RAR
137	01B9	1F	RAR
138	01BA	1F	RAR
139	01BB	E6 1F	ANI 1FH
140	01BD	47	MOV B, A
141	01BE	57	MOV D, A
142	01BF	90	SUB B
143	01C0	D2 C5 01	JNC NOVR6
144	01C3	3E 00	MVI A, MIN
145	01C5	57	MOV D, A
146	01C6	32 02 00	STA D2A *
147	01C9	C3 CC 01	JMP L10
148	01CC	3A 01 00	LDA A2D *
149	01CF	92	SUB D
150	01D0	DA F1 01	JC L100

ASMCVSD - 4

151	01D3	7B		L101	MOV	A, E
152	01D4	07			RLC	
153	01D5	07			RLC	
154	01D6	07			RLC	
155	01D7	E6	08		ANI	B
156	01D9	47			MOV	B, A
157	01DA	7B			MOV	A, E
158	01DB	90			SUB	B
159	01DC	5F			MOV	E, A
160	01DD	1F			RAR	
161	01DE	1F			RAR	
162	01DF	1F			RAR	
163	01E0	E6	1F		ANI	1FH
164	01E2	47			MOV	B, A
165	01E3	7A			MOV	A, D

INTEL 8080 MICRO PROCESSOR ASSEMBLER

PAGE 4

STMNT	LOC	OBJECT CODE	SOURCE STATEMENT
166	01E4	80	ADD B
167	01E5	D2 EA 01	JNC NOVR7
168	01E8	3E FF	MVI A, MAX
169	01EA	57	NOVR7 MOV D, A
170	01EB	32 02 00	STA D2A *
171	01EE	C3 8A 01	JMP L01
172	01F1	7B	L100 MOV A, E
173	01F2	07	RLC
174	01F3	07	RLC
175	01F4	07	RLC
176	01F5	E6 08	ANI B
177	01F7	47	MOV B, A
178	01F8	7B	MOV A, E
179	01F9	90	SUB B
180	01FA	1F	RAR
181	01FB	1F	RAR
182	01FC	1F	RAR
183	01FD	E6 1F	ANI 1FH
184	01FF	47	MOV B, A
185	0200	57	MOV D, A
186	0201	90	SUB B
187	0202	D2 07 02	JNC NOVR8
188	0205	3E 00	MVI A, MIN
189	0207	57	NOVR8 MOV D, A
190	0208	32 02 00	STA D2A *
191	020B	C3 00 01	JMP L00
192	0001		A2D EQU 09001H
193	0002		D2A EQU 09002H
194	00A4		TIMEL EQU 0A4H
195	00C1		TIMEH EQU 0C1H
196	00C2		MODE EQU 0C2H
197	0008		DELO EQU B
198	0000		MIN EQU 0
199	00FF		MAX EQU 0FFH
200	0001		DELO8 EQU 1
201			END

נספח ג' : סכימות מפורטות של המערכת

נספח זה כולל סכימות אלקטרוניות מפורטות של חלקי המערכת. הנספח כולל חמש סכימות:

1. מיקרו מחשב גלאי PITCH
2. אוטוקורלטור
3. מיקרו מחשב כיווץ/מתיחה
4. יחידת החלון המלבני והמכפל
5. יחידת הזימון.

מקורות

1. D. Malah, "Time-domain algorithms for harmonic bandwidth reduction and time scaling of speech signals" IEEE Tran. Acoust., Speech Signal Processing, Vol. ASSP-27 No. 2, pp. 121-133, April 1979.
2. N.C. Seiler, R. Flowers, and J. Friedman, "A monolithic implementation of a CVSD algorithm".
3. R.W. Schafer and L.R. Rabiner, "Design and Simulation of a Speech Analysis-Synthesis System Based on Short-Time Fourier Analysis", I.E.E.E Trans. Audio Electroacoust., vol. AU-21, pp. 165-174, June 1973.
4. ד. מלאך, "פיתוח מערכת לצמצום ספרתי של אותות דיבור דו"ח מחקר לתקופה יוני-אוגוסט 1977" מוסד הטכניון למחקר ופיתוח.
5. L.R. Rabiner et al., "A comparative performance study of several pitch detection algorithms" IEEE Trans. Acoust, Speech, Signal Processing, vol. ASSP-24, pp. 399-418, Oct. 1976.
6. J.J. Dubonwsky, R.W. Schafer, and L.R. Rabiner, "Real-Time digital hardware pitch detectors" IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 399-418, Oct. 1976.
7. N.S. Jayant, "Digital coding of speech waveforms: PCM, DPCM and DM Quantizers" Proc. IEEE, vol. 62, pp. 611-632, May 1974.
8. D. Malah, "Time-domain algorithms for time-scale variations with speech signals", Technion, I.I.T. E.E. publ. 280, May 1976.

9. C.K. UN, D.T. Magill, "The residual excited linear Prediction Vocoder with transmission rate below 9.6 Kbit/sec", IEEE Trans. Comm. COM-23, pp. 1466-1474, Dec. 1975.
10. ד. מלאך, "פיתוח מערכת לצמצום ספרתי של אותות דיבור דו־יח מחקר לתקופה דצמבר 1977 - מרץ 1978", מוסד הטכניון למחקר ופיתוח.

The implementation is based on couple of micro computers that work simultaneously. One performs pitch detection based on center clipping autocorrelator and transfers the pitch values to the second processor. The second microprocessor performs the harmonic compression and ADM. The microprocessors are connected to some hardware systems including 2 bit autocorrelator, serial/parallel multiplier, analog to digital convertor and digital to analog convertor. We used couple of microprocessors and some hardware systems to be able to perform the algorithm in real time. The use of microprocessor decrease the amount of IC's and make the system much more changeable compared with hardware implementation. The word width of the used microprocessor is 8 bit. It is good for sampling and reconstruction, but in our implementation that need multiplication, addition and subtraction per a sample, it causes some decrease in speech quality. These days 16 bit microprocessor are new implementation with 16 bit word width will improve the system results.

A B S T R A C T

Digital communication systems are these days one of the main subjects in many researches and projects.

The rate of data transmission through telephone channels and radio is restricted to 3.2KHz due to the bandwidth of speech.

Nowadays digital communication systems mainly transfer alphanumeric data.

Transmission of speech simply by sampling it in 7.4KHz (2 x 3.2KHz) and sending 60,000 bits per second through the channels is impossible.

It is obvious that in that data rate the speech signals include more information than we need for understanding.

The goal of the project is a real time implementation of an algorithm that reduces the amount of data bits that goes through the channels.

This reducing is carried out by harmonic compression on the speech signals and then Adaptive Delta Modulation on the compressed signals.

The harmonic compression is carried out by time domain algorithm based on a development of Short Time Fourier Transform, exploiting the quasi periodic characteristics of voiced speech signals.

This algorithm needs pitch detection. Some algorithms for pitch detection and ADM were tested in mini computer simulations. These simulations help us to choose the algorithms and their parameters for the implementation.

This project was carried out under the supervision of Dr. David Malah in the Signal Processing Laboratory of the Electrical Engineering.

I am indebted to Dr. David Malah for his excellent guidance and devoted help throughout all the stages of the project, and also to his sympathetic attitude.

I am also grateful to Mr. Yarom Or Chen of the Signal Processing Laboratory for his help and assistance.

SYSTEM FOR HARMONIC COMPRESSION OF SPEECH SIGNAL
USING MICRO PROCESSOR

Project Description

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical Engineering

By

HELLERSTEIN RAN

Submitted to the Senate of the Technion - Israel Institute of Technology

July 1979

H a i f a

Tamuz 5379