

ספר קורס בנושא
תורת הצפינה והקודים

Error Correcting
& Detecting
Codes

ערכו וכתבו : אייל יחיא 027446335
צחי בובליל 025617838

תורת הצפינה והקודים - הקדמה

ספר זה מטפל בקודים לתיקון וגילוי שגיאות. כיום אין מערכת תקשורת ספרתית שאינה משלבת בה קודים אלו. שימוש בקודים מאפשר להגיע להסתברויות שגיאה נמוכות, להגדלת טווחי תקשורת, לחסכון בגודל אנטנה ובהספקי שידור. מגוון הקודים שבשימוש גדול מאוד כאשר הנפוץ ביותר הוא קוד הקונבולוציה הנמצא במיליארד טלפונים סלולאריים. בעתיד צפוי לגדול השימוש בקודי טורבו שהומצאו לפני כעשור והם הטובים ביותר.

ספר זה מיועד לעוסקים בתקשורת ספרתית עם דגש על ההיבטים המעשיים והמערכתיים.

מטרת הספר להקנות את הידע הבסיסי בנושאים הנלמדים בקורס תורת הצפינה והקודים. במבוא נתאר מושגים בסיסיים בקידוד כגון: יחס קידוד ושבח קידוד. בפרקים שני עד רביעי נתאר קודי בלוקים, תכונותיהם ותיאורם בעזרת מטריצות ופולינומים. בפרק החמישי נתאר קוד קונבולוציה ומפענח ויטרבי. פרק שש מוקדש לשוזר המלווה את פעולת הקודים בהרבה יישומים. הפרק השביעי מביא כדוגמא את התקן DVB-S. בפרק השמיני מוצגים מושגי יסוד של קוד טורבו.

ספר זה נכתב על פי הרצאותיו של ד"ר אריה רייכמן במכללת יהודה ושומרון באריאל בשנת הלימודים תשס"ג.

לקורס תורת הצפינה והקודים מומלץ להיעזר בספר Digital Communications שנכתב על ידי SKLAR בהוצאת Prentice Hall מהדורה שנייה 2001.

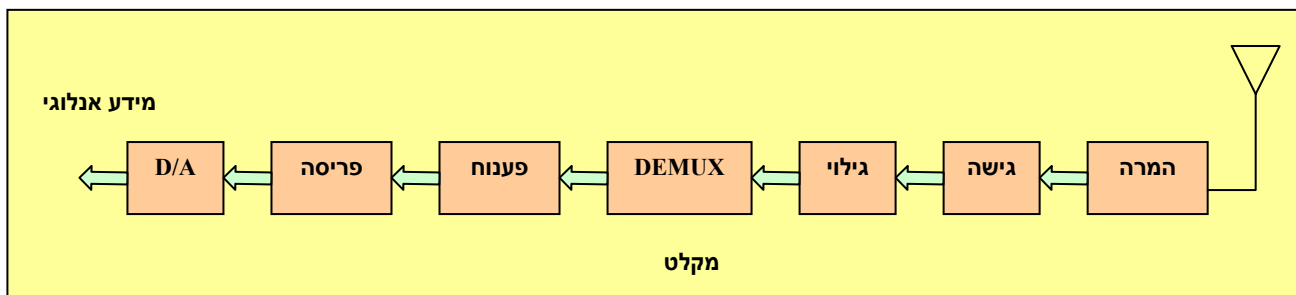
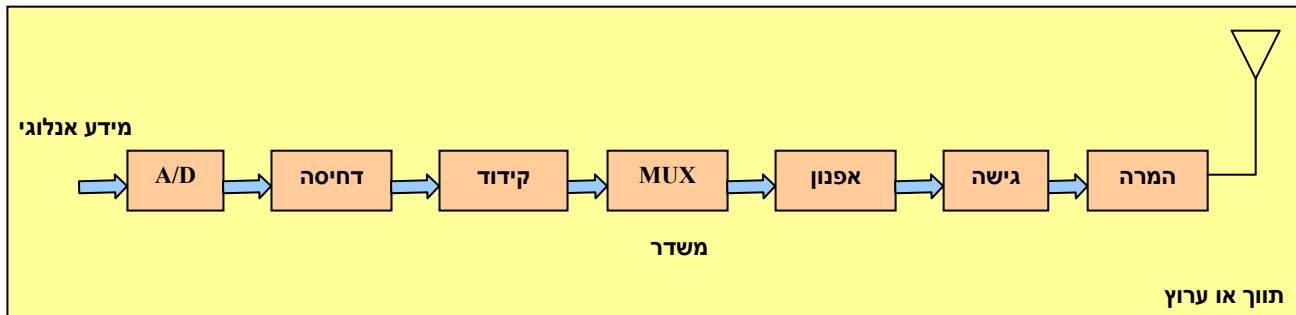
תוכן עניינים:

1	מבוא	3
1.1	מערכת תקשורת ספרתית	3
1.2	מונחים בסיסיים	3
1.3	יחס קידוד R_c – CODING RATE	4
1.4	סיסטמטיות	5
1.5	הסתברות שגיאה	5
1.6	יחס אות לרעש במערכות ספרתיות	7
1.7	ביצועי קוד	10
1.8	סוגי החלטות גלאי	12
1.9	דוגמא להשפעת הקידוד על רוחב הסרט וההספק	16
1.10	שאלות סיכום פרק	17
2	קודי בלוקים BLOCK CODES – מושגים ודוגמאות	19
2.1	מרחק הקוד	19
2.2	מפענח שלם COMPLETE DECODER	21
2.3	יכולת תיקון שגיאות	23
2.4	מפענח לא שלם – INCOMPLETE DECODER	26
2.5	יכולת תיקון וגילוי שגיאות	26
2.6	יכולת תיקון וגילוי שגיאות בערוץ עם מחיקות	29
2.7	סיכום המשפטים וניתוחם:	30
2.8	ביצועים של קודי בלוקים עם החלטות קשות	32
2.9	קודי חזרות - REPETITION - m	36
2.10	קודי בלוקים - קוד BCH (N,K,T)	37
2.11	שינויים לקוד בלוקים נתון	39
2.12	קוד REED - SOLOMON	45
2.13	קוד GOLAY-קוד מושלם	50
2.14	שאלות סיכום פרק	53
3	קודי בלוקים – קידוד ופענוח קודים ליניאריים בעזרת מטריצות	56
3.1	הגדרות	56
3.2	יצירת קוד בעזרת מטריצה יוצרת	57
3.3	מטריצה בודקת -H – PARITY CHECK MATRIX	58
3.4	קוד המינג- HAMMING	61
3.5	פענוח קודים ליניאריים	63
3.6	הרחבה של קוד המינג	66
3.7	שאלות סיכום פרק	67

69	CYCLIC CODE	קודי בלוקים - קודים ציקליים	4
69	ציקליות		4.1
71	שיטת קידוד ציקלי ראשונה (לא סיסטמטית):		4.2
72	שיטת קידוד ציקלי שניה (סיסטמטית)		4.3
75	סיכום שתי השיטות		4.4
76	מימוש של מקודד ומפענח של קוד ציקלי		4.5
81	ביצועים של קוד לגילוי שגיאות		4.6
84	שאלות סיכום פרק		4.7
85	קודי קונבולוציה		5
85	קידוד ביחס $\frac{1}{n}$		5.1
87	תיאור הקוד בעזרת דיאגרמת מצבים		5.2
89	תיאור הקוד בעזרת טרליס TRELLIS		5.3
91	פענוח		5.4
91	פענוח בעזרת אלגוריתם ויטרבי		5.5
95	פענוח עם החלטות רכות		5.6
103	ניקוב - PUNCTURING		5.7
108	שיקולים לבחירת קוד		5.8
109	ביצועי קוד קונבולוציה		5.9
116	שאלות סיכום פרק		5.10
118	שוזר - INTERLEAVER		6
120	שוזר בלוקים:		6.1
121	השהיה הנגרמת על ידי שוזר/משחזר		6.2
122	טיפול בפרץ שגיאות		6.3
122	שוזר קונבולוציה		6.4
125	שאלות סיכום פרק		6.5
126	DVB - S		7
127	קידוד חיצוני		7.1
128	קידוד פנימי		7.2
130	קודי טורבו		8
130	מבוא לקודי טורבו - ITERATIVE DECODING - TURBO		8.1
131	מבנה של מקודד		8.2
134	הסבר פענוח איטרטיבי של קוד מקבילי:		8.3

1 מבוא

1.1 מערכת תקשורת ספרתית



איור 1.1 דיאגרמת בלוקים של מערכת תקשורת

קורס זה מטפל במשבצות המקודד והמפענח. במערכת תקשורת ספרתית המרכיבים החיוניים הם האפנון והגלאי, אך מערכת תקשורת ספרתית מודרנית כוללת קודים לתיקון שגיאות. מטרת המקודד והמפענח היא להתמודד עם רעשים והפרעות בערוץ.

1.2 מונחים בסיסיים

מטרתם לשפר את ביצועי המערכת. מבחינת הסתברות השגיאה. הוספתם מצריכה לפעמים משאבים של זמן ורוחב סרט, אך השקעה זו לרוב כדאית.

ECC – Error Correcting Code – מונח כללי לקודים לתיקון שגיאות.

FEC – Forward Error Correcting Code – קודים לתיקון שגיאות ללא משוב בין המקלט למשדר.

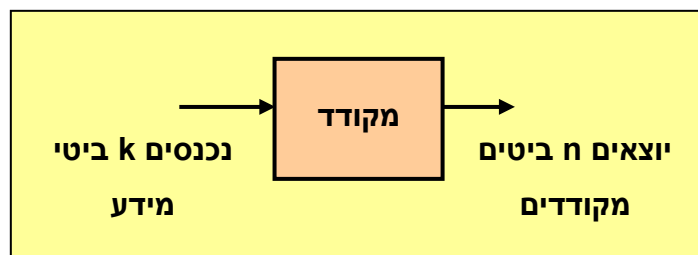
CRC – Cyclic Redundancy Code – שיטת קידוד המשמשת בעיקר לגילוי שגיאות.

ARQ – Automatic Repeat Request – מנגנון לתיקון שגיאות בעזרת בקשת שידור חוזר בערוץ חוזר בין המקלט למשדר.

FEC לעומת ARQ

נדגיש את ההבדל העיקרי בין FEC למנגנון ARQ. FEC משדר והמקלט מתקן מה שאפשר, ואילו בשיטת ARQ המקלט מוודא שההודעה הנקלטת נכונה ואם לא דורשת שידור חוזר.

1.3 יחס קידוד - Coding Rate R_c



איור 1.2 דיאגרמת מקודד k ל n

$$R_c = \frac{k}{n}$$

$n > k$ ולכן יחס הקידוד $0 < R_c < 1$

אם R_c קרוב לאחד אזי יחס הקידוד גבוה.

אם R_c קרוב לאפס אזי יחס הקידוד נמוך.

כאשר יחס הקידוד גבוה אזי מוסיפים מעט יתירות.

כאשר יחס הקידוד נמוך אזי מוסיפים הרבה יתירות.

אם קצב המידע הינו R_b אזי לאחר הקידוד המידע המקודד יהיה בקצב R_{eb} כאשר:

$$R_{eb} = \frac{R_b}{R_c}$$

כאשר R_b קבוע, הקידוד מגדיל את הקצב וכן את רוחב הסרט. הקטנת R_c מגדילה עוד יותר את R_{eb} . ישנן מערכות שבהן קצב המידע המקודד קבוע, במקרה זה הוספת קוד לתיקון שגיאות מקטינה את קצב המידע R_b .

1.4 סיסטמטיות

1.4.1 קוד סיסטמטי

קוד שבו n ביטי היציאה כוללים את k ביטי הכניסה ו $n - k$ ביטי יתירות.

1. k ביטי כניסה
2. $n - k$ ביטי יתירות – Redundancy

1.4.2 קוד לא סיסטמטי

n ביטי היציאה לא כוללים את כל k ביטי הכניסה.

- בדרך כלל : קודי הבלוקים סיסטמטיים
קודי הקונבולוציה לא סיסטמטיים
קודי הטורבו סיסטמטיים.

- הקוד לתיקון שגיאות בא בעיקר לטפל בבעיות רעש המתרחשות בערוץ.
באופן מעשי לא ניתן לתקן את כל השגיאות בערוץ.

1.5 הסתברות שגיאה

1.5.1 הסתברות שגיאה לביט P_b

הסתברות שגיאה לביט P_b המכונה גם $BER - Bit Error Rate$ מוגדרת במערכת מקודד כהסתברות לביט שגוי לאחר הפענוח.

דוגמאות של הסתברויות שגיאה נדרשות :

$10^{-2} - 10^{-3}$	דיבור סיפרתי (סלולארי) :
10^{-5}	ספרתי כללי (מדד מאוד מקובל)
$10^{-6} - 10^{-8}$	מידע ספרתי רגיש (אינטרנט)
$10^{-10} - 10^{-12}$	QEF – Quasi Error Free

1.5.2 הסתברות הודעה שגויה P_m

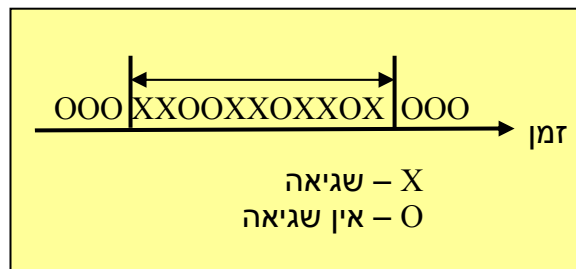
הסתברות שגיאה להודעה המכונה גם $MER - Message Error Rate$, מוגדרת כהסתברות לשגיאה בהודעה לאחר פענוח הקוד.
במקרים מסוימים שימוש ב- MER עדיף כמדד לביצועים.

1.5.3 אופי השגיאות

בדרך כלל מניחים שהשגיאות אקראיות ובלתי תלויות, ונגרמות על ידי רעש לבן בערוץ. לעומת זאת ישנם מקרים בהם השגיאות מופיעות בפרצים (Burst). זה קורה כאשר יש בערוץ רעש פרצי. מקרה נוסף שבו השגיאות מופיעות בפרצים זה ביציאת מפענח קוד קונבולוציה כפי שנלמד בהמשך.

הגדרה של אורך פרץ שגיאות:

שגיאות המופיעות במרוכז. מספר הביטים בין הביט השגוי הראשון והאחרון. בדוגמא הנתונה אורך הפרץ 11 ומתוכו 7 ביטים שגויים.



איור 1.3 פרץ שגיאה באורך 11

דוגמא 1.1:

$$P_b = p = 10^{-5}$$

ישנה הודעה באורך $m = 10^3$, מה ההסתברות לשגיאה בהודעה MER?

ההסתברות שההודעה תהיה נכונה: $(1 - p)^m$

אזי ההסתברות שההודעה תהיה שגויה: $MER = 1 - (1 - p)^m$

עבור $p \cdot m \ll 1$ ניתן לבצע את הקירוב הבא:

$$MER = 1 - (1 - p)^m \approx 1 - (1 - p \cdot m) = m \cdot p = 10^{-2}$$

דוגמא 1.2:

נציג שני מקרים בהם הסתברות השגיאה זהה. במקרה אחד השגיאות אינן תלויות והסתברות השגיאה להודעה מוגדרת כ-MER1 וניתנת לחישוב כפי שמתואר בדוגמא 1.1. ואילו במקרה השני השגיאות באות בפרצים ופוגעות בפחות הודעות וקטנות יותר מאשר במקרה הראשון. הסתברות השגיאה להודעה מוגדרת כ-MER2.

$$MER1 > MER2$$

1.6 יחס אות לרעש במערכות ספרתיות

המדד למדידת יחס אות לרעש במערכות ספרתיות הוא:

$$\frac{E_b}{N_0}$$

E_b – אנרגיה של ביט מידע
 N_0 – צפיפות הספק הרעש

חשיבות המדד במערכות ספרתיות נובעת מהעובדה שהשידור הספרתי לפעמים בעל אופי פרצי, לכן אין אפשרות לדבר על הספק. למרות זאת ניתן להגדיר יחס אות לרעש SNR = Signal To Noise Ratio כ:

$$SNR = \frac{S}{N}$$

S – הספק אות
 N – הספק רעש

עבור רעש אחיד בתדר בצפיפות הספק N_0 ורוחב סרט B :

$$SNR = \frac{S}{N_0 \cdot B}$$

נציב בביטוי את: $\frac{E_b}{N_0}$

$$E_b = S \cdot T_b = \frac{S}{R_b}$$

T_b – משך זמן של ביט מידע

כאשר R_b (קצב הביטים) קשור ל- T_b (משך הביט) על ידי:

$$R_b = \frac{1}{T_b}$$

נקבל:

$$\frac{E_b}{N_0} = \frac{S \cdot T_b}{N_0} = \frac{S}{N_0 \cdot R_b}$$

אם רוחב הסרט בביטוי של SNR שווה לקצב הביטים אזי ניתן לומר ש:

יחס אות
לרעש ברוחב
סרט R_b

$$SNR = \frac{E_b}{N_0}$$

על כן:

- $\frac{E_b}{N_0}$ הוא יחס אות לרעש כאשר הרעש נמדד ברוחב סרט ששווה לקצב הביטים: $B = R_b$.

רוחב הסרט האמיתי של מערכת על פי הגדרה כלשהי יכול להיות גדול, קטן או שווה ל- R_b

- $\frac{E_b}{N_0}$ זהו מספר טהור וניתן למדוד אותו ב- dB כלומר:

$$\left(\frac{E_b}{N_0} \right)_{dB} = 10 \log \left(\frac{E_b}{N_0} \right)$$

ל- $\frac{E_b}{N_0}$ יש שם חיבה: "**אבנו** - **Ebno**" השגור בעיקר בארה"ב.

במקלט ניתן להגדיר שני סוגי $\frac{E_b}{N_0}$ והם:

הנמדד במקלט או הנקלט או שמגיע למקלט $\leq$ רצוי שיהיה **גבוה**.

הנדרש במקלט על סף הקליטה כדי להגיע לביצועים מסוימים $\leq$ רצוי שיהיה **נמוך**.

ההפרש בין שני גדלים אלו נקרא **שוליים – Margin**

$$\text{Margin} = \left(\frac{E_b}{N_0} \right)_{rec} - \left(\frac{E_b}{N_0} \right)_{th}$$

השוליים נועדו להתמודד בדעיכות או בשינויים בפרמטרים של המערכת.

לפעולה תקינה של המערכת נדרש:

$$\frac{E_b}{N_0} > \frac{E_b}{N_0}$$

נקלט נדרש

אנו נגדיר את $\frac{E_b}{N_0}$ הנדרש בצורה הבאה:

$$\left(\frac{E_b}{N_0} \right)_{th}$$

כאשר הסימן th בא מהמילה Threshold שמשמעותה סף. כלומר מתחת לזה לא ניתן להשיג ביצועים נדרשים.

מספרים טיפוסיים ל $\left(\frac{E_b}{N_0} \right)_{th}$ הם:

בשנות השבעים: $10 - 15dB$ - ללא קידוד

בשנות השמונים: $5 - 10dB$ - עם קידוד

בשנות התשעים: $3 - 5dB$ - עם קידוד משורשר

כיום המגמה לקראת: $0dB$ - עם קידוד טורבו

1.7 ביצועי קוד

כאשר משווים בין קודים שונים יש להתייחס למספר פרמטרים ביניהם:

- $\frac{E_b}{N_0}$ הסתברות השגיאה בתלות
- השפעה על רוחב הסרט
- סיבוכיות מימוש

נדון בהשלכות.

1.7.1 הבדלים בין לפני קידוד ואחרי קידוד:

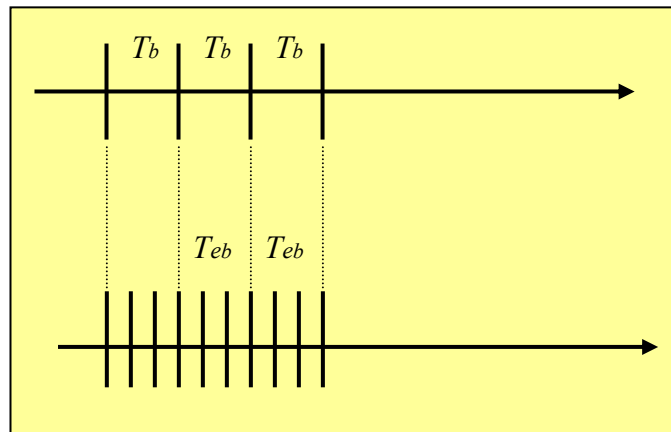
$$R_b = \frac{1}{T_b}$$

T_b - זמן של ביט

$$E_b = S \cdot T_b$$

E_b - אנרגיה של ביט מידע.

S - הספק



איור 1.4 זמן ביט מידע ומידע מקודד עבור יחס קידוד $R_c = \frac{1}{3}$

כאשר המידע מקודד $T_{eb} = T_b \frac{k}{n}$, $E_{eb} = E_b \frac{k}{n}$, $R_{eb} = R_b \frac{n}{k}$

1.7.2 השפעת הקידוד על הקצב והאנרגיה של ביטים

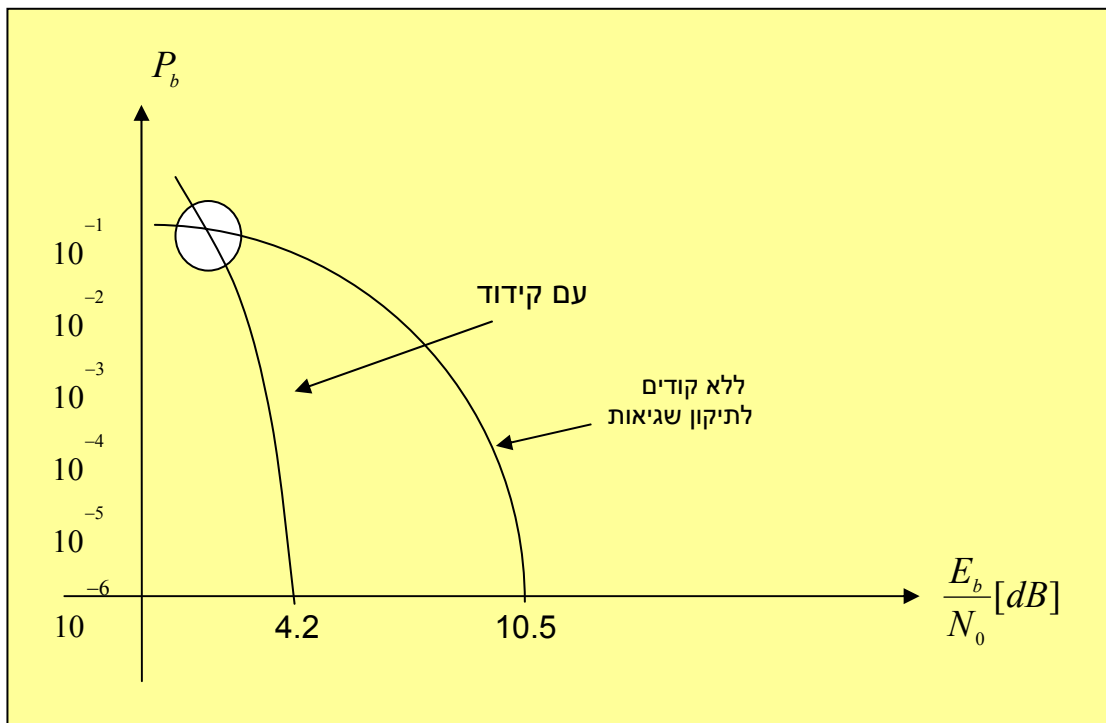
R_b - קצב המידע

קצב ביטים מקודדים

$$R_{eb} = \frac{R_b}{R_c}$$

יחס קידוד קטן אזי קצב ביטים מקודדים גדל.

1.7.3 שבח קידוד (CG)



איור 1.5 עקום אופייני של הסתברות שגיאה בתלות של $\frac{E_b}{N_0}$ ב dB

הסתברות השגיאה P_b בתלות ב $\frac{E_b}{N_0}$ בכניסה למקלט כמתואר באיור 1.5 .

שבח קידוד מוגדר כהפרש ב dB בין $\frac{E_b}{N_0}$ שנדרש לקבלת P_b מסוים ללא קידוד לתיקון

שגיאות ובין $\frac{E_b}{N_0}$ הנדרש לקבל אותו P_b עם קידוד. שבח הקידוד מתאפס ואפילו נעשה

שלילי עבור הסתברויות גדולות. ההסתברות הכי גרועה לשגיאה היא $\frac{1}{2}$.

שבח קידוד תלוי בגורמים הבאים:

- הקוד (כולל יחס קידוד).
- האפנון (BFSK, QPSK וכו').
- ההחלטות שהגלאי מעביר למפענח - קשות או רכות.
- הסתברות השגיאה המוגדרת P_b .
- אופיין הרעש בערוץ. אנו נתייחס לרעש לבן שגורם לשגיאות אקראיות.

דוגמא 1.3:

על פי איור 1.5

עבור אפנון QPSK קוד קונבולוציה ביחס של $\frac{1}{3}$ החלטות רכות.

עבור $P_b = 10^{-5}$:

$$CG = 9.6 - 4 = 5.6dB$$

עבור $P_b = 10^{-6}$:

$$CG = 10.5 - 4.2 = 6.3dB$$

1.8 סוגי החלטות גלאי

סוג החלטות שהגלאי (Demodulator) מעביר למפענח (Decoder).

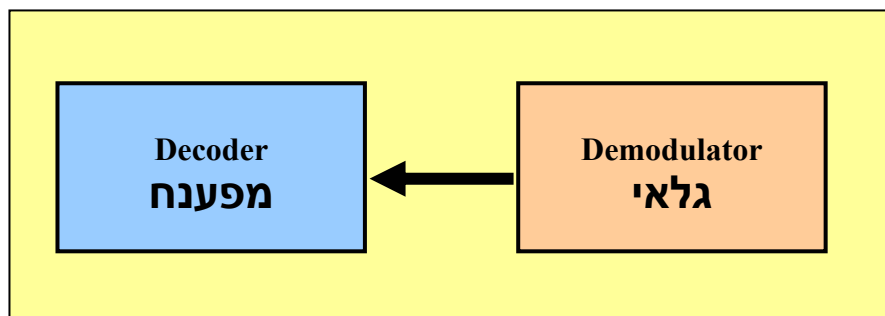
החלטה קשה Hard Decision

החלטה רכה Soft Decision

החלטה קשה - כלומר מחליט בין 0 ל-1. קוראים לזה גם ביט אחד של החלטה.

החלטה רכה - החלטה קשה ובנוסף גם תו איכות על ידי תוספת ביטים. בדרך כלל 3,4 ביטים סה"כ, כלומר $1 + HD + 2,3$ תווי איכות.

בהחלטה רכה יש יותר מידע למפענח ולכן הביצועים ב HD פחות טובים מ SD בדרך כלל בסדר גודל של $2dB$.



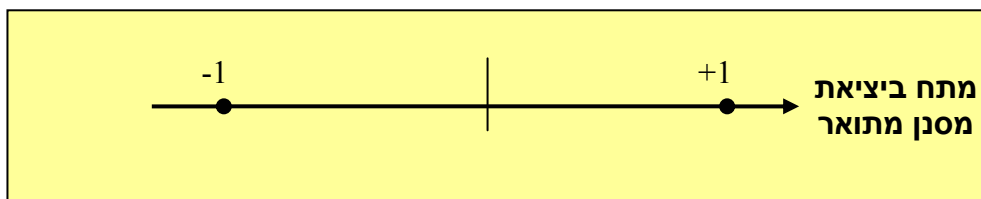
איור 1.6 גלאי ומפענח

סוג נוסף של החלטה היא מחיקה (Erasure). זהו מצב של הימנעות מהחלטה, כלומר הגלוי לא יכול להחליט האם האות הנקלט הוא "0" או "1".

איורים 1.7 מתארים החלטה על פי רמת המתח של המסנן המתואם בגלוי של אפנון בינארי אנטיפודאלי:

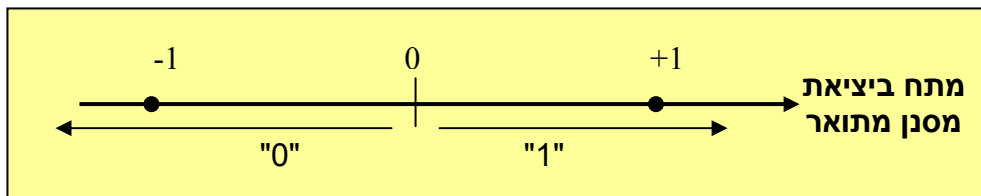
- איור א' - מתח המסנן המתואם
- איור ב' - החלטה קשה.
- איור ג' - החלטה רכה של 3 ביטים.
- איור ד' - החלטה קשה בתוספת מחיקה.
- איור ה' - החלטה רכה של 3 ביטים בתוספת מחיקה.

נניח ערוץ בינארי. במקלט הגלוי דוגם מסנן מתואם וללא רעש מתקבל המתח $+1$ עבור "1" לוגי ו -1 עבור "0" לוגי. עקב הרעש בערוץ יתכנו מתחים נוספים. נסמן בציר האופקי את המתח ביציאת המסנן.



איור 1.7 א

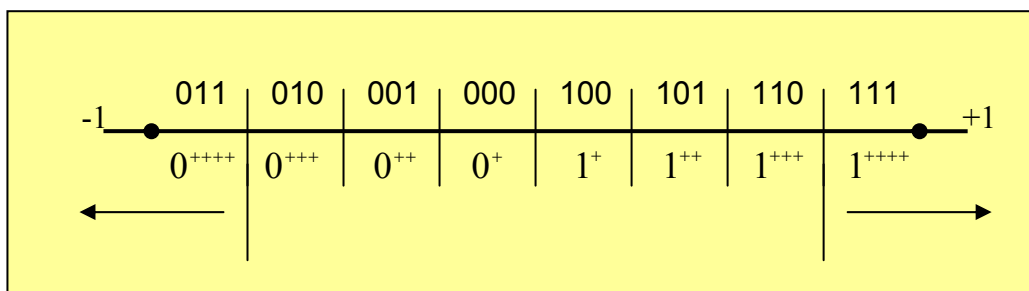
אם החלטה קשה מחליטים על מתח חיובי כ "1" ומתח שלילי כ "0".



איור 1.7 ב

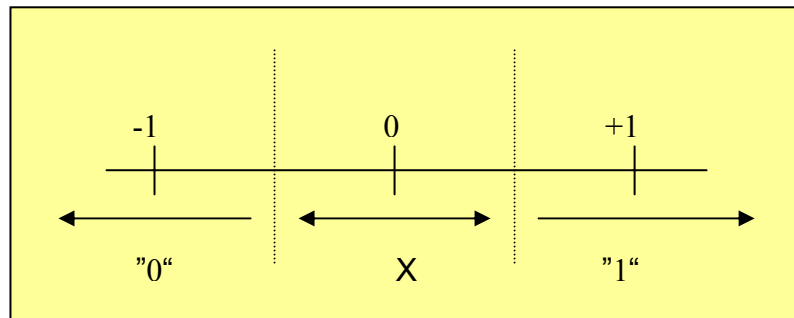
לפי הדוגמא שלפנינו החלטה רכה מעשית מתקבלת ע"י 3 ביטים כאשר:

- 7 ספים (threshold)
- 8 אזורי החלטה
- 4 רמות איכות



איור 1.7 ג

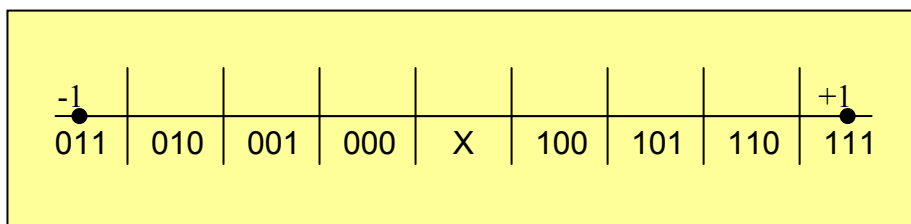
ערוץ עם מחיקות והחלטה קשה נקרא בסלנג ביט וחצי. אזורי ההחלטה יהיו כמתואר, כאשר X מסמן מחיקה.



איור 1.7 ד

כאשר יש שני ספים ושלושה אזורי החלטה.

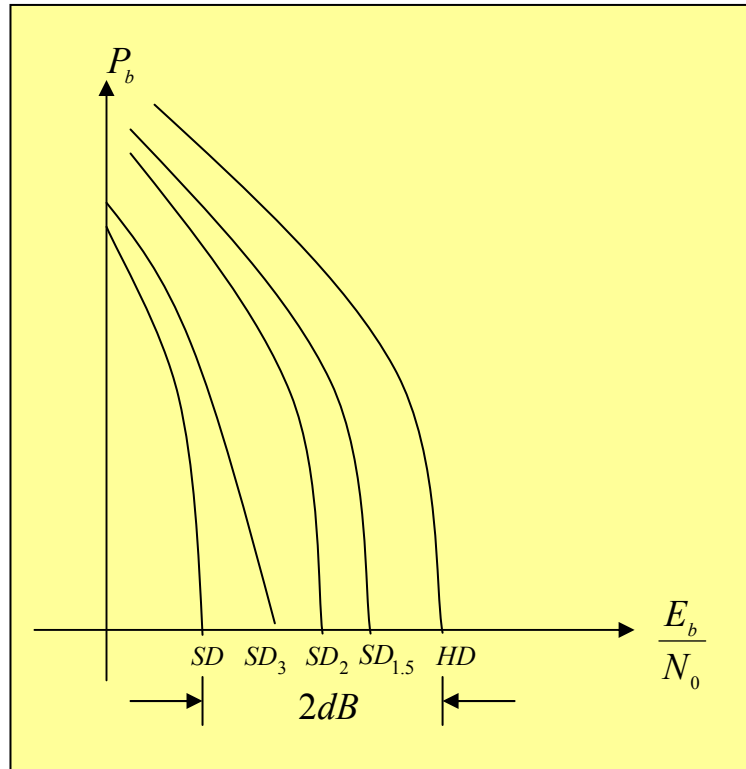
ערוץ עם מחיקות והחלטה רכה או בסלנג 3 וחצי ביטים.



איור 1.7 ה

ניתן להוסיף מחיקות גם בשני הקצוות כי כאשר יש רעש חזק הרעש יהיה למשל 3 או 4 ולכן יש צורך למחוק אותו ולכן נשים X בקצוות.

הביצועים ישתפרו עם הגדלת מספר הביטים של החלטה רכה. בציור מתואר באופן איכותי השיפור בין החלטה קשה HD לבין החלטה רכה SD.



איור 1.8 השפעת אופי ההחלטה על הביצועים

$SD_{1.5}$ מסמן ערוץ מחיקות, SD_2 ו SD_3 ערוצים עם החלטות רכות של 2 ו 3 ביטים בהתאמה ו SD החלטה רכה אידיאלית שבו המתח של המסנן המתואר מועבר למפענח ללא חלוקה לרמות (∞ ביטים). SD_3 מתקרב ל SD האידיאלי והמקובל בתעשייה.

1.9 דוגמא להשפעת הקידוד על רוחב הסרט וההספק

שבח קידוד של קוד קונבולציה ביחס $R_c = \frac{1}{4}$ הוא $6dB$ עבור אפנון QPSK והסתברות שגיאה 10^{-5} . במערכת QPSK עם קצב מידע R_b ורוחב סרט B והספק שידור S , ללא קידוד מתקבלת הסתברות שגיאה של $P_b = 10^{-5}$.

א. איך ישתנה רוחב הסרט וההספק הנדרשים עבור אותו קצב מידע?
ב. אם רוצים להישאר באותו רוחב סרט B מה יהיה קצב המידע וההספק הנדרש?

מלא את הטבלה: (הנתונים מופיעים באדום)

קידוד	קצב המידע	רוחב הסרט	הספק
ללא קידוד	R	B	S
עם קידוד	R	4B	$S - 6dB \left(\frac{S}{4} \right)$
עם קידוד	$\frac{R}{4}$	B	$S - 12dB \left(\frac{S}{16} \right)$

$$R_{eb} = \frac{R_b}{R_c} = \frac{9.6k}{\frac{1}{2}} = \frac{4.8k}{\frac{1}{4}} = \frac{2.4k}{\frac{1}{8}} = \frac{1.2k}{\frac{1}{16}} = 19.2k$$

$$\text{הנדרש ללא קידוד} \quad \left(\frac{E_b}{N_0} \right)_{Code} = \left(\frac{E_b}{N_0} \right)_{NoCode}$$

$$\text{הנדרש עם קידוד} \quad \left(\frac{E_b}{N_0} \right)_{Code} = \left(\frac{E_b}{N_0} \right)_{NoCode} - CG$$

$$\frac{E_b}{N_0} = \frac{S \cdot T_b}{N_0} = \frac{S}{R_b \cdot N_0} \Rightarrow \frac{S_{Code}}{R_{b_Code} \cdot N_0} = \frac{S_{No_Code}}{R_{b_No_Code} \cdot N_0} \cdot \frac{1}{CG}$$

נראה איך הנוסחה מתבטאת בטבלה בשורה השנייה והשלישית:

$$\frac{S_{Code}}{R \cdot N_0} = \frac{S_{No_Code}}{R \cdot N_0} \cdot \frac{1}{4} \Rightarrow S_{Code} = \frac{S_{No_Code}}{4}$$

$$\frac{S_{Code}}{\frac{R}{4} \cdot N_0} = \frac{S_{No_Code}}{R \cdot N_0} \cdot \frac{1}{4} \Rightarrow S_{Code} = \frac{S_{No_Code}}{16}$$

1.10 שאלות סיכום פרק

1. הסתברות שגיאה לביט היא P . בהנחת חוסר תלות בין שגיאות מה הסתברות נפילת הודעה P_M של M ביטים? תן קירוב עבור P קטן. השווה את הקירוב לחישוב המדויק עבור ערכים שונים של M ו P .

P	M	P_M מדויק	P_M קירוב
10^{-5}	100		
10^{-5}	1000		
10^{-4}	100		
10^{-4}	1000		
10^{-3}	100		
10^{-3}	1000		

2. שבח קידוד של קוד קונבולוציה ביחס קידוד $\frac{1}{2}$ היא 5dB עבור QPSK והסתברות שגיאה 10^{-5} . במערכת עם קצב מידע R, רוחב סרט B והספק שדור S ללא קודים לתיקון שגיאות הוחלט להכניס את הקוד הנ"ל.

2.1. איך ישתנה רוחב הסרט והספק הנדרשים עבור אותו קצב מידע?

2.2. אם רוצים להישאר באותו רוחב סרט B מה יהיה קצב המידע והספק הנדרש?

מלא את הטבלה.

קידוד	קצב מידע	רוחב סרט	הספק
ללא קוד לתיקון שגיאות	R	B	S
עם הקוד לתיקון שגיאות מקרה 2.1	R		
עם הקוד לתיקון שגיאות מקרה 2.2		B	

3. נתון משדר עם מקודד $RS_8(240,192)$ ואפנון BFSK.

3.1. מה הקוד המקורי ממנו נגזר קוד זה?

3.2. שרטט סכמת מלבנים עקרונית של המשדר.

3.3. מהו יחס הקידוד R_c ? במערכת עם קצב מידע R_b , מה קצב מידע המקודד R_{eb} ?

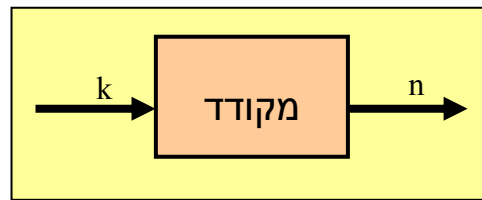
3.4. חשב הסתברות שגיאה למילה.

3.5. חשב הסתברות שגיאה לביט.

3.6. נניח ששבח קידוד של הקוד חושב והוא 2.5dB עבור הסתברות שגיאה מסוימת. במערכת עם קצב מידע R, רוחב סרט B והספק שדור S ללא קודים לתיקון שגיאות הוחלט להכניס את הקוד הנ"ל. איך ישתנה רוחב הסרט והספק הנדרשים עבור אותו קצב מידע וארתה הסתברות שגיאה? אם רוצים להישאר באותו רוחב סרט B מה יהיה קצב המידע והספק הנדרש? רשום את התשובות בטבלה.

קידוד	קצב מידע	רוחב סרט	הספק
ללא קוד לתיקון שגיאות	R	B	S
עם הקוד לתיקון שגיאות	R		
עם הקוד לתיקון שגיאות		B	

2 קודי בלוקים Block Codes – מושגים ודוגמאות



$$n > k$$

איור 2.1 מקודד בלוקים k ל n

n הביטים המהווים מילת קוד Code Word .
יש 2^k מילות קוד (מילים חוקיות) מתוך 2^n מילות קוד אפשריות.

בקוד סיסטמטי:

$$n = \begin{cases} k & \text{במקור} \\ n - k & \text{יתירות} \end{cases}$$

קודים מסוג המינג, ציקלים, BCH, Golay, חזרות.
בפרק זה אנו נדון בשיטות גילוי אשר מבוססות על החלטה קשה בלבד.

2.1 מרחק הקוד

אחד הפרמטרים החשובים בקודי בלוקים הוא מרחק הקוד d .
מרחק המינג בין שתי מילות קוד הינו מספר הביטים ששונים (בהתאמה) בין שתי מילות הקוד.

אם $\underline{a}$ ו $\underline{b}$ הן שתי מילות קוד:

$$\underline{a} = a_1 a_2 a_3 a_4 \dots a_n$$

$$\underline{b} = b_1 b_2 b_3 b_4 \dots b_n$$

אזי המרחק מחושב על ידי פעולת XOR ולכן :

$$\text{מרחק המינג בין } \underline{a} \text{ ו } \underline{b} = \sum_{i=1}^n a_i \oplus b_i = d(\underline{a}, \underline{b})$$

כאשר פעולת ה- XOR מוגדרת:

הערה	$a \oplus b$	b	a
זזה	0	0	0
שונה	1	1	0
שונה	1	0	1
זזה	0	1	1

טבלה 2.1 פעולת XOR

דוגמא 2.1 :

$$\begin{aligned}\underline{a} &= 00001111 \\ \underline{b} &= 00111100 \\ \underline{a} \oplus \underline{b} &= \overline{00110011}\end{aligned}$$

המרחק המינימלי בין 4 כלומר ישנם ארבעה מקומות שונים בין שתי המילים.

מרחק של קוד – המרחק המינימלי בין כל שתי מילות קוד $d = \min d(\underline{a}, \underline{b})$

2.2 מפענח שלם complete decoder

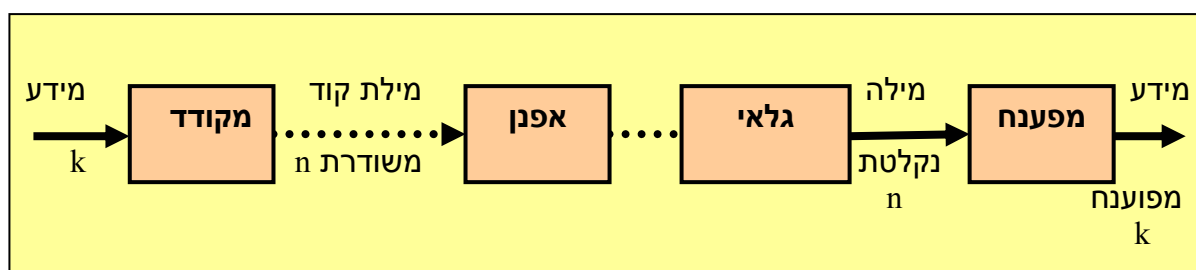
עבור קודי בלוקים וגלאי עם החלטות קשות המפענח השלם מוגדר על פי התהליך כדלקמן:
המפענח מקבל מילה נקלטת אחת. נמדד מרחק המינג בין המילה הנקלטת ובין כל מילות הקוד. המילה המפוענחת היא בעלת המרחק הקטן ביותר מבין מילות הקוד למילה הנקלטת.

נתאר את התהליך בעזרת טבלה. שלוש העמודות הראשונות משמאל הינן קבועות עבור קוד מסוים והן מהוות LUT – look up table.
שתי העמודות האחרות הן עבור קליטה של מילה מסוימת. בעמודה הרביעית נכניס את המילה הנקלטת b לכל אורך העמודה. בעמודה החמישית נכניס את המרחק בין המילה הנקלטת ובין המילה המתאימה בעמודה הרביעית.

מרחק	b מילה נקלטת	n – ביטים מקודדים מילת קוד	k ביטי מידע בכניסה למקודד	מספר סידורי
3	00000111	00000000	00000	0
2	00000111	00011111	00001	1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	$a_1^{(i)}, a_2^{(i)}, \dots, a_n^{(i)}$	.	i
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
$n-3$	00000111	.	11111	2^k-1

look up table – LUT

טבלה 2.2 מילות הקוד ומרחקן והמילה הנקלטת



איור 2.2 מקודד ואפנן במשדר, גלאי ומפענח במקלט

דוגמא 2.2 :

$k = 2$ 2 ביטים בכניסה
 $n = 6$ 6 ביטים ביציאה

$$R_c = \frac{k}{n} = \frac{2}{6} = \frac{1}{3} \text{ יחס הקידוד}$$

נבנה מקודד בעזרת LUT - Look Up Table המתואר בטבלה 2.3 .

טבלה 2.3 מהווה את החלק השמאלי בטבלה 2.4 המתוארת בהמשך.

מידע	מילת קוד
00	000000
01	010101
10	101010
11	111111

טבלה 2.3 תהליך הקידוד בעזרת LUT של קוד דוגמא 2.2

אם נשווה בין כל שתי מילות קוד המרחק המינימלי הינו $d = 3$

מרחק של קוד $d = 3$

נקלטה מילה 000001 ע"פ אלגוריתם המפענח השלם. נבחר את מילת הקוד 000000 הנמצאת במרחק המינימלי מהמילה הנקלטת כלומר המידע הינו 00. נתאר את תהליך הקליטה בטבלה 2.4 .

נרשום בעמודה ליד LUT את מרחקה ממילת הקוד (צד ימין בטבלה).

	מידע	מילת קוד	מילה נקלטת	מרחק למילת קוד
1	00	000000	000001	1
2	01	010101	000001	2
3	10	101010	000001	4
4	11	111111	000001	5

טבלה 2.4 תהליך הפענוח עבור הקוד מדוגמא 2.2 ומילה נקלטת 000001

נחזור על הדוגמא כאשר נוסיף את תהליך הקידוד והשגיאות בערוץ.
נתאר שני מקרים בהם נקלטת אותה מילה:

- א. המידע לשידור 00
מילת הקוד לאחר קידוד 000000
בערוץ אירעה שגיאה במקום 6 ולכן קיבלנו 000001
בפענוח קיבלנו 00 ולכן הפענוח הצליח.
- ב. המידע לשידור 01
מילת הקוד לאחר קידוד 010101
בערוץ אירעו שגיאות במקום 2 ו 4 ולכן קיבלנו 000001
בתהליך הפענוח קיבלנו 00 ולכן הפענוח שגוי.

2.3 יכולת תיקון שגיאות

משפט 2.1: קוד בעל מרחק d יכול לתקן t שגיאות כאשר:

$$d \geq 2t + 1$$

ניתן לרשום את קשר התיקון בצורה הבאה:

$$\begin{aligned} \text{עבור } d \text{ אי זוגי} \quad d = 2t + 1 \quad \text{כאשר} \quad t = \frac{d-1}{2} \\ \text{עבור } d \text{ זוגי} \quad d = 2t + 2 \quad \text{כאשר} \quad t = \left\lfloor \frac{d-1}{2} \right\rfloor \end{aligned}$$

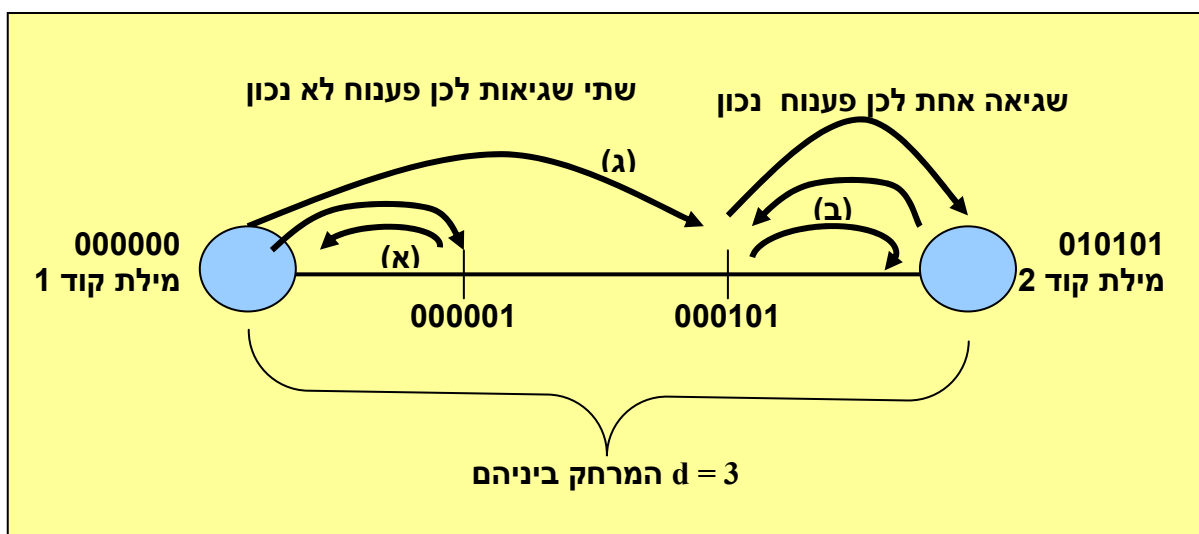
דוגמא 2.3: בדוגמא 2.2 מרחק הקוד $d = 3$ אזי לפי המשפט 2.1:

$$3 \geq 2t + 1$$

כלומר יכולת התיקון של הקוד היא $t = 1$. לקוד יש יכולת לתקן שגיאה אחת.
אם נתבונן במילים של דוגמא 2.2 כווקטורים במרחב 6 ממדים נוכל לצייר שתי מילים לדוגמא 000000, 010101 כשתי נקודות במרחק שלוש.

נסמן באיור 2.3 שלושה אירועים:

- (א) ישנה שגיאה במקום השישי של 000000 שמביאה ל - 000001 במרחק אחד מ - 000000 ובמרחק שתיים מ - 010101. המפענח יתקן את השגיאה ויפענח נכון 000000.
- (ב) ישנה שגיאה במקום השני של 010101 שמביאה ל - 000101 במרחק אחד מ - 010101 ומרחק שתיים מ - 000000. המפענח יתקן את השגיאה ויפענח נכון 010101.
- (ג) ישנן שתי שגיאות במקום הרביעי והשישי של 000000 שמביאה ל - 000101 במרחק אחד מ - 010101 ובמרחק שתיים מ - 000000. המפענח יבחר את 010101 וזאת אינה המילה ששודרה.



איור 2.3 יכולת תיקון שגיאות במילות קוד מדוגמא 2.3

הוכחה של משפט 2.1 :

- המילה המשודרת היא a^0
 - נניח שהיו t שגיאות והמילה הנקלטת עקב כך הינה b . לכן המרחק בין המילה הנקלטת למילה הנכונה הינה t .

$$d(a^0, b) = t$$

המרחק למילה אחרת a^1 הקרובה במרחק d מ- a^0 והוא שווה או גדול מ- $d - t$

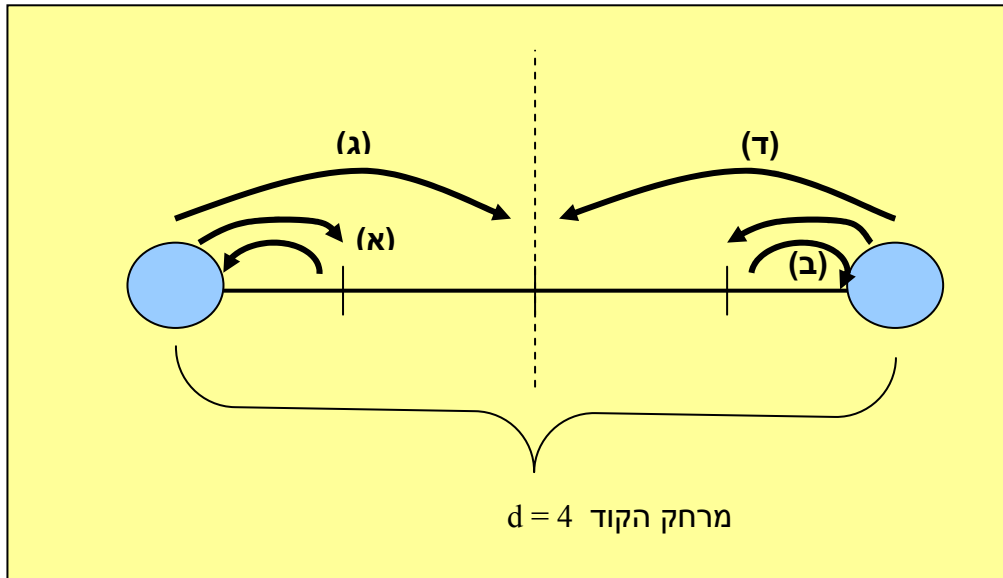
$$d(a^1, b) \geq d - t$$

הפענוח מצליח כאשר $d - t > t$ כלומר $d > 2t$

ולכן נרשום $d \geq 2t + 1$, מש"ל

דוגמא 2.4:

נבחר קוד בעל $d = 4$. באיור 2.4 נסמן שתי מילות קוד במרחק 4 ונציין 3 אירועים בעזרת חצים. חצים היוצאים מהנקודות מציינים שגיאה ואילו החצים הנכנסים לנקודות מציינים פענוח. בדומה למקרה הקודם במקרים (א) ו (ב) יש שגיאה המפוענחת נכון. במקרים (ג) ו (ד) יש שתי שגיאות בכל אחת אשר מביאות לנקודה כלשהי באמצע הדרך בין שתי המילים במרחק של שתיים משניהם. במצב זה המפענח יכול רק לבחור באקראי את אחת משתי המילים או לחילופין לא לתקן.



איור 2.4 קידוד ופענוח עבור $d = 4$

$$d \geq 2t + 1$$

$\Downarrow$

$$4 \geq 2t + 1$$

$\Downarrow$

$t = 1$ לכן נוכל לתקן רק שגיאה אחת

נבנה טבלה של יכולת תיקון השגיאות - t לעומת המרחק - d עבור $1 < d < 9$

מרחק d הקוד	t כמות שגיאות שניתן לתקן
1	0
2	0
3	1
4	1
5	2
6	2
7	3
8	3
9	4

טבלה 2.5 t בתלות ב- d

2.4 מפענח לא שלם – incomplete decoder

תהליך הפענוח מתחיל כמו במפענח השלם: עבור מילה נקלטת נמדד המרחק בינה ובין כל מילות הקוד ונבחרת המילה עם המרחק הקטן ביותר, אבל עבור מרחק בין מילה נקלטת למילות הקוד הקטנה מ- t מתבצע תיקון כמו במפענח השלם. אם המרחק המינימלי בין המילה הנקלטת למילות הקוד (הנבחרת) גדול מ- t הוא לא מתקן, כלומר אין לנו מצב של תיקון שגיאות במקרה זה. נשאלת השאלה מה יכול להתקבל במוצא במקרה זה.

א. דגל "לא מתקן" על ידי תו מידע שהוא (לדוגמה) 0 עבור מרחק קטן או שווה ל- $t - 1$ עבור מרחק גדול מ- t .

ב. מילה הנקלטת ללא תיקון (n ביטים).

ג. k ביטים מתוך n שמתאימים למידע (ללא תיקון).

ד. יוציא את המילה הקרובה ביותר והמידע המתאים כמו במפענח השלם.

החיסרון של מפענח לא שלם שהוא "מחמיץ" הזדמנויות לתקן יותר מ- t שגיאות במקרים מסוימים, יתרונו שיש לו יכולת מסוימת לגלות קיום שגיאות במרחק גדול מ- t .

2.5 יכולת תיקון וגילוי שגיאות

נבהיר את המושגים תיקון וגילוי שגיאות

לתקן = לגלות את מקום השגיאות ולתקן לביטים הנכונים.

לגלות = לגלות שיש שגיאות, להצביע על קיומם אך לא לתקן, לא לציין את מיקומם, ובדרך כלל לא לציין את מספרן.

משפט 2.2:

עבור קוד עם מרחק d אפשר למצוא מפענח:

- א. שמתקן t שגיאות או פחות.
- ב. מגלה u שגיאות כאשר $u > t$ (אם יש $t+1$ עד u אזי הוא לא מתקן אלא מגלה בלבד).

כך שמתקיים:

$$d \geq t + u + 1$$

; כאשר $u > t$ חייב להתקיים

הדגשה:

כאשר אומרים שקוד מתקן t שגיאות ומגלה u שגיאות מתכוונים לתהליך הבא:

אם במפענח יש $1, 2, 3, \dots, t$ שגיאות אזי מתקן (מתקן $1 \div t$ שגיאות).
אם במפענח יש $t+1, t+2, \dots, u$ שגיאות אזי אינו מתקן ומגלה קיום שגיאות בלבד (מגלה $t+1 \div u$ שגיאות).

דוגמא 2.5:

עבור קוד עם מרחק $d = 7$ יש מספר אופני עבודה.

אופן עבודה א':

ללא יכולת גילוי (תיקון בלבד)

$$\text{לכן ידוע כי } d \geq 2t + 1$$

$\Downarrow$

$$t \leq 3 \text{ כמות שגיאות שניתן לתקן}$$

אופן עבודה ב':

נניח כי רוצים כעת לתקן שגיאות כאשר יכולת הגילוי הינה $u = 4$

$$d \geq t + u + 1$$

$\Downarrow$

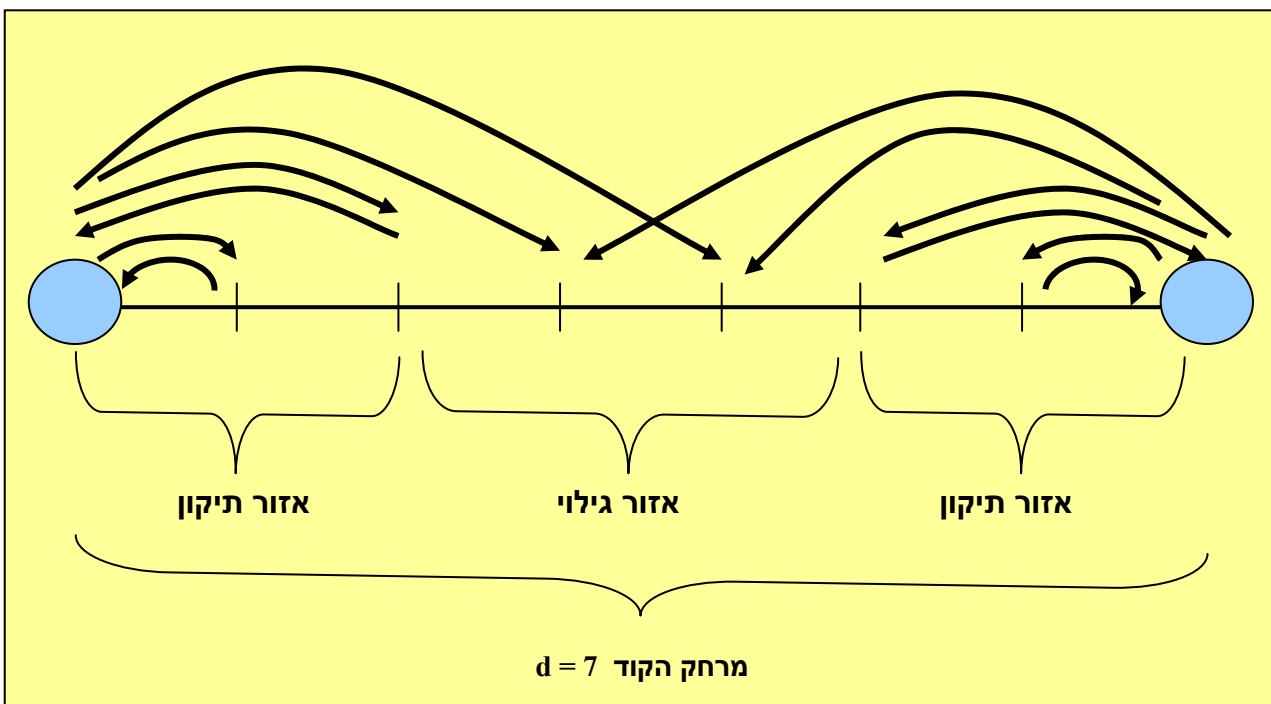
$$d \geq t + 4 + 1$$

$\Downarrow$

$$t \leq 2$$

כמות השגיאות שניתן לתקן כרגע עד 2. כלומר מתקן עד 2 ומגלה 3 עד 4.

נראה באיור 2.5 את אופן עבודה ב' בצורה גרפית:



איור 2.5 תיאור של תיקון עד 2 שגיאות וגילוי 3 או 4 שגיאות עבור מרחק קוד 7

אופן עבודה ג':

נניח כי רוצים כעת לתקן שגיאות כאשר יכולת הגילוי הינה $u = 5$

$$\begin{aligned}d &\geq t + u + 1 \\&\Downarrow \\d &\geq t + 5 + 1 \\&\Downarrow \\t &\leq 1\end{aligned}$$

כלומר מתקן עד שגיאה אחת ומגלה 2 עד 5 שגיאות.

אופן עבודה ד':

נניח כי רוצים כעת לתקן שגיאות כאשר יכולת הגילוי הינה $u = 6$

$$\begin{aligned}d &\geq t + u + 1 \\&\Downarrow \\d &\geq t + 6 + 1 \\&\Downarrow \\t &\leq 0\end{aligned}$$

במקרה הזה לא נוכל לתקן שגיאות בכלל אך נוכל לגלות עד 6 שגיאות.

נרכז בטבלה את הקשר בין יכולת הגילוי- u והתיקון - t למרחק שהינו $d = 7$

עמודת המשמעות מציינת שהמערכת או שמגלה או שמתקנת (לא מבצעת את שתיהן יחד).

אופן עבודה	d	u	t	משמעות
א	7	-	3	יכול לתקן עד 3 שגיאות לא יכול לגלות בכלל.
ב	7	4	2	אם יש עד שתי שגיאות – מתקן, אם יש יותר מ-2 שגיאות הוא רק מגלה
ג	7	5	1	אם יש שגיאה אחת מתקן, אם יש יותר משגיאה אחת אז הוא רק מגלה
ד	7	6	0	מגלה עד 6 שגיאות

טבלה 2.6 הקשר בין יכולת הגילוי והתיקון עם קוד במרחק $d = 7$ ומשמעותה

2.6 יכולת תיקון וגילוי שגיאות בערוץ עם מחיקות

משפט 2.3:

אם יש x מחיקות (כלומר שלא ניתן לדעת מהי ההחלטה),
המפענח יכול לתקן עד t שגיאות כך שמתקיים:

$$d \geq x + 2t + 1$$

$\Downarrow$

$$d - x \geq 2t + 1$$

כלומר מחיקה מקטינה את היכולת האפקטיבית של הפענוח לתיקון שגיאות או מקטינה את המרחק האפקטיבי של הקוד.

דוגמא 2.6 :

עבור קוד עם מרחק $d = 3$ יכולת הפענוח מתוארת בטבלה:

x	t	פענוח
0	1	נכון
1	0	נכון
1	1	לא נכון
2	0	נכון
2	1	לא נכון
3		לא נכון

מעל $x > 2$ המפענח כבר לא מסוגל לתקן.
במקרה של רעש חזק עדיף להגיע להחלטה על ביצוע מחיקה במקלט, כלומר אין התייחסות לאות הנקלט ואז כאילו הגדלנו את כושר התיקון שלו פי 2.

אם נציב $t = 0$ נקבל $x_{\max} = d - 1$ כאשר $x_{\max}$ הינו מספר המחיקות המכסימלי שבו נקבל פענוח נכון.

דוגמא 2.7 :

בקוד שמתואר בדוגמא 2.2 נקלטת מילה עם שתי מחיקות $010x0x$ נרשום את מרחק המילה הנקלטת תוך התעלמות מהמחיקות.

מרחק	נקלט	מילת קוד (n)	מידע (k)
1	010X0X	000000	00
0	010X0X	010101	01
4	010X0X	101010	10
3	010X0X	111111	11

המפענח בוחר את המילה עם מרחק 0.

משפט 2.4 :

בערוץ עם מחיקות ושגיאות, אם בערוץ יש x מחיקות אזי המפענח יכול לתקן עד t שגיאות ולגלות $u > t$ שגיאות (גילוי u שגיאות ותיקון t שגיאות).

$$\boxed{d \geq x + t + u + 1}$$
 אם מתקיים :

2.7 סיכום המשפטים וניתוחם:

נסכם את 4 המשפטים העיקריים:

(1) תיקון שגיאות

$$\boxed{d \geq 2t + 1}$$

(2) תיקון וגילוי שגיאות

$$\boxed{d \geq t + u + 1}$$

(3) תיקון שגיאות בערוץ עם מחיקות

$$\boxed{d \geq 2t + x + 1}$$

(4) תיקון וגילוי שגיאות בערוץ עם מחיקות

$$\boxed{d \geq u + t + x + 1}$$

הערה 1: רשום נוסף של הקשר (1)

נראה שאת משוואה (1): $d \geq 2t + 1$

ניתן לרשום גם בצורה הבאה:

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor$$

הסימון משמעו שמעגלים כלפי מטה.

אם לדוגמא נציב $d = 8$ נקבל: $t = \left\lfloor \frac{8-1}{2} \right\rfloor$ כלומר המנה 3.5, מעגלים כלפי מטה, כלומר התוצאה $t = 3$.

הערה 2: השוואה בין תיקון שגיאות ללא גילוי וגילוי ללא תיקון עבור d אי זוגי.

ניקח את משוואה (2) עבור $t = 0$ (כלומר נוכל רק לגלות ולא לתקן שגיאות)

$$u = d - 1 \iff d \geq 0 + u + 1$$

ואם נשווה את מה שקיבלנו נראה שעבור גילוי קיבלנו $d - 1$ ועבור תיקון קיבלנו

$\frac{d-1}{2}$ כלומר: הקוד יכול לגלות פי 2 שגיאות אם הוא לא מתקן מאשר הוא יכול לתקן מבלי לגלות. זה נכון במדויק עבור d אי זוגי ונכון בערך עבור d אי זוגי

הערה 3: השוואה בין תיקון שגיאות (ללא גילוי) ותיקון מחיקות (ללא גילוי)

כעת נעסוק במשוואות (1) ו-(3):

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor \text{ עבור משוואה (1) קיבלנו}$$

עבור משוואה: $d \geq 2t + x + 1$ (3) נעסוק במצב של מחיקות בלבד כלומר עדיין $t = 0$ ואז נקבל כי $x = d - 1$ וקיבלנו שיכולת הטיפול והתיקון במחיקות הוא פי 2 מיכולת תיקון שגיאות (זה נכון במדויק עבור d אי זוגי ונכון בערך עבור d אי זוגי).

2.8 ביצועים של קודי בלוקים עם החלטות קשות

נניח הסתברות לשגיאה לביט ביציאת הגלאי p (של הביטים המקודדים בכניסה למפענח) אם יש לנו קוד (n, k, t) כלומר:

k - ביטי מידע

n - ביטים מקודדים

t - יכולת התיקון

מה ההסתברות לפענוח נכון של מילה p_{cw} ?
מה ההסתברות לפענוח לא נכון למילה p_w ?
הסתברות לפענוח שגוי של מילה נזכור שמספר האפשרויות לבחור i מתוך k הוא:

$$\binom{k}{i} = \frac{k!}{i!(k-i)!} \quad k \geq i$$

נתייחס למפענח שלם ולמפענח לא שלם
נתחיל עם מפענח לא שלם בו מתקנים מילים שגויות במרחק שקטן או שווה ל t

הסתברות לפענוח מילה נכונה:

עבור מפענח לא שלם

$$p_{cw} = \sum_{i=0}^t \binom{n}{i} (1-p)^{n-i} p^i$$

מפענח שלם יכול לתקן חלק מהמילים השגויות במרחק גדול מ t ועבור הסתברות לפענוח נכון:

עבור מפענח שלם

$$p_{cw} \geq \sum_{i=0}^t \binom{n}{i} (1-p)^{n-i} p^i$$

הסתברות לפענוח שגוי של מילה במפענח שלם:

$$p_w = 1 - p_{cw} =$$

$$p_w \leq \sum_{i=t+1}^n \binom{n}{i} p^i (1-p)^{n-1}$$

עבור מפענח לא שלם כאשר המפענח לא מתקן נכון אזי הוא יכול לשגות או במקרים מסוימים לגלות שגיאות אזי:

$$P_w + P_{cw} + P_d = 1$$

ושב מקבלים:

$$p_w \leq \sum_{i=t+1}^n \binom{n}{i} p^i (1-p)^{n-1}$$

כמו בעבור מפענח שלם.

הסתברות לפענוח של מילה לא נכונה היא הסתברות ליותר מ- t שגיאות כאשר:

i - מספר השגיאות
 p - הסתברות השגיאה לביט
 t - כמות שגיאות

נשאלת השאלה מה הסתברות השגיאה בביט בודד לאחר הפענוח ?

עבור מפענח לא שלם שמתקן עד t שגיאות ומעבר לזה הוא לא מנסה, ניתן להשתמש בביטוי:

$$p_b \leq \sum_{i=t+1}^n \frac{i+t}{n} \binom{n}{i} p^i (1-p)^{n-1}$$

הביטוי $\frac{i+t}{n}$ נותן חסם להסתברות שגיאה לביט במידע שגוי שהיו בו i שגיאות לפני הפענוח והמפענח טעה והוסיף t שגיאות.

ניתן לעשות קירוב מאוד מקובל:

$$p_b \cong \sum_{i=t+1}^n \frac{i}{n} \binom{n}{i} p^i (1-p)^{n-1}$$

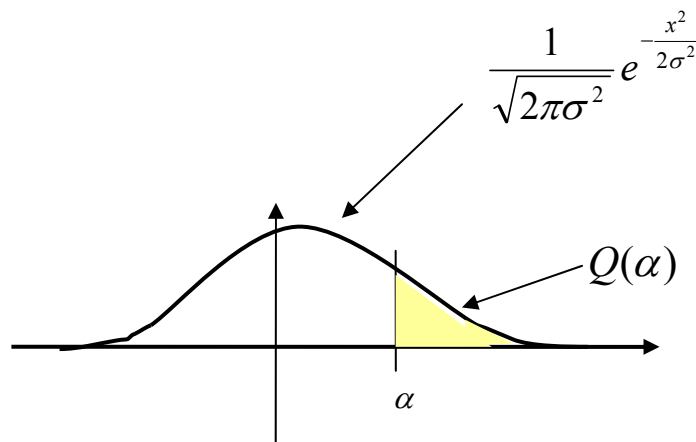
הסתברות שגיאת מידע מקודד

קצב שגיאות המידע המקודד P תלוי ביחס אות לרעש ובשיטת האפנון.

- עבור שיטות אפנון אנטיפודליות כגון $QPSK$, $BPSK$ ללא קוד לתיקון שגיאות נקבל:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

כאשר $Q(\alpha)$ מוגדר בפעמון גאوسی כ:



$$Q(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{-\frac{x^2}{2}} dx$$

- הסתברות שגיאה עבור פענוח דיפרנציאלי של אותות בינאריים עם קידוד דיפרנציאלי DBPSK :

DBPSK = Differential Binary Phase Shift Keying

נתונה על ידי:

$$p_b = \frac{1}{2} e^{-\frac{E_b}{N_0}}$$

דוגמא 2.8 :

נדגים את תהליך החישוב של אות מקודד עבור DBPSK

ביציאת הגלאי הסתברות השגיאה:

$$p = \frac{1}{2} e^{-\frac{E_{eb}}{N_0}}$$

Encoded Bit = eb

E_{eb} אנרגיה של ביט מקודד ניתן לרשום בתלות ב E_b :

$$p = \frac{1}{2} e^{-\frac{E_b \cdot k}{N_0 \cdot n}}$$

כאשר $\frac{k}{n}$ הינו יחס הקידוד. את הגודל p הנ"ל מציבים במשוואות של p_b מעמוד קודם.

הערה:

הקידוד גורם להקטנת האנרגיה מ E_b ל E_{eb} ועל ידי כך להגדלת מספר השגיאות. המפענח צריך לתקן את השגיאות ולכן הקוד יועיל כאשר יכולת התיקון יכולה להתגבר על כמות השגיאות שגדלה. במקרה זה שבח הקידוד הוא חיובי.

2.9 קודי חזרות - Repetition - m

העיקרון הוא חזרה על הביט m פעמים כלומר:

$$1 \Rightarrow 1111111111$$

חזרה על "1" m פעמים

$$0 \Rightarrow 0000000000$$

חזרה על "0" m פעמים

הפענוח נעשה על פי החלטת הרוב. אם יש יותר אחדים מאפסים אזי "1", אם יש יותר אפסים מאחדים אזי "0". נניח לדוגמא: $m = 5$ ונתון 11010 אז הפענוח הוא לפי החלטת הרוב כלומר "1".

תיאור (n, k, d, t) עבור קודי חזרות:

$$d = m, \quad n = m, \quad k = 1$$

$$t = \left\lfloor \frac{m-1}{2} \right\rfloor$$

ולכן עבור קוד חזרות m נקבל:

$$\left(m, 1, m, \left\lfloor \frac{m-1}{2} \right\rfloor \right)$$

2.10 קודי בלוקים - קוד (n,k,t) BCH

קודי BCH הם קודים בינאריים שתהליך הקידוד והפענוח לא יכולה בספר זה. בטבלה 2.7 נתונים קודי BCH הטובים ביותר בעלי מרחק אי זוגי.

טבלה 2.7 קודי BCH :

כל הקודים בטבלה זו הינם עבור

$$d = 2t + 1$$

עבור $n = 7, 15, 31, 63, 127$

n	k	t
7	4	1
7	1	3
15	11	1
15	7	2
15	5	3
15	1	7
31	26	1
31	21	2
31	16	3
31	11	5
31	6	7
31	1	15
63	57	1
63	51	2
63	45	3
63	39	4
63	36	5
63	30	6
63	24	7
63	18	10
63	16	11
63	10	13
63	7	15
63	1	31
127	120	1
127	113	2
127	106	3
127	99	4
127	92	5
127	85	6
127	78	7
127	71	9
127	64	10
127	57	11
127	50	13
127	43	14
127	36	15
127	29	21
127	22	23
127	15	27
127	8	31
127	1	63

מתקיים:

$$n = 2^l - 1. \quad l \text{ הינו מספר שלם, כלומר } n \text{ יכול להיות } 63, 127 \text{ וכו'.$$

הקודים עם $k = 1$ הם למעשה קודי חזרות, לדוגמא $(63,1,31)$ ביט אחד מיוצג ע"י 63 ביטים ואז ניתן לתקן 31 שגיאות מתוכם.

2.10.1 ביצועי קוד BCH

כאשר אנו דנים בביצועי הקוד אנו מתייחסים:

- $\frac{E_b}{N_0}$ הסתברות השגיאה בתלות
- רוחב סרט.
- סיבוכיות.

הערה 1:

הגדלת n מסבכת את המימוש אך גם משפרת את הסתברות השגיאה.

הערה 2:

עבור n מסוים כאשר t הולך וגדל אזי יחס הקידוד קטן R_{eb} גדל וגם רוחב הסרט גדל.

הערה 3:

כאשר אין מגבלה של רוחב סרט ובחרנו n מסוים, עבור קצב מידע R_b מוגדר והספק מוגדר, איזה קוד נבחר?

הקוד שנבחר יהיה בעל הסתברות השגיאה הטובה ביותר כאשר הוא נמצא איפה שהוא באמצע
טבלה של n - המסוים. (זה יהיה האופטימום) ומדוע? כי כאשר t גדל k קטן גם R_c קטן.
ההספק של הביט המקודד E_{eb} יקטן לפי:

$$E_{eb} = E_b \cdot R_c$$

והסתברות לשגיאה ביציאת הגלאי מתקלקלת לפי:

$$p = f\left(\frac{E_b}{N_0} \cdot R_c\right)$$

כאשר $f(\cdot)$ תלויה בצורת האפנון.

למשל במקרה של גלאי דיפרנציאלי נקבל:

$$p = \frac{1}{2} e^{-\frac{E_b}{N_0} R_c}$$

לכן כאשר אנרגיה לביטים קטנה יש יותר הסתברות לשגיאות ואז אומנם העלנו את t אבל העלנו במקביל את הסתברות השגיאות.

נסכם את התופעה:

- (א) כאשר R_c קטן R_{eb} גדל וגורם להגדלת רוחב הסרט $\Leftarrow$ אנרגיה של הביטים המקודדים קטנה $\Leftarrow$ הסתברות השגיאה p_e ביציאת הגלאי גדלה.
- (ב) כאשר R_c קטן t גדל ומתקן יותר שגיאות.
- (ג) בקודים מעשיים ישנו t אופטימאלי.

2.11 שינויים לקוד בלוקים נתון

נתייחס שלושה תהליכים:

- קיצור
- ניקוב
- הרחבה

2.11.1 קיצור - Shortening

תהליך הקיצור מקובל בקודי בלוקים סיסטמטיים.

כיצד מקצרים קוד סיסטמטי נתון (n, k, t) לאורך הודעה יותר קטנה מ k ?

תהליך הקידוד

$\Leftarrow$ ההודעה באורך $k - l$ ולכן בשלב הראשון נוסיף l אפסים כדי להגיע להודעה באורך k .

$\Leftarrow$ בשלב השני נוכל לקודד את ההודעה עם הקוד המקורי (n, k, t) .

← בשלב השלישי מכון והקוד סיסטמטי ננפה את l האפסים ונקבל $n - l$ ביטים מקודדים.

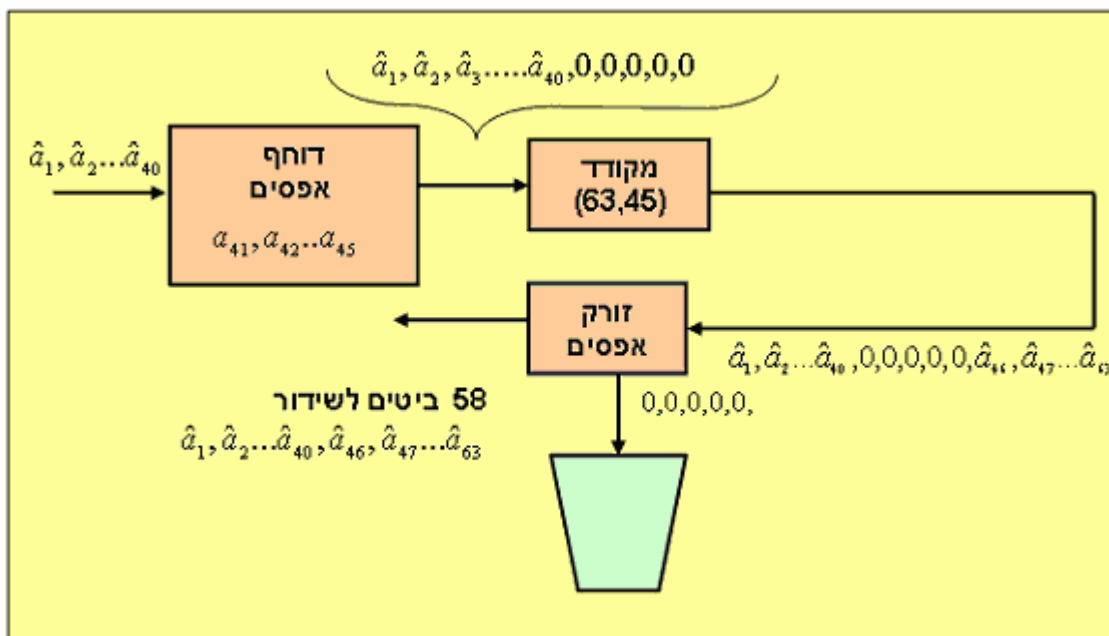
ואז הקוד המקוצר יהיה: $(n - l, k - l, t)$ יחס הקידוד קטן עקב הקיצור.

דוגמא 2.9:

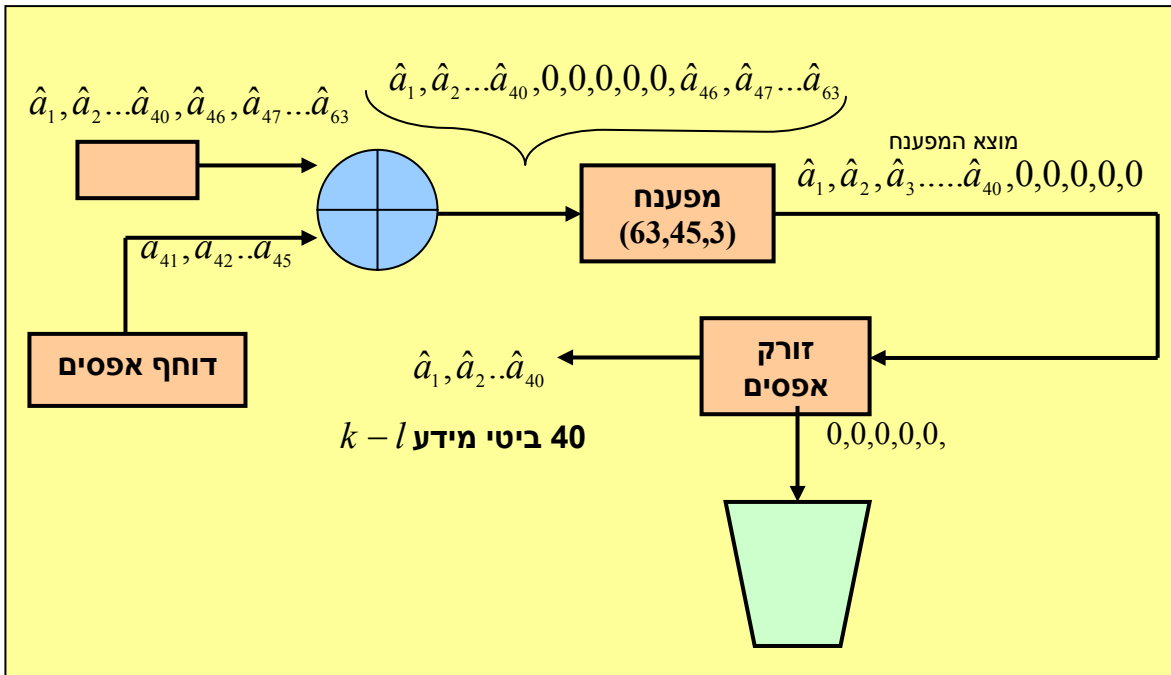
נתון קוד מקורי: $(63, 45, 3)$. אנו דורשים קיצור של $l = 5$ ולכן הקוד החדש יהיה $(58, 40, 3) \leftarrow (63 - 5, 45 - 5, 3)$ בפענוח מוסיפים אפסים במקום האפסים שלא שודרו ומבצעים פענוח רגיל. לאחר מכן מסלקים את הביטים במקומות שמתאימים לאפסים שנוספו. אם הפענוח הצליח במקומות אלו יש אפסים.

מהלך הקידוד של קוד מקוצר מתואר באיור 2.6

מהלך הפענוח של קוד מקוצר מתואר באיור 2.7



איור 2.6 קידוד קוד מקוצר (58,40)



איור 2.7 פענוח קוד מקוצר (58,40)

2.11.2 ניקוב קוד - Puncturing

תהליך הניקוב מקובל בקודי קונבולוציה לצורך קבלת יחס קידוד גדול מהמקור.

ישנם k ביטי מידע מקודדים ל- n ביטים. לא משדרים l מתוכם (מסוימים) ומשדרים את ה- $n-l$ האחרים.

עבור הקוד המקורי: (n, k, t)

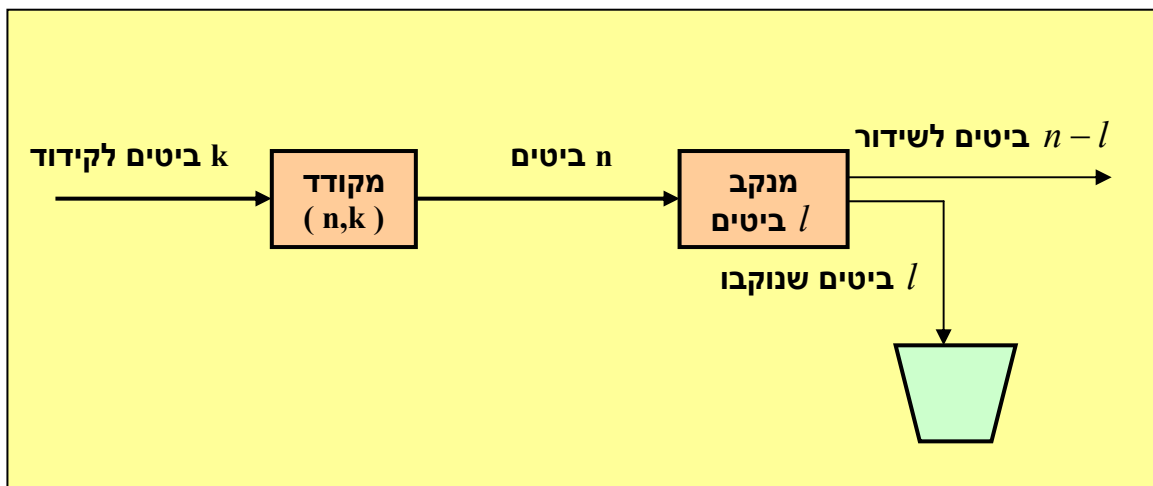
נקבל לאחר ניקוב l ביטים קוד מנוקב: $(n-l, k, t')$, כאשר $t' = t - \left\lceil \frac{l}{2} \right\rceil$. בתהליך הפענוח במקום הביטים שלא שודרו מעבירים למפענח מחיקות.

דוגמא 2.10 :

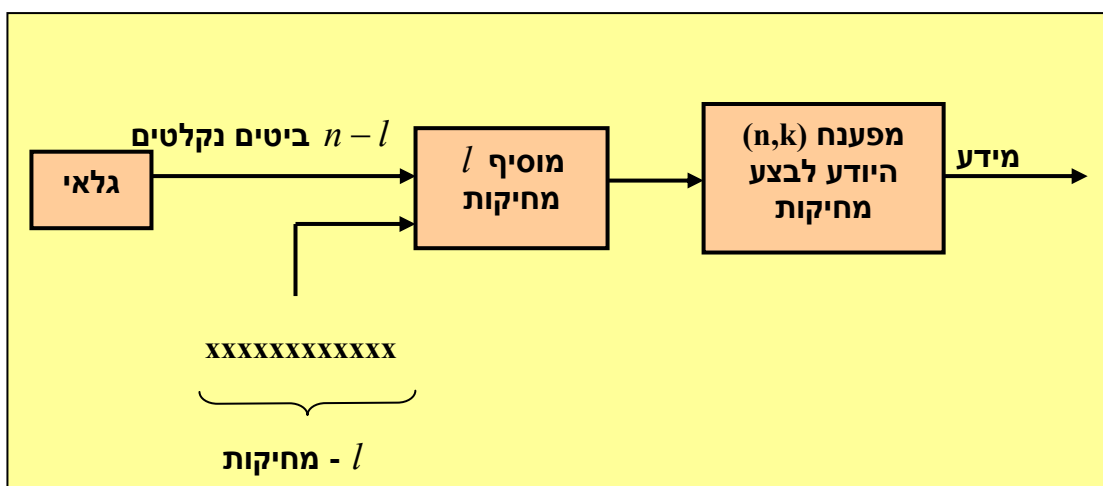
נתון קוד $(63,30,6)$ ואנו רוצים להגיע ליחס קידוד $\frac{1}{2}$ על ידי ניקוב כלומר $l=3$ אזי הקוד המנוקב יהיה $(60,30,t')$ כאשר $t' = t - 2$

מהלך הקידוד של קוד מנוקב מתואר באיור 2.8 .

מהלך הפענוח של קוד מנוקב מתואר באיור 2.9 .



איור 2.8 קידוד קוד מנוקב



איור 2.9 פענוח קוד מנוקב

2.11.3 הרחבת קוד – Expanding

הרחבה הינה הגדלת n המתקבלת על ידי הוספת ביטי יתירות המקובלת בקודי בלוקים. אנו נתייחס לצורה הבאה בלבד:

עבור קוד בלוקים עם קוד מקורי (n, k, d) עם מרחק אי זוגי, ניתן להוסיף ביט זוגיות כביט יתירות אחד ואז מקבלים: $(n+1, k, d+1)$

במקרה ש- d הוא אי זוגי אזי:

$$d = 2t + 1 \quad (\text{הקוד המקורי})$$

הערה: כאשר d אי זוגי מתקיים שוויון ואפשר לכתוב $=$ ולא צריכים לכתוב $\leq$ או $\geq$.

הקוד המורחב יהיה: $d' = d + 1 = t + u + 1$ מרחק הקוד המורחב

$$2t + 1 + 1 = t + u + 1$$

$\Downarrow$

$$u = t + 1$$

הגדלת המרחק באחד אינה מגדילה את t אך מוסיפה יכולת גילוי $u = t + 1$ שגיאות.

דוגמא 2.11 :

אם נתון קוד מקורי BCH $(63, 45, 3)$ וידוע כי $d = 2t + 1$ אזי $d = 7$ נרחיב את הקוד בביט יתירות אחד ואז נקבל קוד $(64, 45, 3)$. t נשאר קבוע ואילו המרחק גדל ל- $d = 8$. ניתן לראות כי יכולת התיקון לא השתנתה, אך אם d גדל אזי יכולת הגילוי גדלה, כיצד?

ידוע כי $d \geq u + t + 1$. נבדוק את שני המצבים לפני הרחבה ואחרי:

במצב לפני הרחבה בו $d = 7$ אזי: $7 \geq u + 3 + 1$ כלומר $u = 3$

במצב לאחר הרחבה בו $d = 8$ אזי: $8 \geq u + 3 + 1$ כלומר $u = 4$

ניתן לראות שלאחר הרחבה עבור אותה יכולת תיקון $t = 3$ גדלה יכולת הגילוי של u .

סיכום

	(n, k, d, t)	עבור קוד נתון:
יחס קידוד קטן	$(n-l, k-l, d, t)$	לאחר קיצור :
יחס קידוד גדל	$\left(n-l, k, d-l, t-\left\lceil \frac{l}{2} \right\rceil\right)$	לאחר נקוב :
עבור d אי זוגי יחס הקידוד קטן	$(n+1, k, d+1, t)$	לאחר הרחבה :

2.12 קוד Reed - Solomon

זהו קוד סיסטמטי לא בינארי, העובד על אותיות .

הוא מסומן : $RS_b(n, k)$ ופועל על משפחה של אותיות, הנקראת אלף-בית, של 2^b אותיות .

לדוגמא עבור $b = 4$

(0)	0 0 0 0	האלף-בית יהיה
(1)	0 0 0 1	
(2)	0 0 1 0	
(3)	0 0 1 1	
(4)	0 1 0 0	
(5)	0 1 0 1	
(6)	0 1 1 0	
(7)	0 1 1 1	
(8)	1 0 0 0	
(9)	1 0 0 1	
(10)	1 0 1 0	
(11)	1 0 1 1	
(12)	1 1 0 0	
(13)	1 1 0 1	
(14)	1 1 1 0	
(15)	1 1 1 1	

ה- b הכי נפוץ הוא $b = 8$, כל אות היא בת שמונה סיביות כלומר כל אות היא בעצם בית byte .

תכונות קוד RS : (א) $n = 2^b - 1$

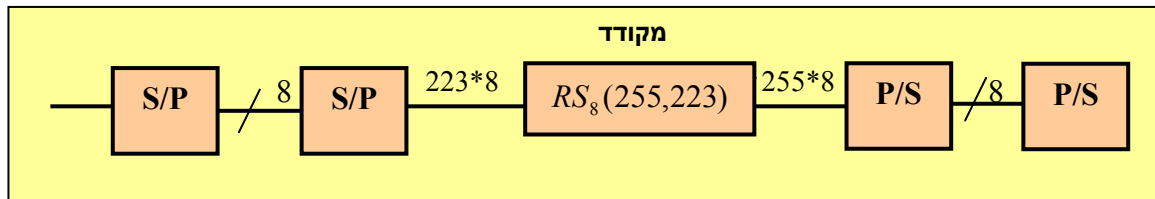
(ב) $d = n - k + 1$

(ג) $t = \left\lfloor \frac{d-1}{2} \right\rfloor \Rightarrow t = \left\lfloor \frac{n-k}{2} \right\rfloor$

דוגמא 2.12 :

נדון במבנה : $RS_8(255,223)$

223 נבחר כך שיכולת התיקון תהיה $t = 16$ ולכן $n - k = 32$



איור 2.10 מקודד RS של דוגמא 2.12

דוגמא של שימוש בתקנים של DVB :

DVB-S - לוויינים
DVB-C - כבלים
DVB-T - תחום UHF

דוגמא 2.13 :

שיטת הדחיסה של אות חוזי בה משתמשים היא : MPEG2
גודל חבילה היא של 188 בתים.

אם נתון $t = 8$ אזי:

א. בחירת n

$$t = \left\lfloor \frac{n - k}{2} \right\rfloor$$

$$\Downarrow$$

$$\begin{cases} k = 188 \\ t = 8 \end{cases}$$

$$\Downarrow$$

$$n = 204$$

ב. קוד לפני קיצור

אנחנו יודעים שקוד RS צריך להיות 255 ולכן נעשה קיצור שיביא אותנו ל- 204 :

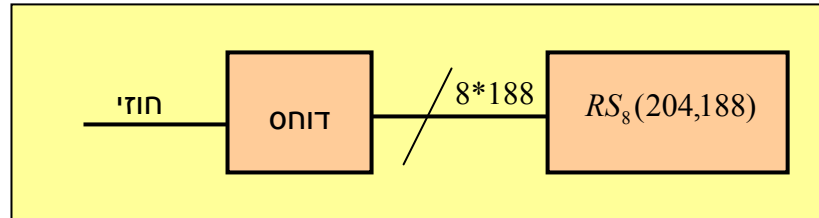
$$RS_8(255,239)$$

$$255 - 239 = 16 = 2t \text{ נתון}$$

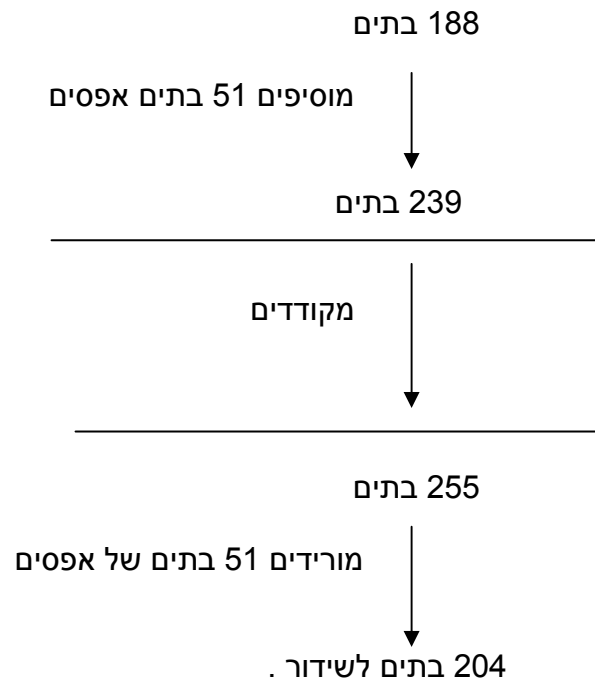
ג. פרמטר קיצור l

רואים ש :

$$l = 51 \quad (255 - 204)$$



ד. תהליך הקידוד



למה נשתמש בקוד RS ?

- טוב נגד פריצי שגיאות (לא כל- כך יעיל נגד רעש אקראי בלתי תלוי) .
- יעיל למימוש בקצבים גבוהים ועבור בלוקים גדולים .

איזה אורך פרץ שגיאות n_b יכול לתקן קוד RS שמתקן t שגיאות? כדי לענות לשאלה זאת נראה בכמה אותיות הפרץ פוגע.

נתחיל את ההסבר עבור $n_b = ib$ כאשר i מספר שלם.
 אם n_b הביטים מתחילים בתחילת אות אזי i אותיות נפגעו ולא $i+1$ אותיות נפגעו.
 ללא תלות בתחילת הפרק נפגעו $i+1$ אותיות. $n_b = ib + 1$

אם הפרץ מתחיל בסוף אות אזי $i+2$ אותיות נפגעו ולא רק $i+1$ אותיות.
בדיקה של $n_b = ib + l$ עבור $l = 3, \dots, b-1$ מראה שנפגעו $i+1$ או $i+2$ אותיות.

לסיכום מספר האותיות שנפגעו על ידי פרץ באורך $n_b = ib + l$ הוא $i+1$ עבור $l = 0, 1$ ו $i+2$ עבור $l = 2, \dots, b-1$. או לחילופין:

$$\left\lfloor \frac{n_b - 2}{b} \right\rfloor + 2$$

על כן יש להשוות :

$$\begin{aligned} \left\lfloor \frac{n_b - 2}{b} \right\rfloor + 2 &\leq t \\ \Downarrow \\ \left\lfloor \frac{n_b - 2}{b} \right\rfloor &\leq t - 2 \\ \Downarrow \\ n_b - 2 &\leq (t - 2)b + l' \\ l' &\leq b - 1 \\ \Downarrow \\ n_{b \max} &= (t - 2)b + b - 1 + 2 \\ \Downarrow \\ &= (t - 2)b + b + 1 \\ \Downarrow \\ n_{b \max} &= (t - 1)b + 1 \end{aligned}$$

דוגמא 2.14 :

האם הקוד (204,188) יכול לתקן פרץ באורך של 60 ביטים הפוגע במילה ?

אם יש לנו פרץ של 60 ביטים אז זה שווה ערך ל - 9 בתים במצב הכי גרוע כי בעצם לא יודעים מתי הפרץ יתחיל . הקוד לא יכול לתקן אותו.

$$\begin{aligned} n_{b \max} &= (t - 1)b + 1 = && \text{הפרץ הגדול ביותר} \\ &= (8 - 1)8 + 1 = 58 && \text{ביטים} \end{aligned}$$

חישובי שגיאה עבור קוד RS

ניקח ערוץ שהסתברות לשגיאה לביט p .
מה ההסתברות לאות שגוי (אות $b =$ ביטים)?

תשובה מקורבת הינה :

$$P_{ch} \approx bp$$

אבל התשובה המדויקת היא:

$$p_{ch} = 1 - (1 - p)^b$$

הסבר :

$$1 - p \Leftarrow \text{אין שגיאה לביט}$$

$$(1 - p)^b \Leftarrow \text{אין שגיאה באות}$$

$$1 - (1 - p)^b \Leftarrow \text{יש שגיאה באות}$$

מה ההסתברות P_w שה RS – שוגה בפענוח המילה ?

שגיאה בפענוח מילה של $RS_b(n, k, t)$

$$p_w = \sum_{i=t+1}^n \binom{n}{i} p_{ch}^i (1 - p_{ch})^{n-i}$$

נראה כעת מהי ההסתברות לאות שגויה בקרוב ?

$$p_{ch} \approx \sum_{i=t+1}^n \frac{i}{n} \binom{n}{i} p_{ch}^i (1 - p_{ch})^{n-1}$$

כעת נראה מה ההסתברות לביט שגוי לאחר פענוח בקרוב ?

$$p_b \approx \sum_{i=t+1}^n \frac{2^{b-1}}{2^b - 1} \frac{i}{n} \binom{n}{i} p_{ch}^i (1 - p_{ch})^{n-1}$$

הגורם $\frac{2^{b-1}}{2^b - 1}$ - מבטא את ההסתברות של ביט שגוי מתוך אות שגוי
לדוגמא עבור $b = 3$ יש 2^b אותיות:

$(I) \Rightarrow$ 000
001
010
011
100
101
110
111

אם המילה הנכונה היא השורה I הראשונה אז נראה שההסתברות לשגיאה של כל אחד
משלושת הביטים שבה היא $\frac{4}{7}$.

2.13 קוד Golay-קוד מושלם

2.13.1 קוד גוליי

הקוד נתון: $G(23,12)$ עם $d = 7$ $t = 3$

בגלל שהמרחק הוא אי זוגי אז ניתן להרחיב ולכן לאחר הרחבה נקבל:

$G(24,12)$ עם $d = 8$ $t = 3$

הקוד הבסיסי הוא קוד מושלם (Perfect Code).

2.13.2 מודל הכדורים:

מילת קוד באורך n ניתנת לתיאור על ידי ווקטורים באורך n . כל מילת קוד היא נקודה במרחב ה- n מימדי.

סביב כל נקודה שמייצגת מילת קוד ניתן לבנות קידוד כדור ברדיוס t . המייצג את כל המילים במרחק t או פחות ממילת הקוד המקורית.

- כמה מילים יש בכדור אחד?

$$\underbrace{1}_{\text{במרחק 1}} + \underbrace{n}_{\text{במרחק 2}} + \underbrace{\binom{n}{2}}_{\text{במרחק 2}} + \dots + \underbrace{\binom{n}{t}}_{\text{במרחק t}}$$

- כמה כדורים יש בקוד? $2^k =$ כמספר מילות הקוד.

$$2^k \sum_{i=0}^t \binom{n}{i} \quad \text{מספר המילים בכל הכדורים:}$$

2.13.3 הגדרת קוד מושלם - Perfect code

כל המילים הן במרחק t או פחות ממילה חוקית כלומר לא יוצאות מגבולות הכדור ברדיוס t . לכן מתקיים:

$$2^n = 2^k \sum_{i=0}^t \binom{n}{i}$$

נבדוק את הנוסחה לגבי קוד Golay $G(23,12)$ על ידי הצבת $t=3, k=12, n=23$

$$2^{23} = 2^{12} \left(1 + 23 + \frac{23 \cdot 22}{2} + \frac{23 \cdot 22 \cdot 21}{23} \right) = 2^{12} (1 + 23 + 253 + 1771) = 2^{12} (2048)$$

אכן קיבלנו שיוון מה שמאמת את הטענה שקוד Golay הינו קוד מושלם.

טענה 1

עבור קוד מושלם - מפענח שלם = מפענח לא שלם.

הסבר: מפענח שלם מוצא את מילת הקוד הקרובה ביותר למילה הנקלטת. מפענח לא שלם גם הוא מוצא את מילת הקוד הקרובה ביותר למילה הנקלטת אבל אם המרחק הקטן ביותר גדול מ- t לא מתבצע תיקון בכלל. היות ואין מילים שמרחקן גדול מ- t מהמילה החוקית הקרובה ביותר אזי הטענה מוכחת.

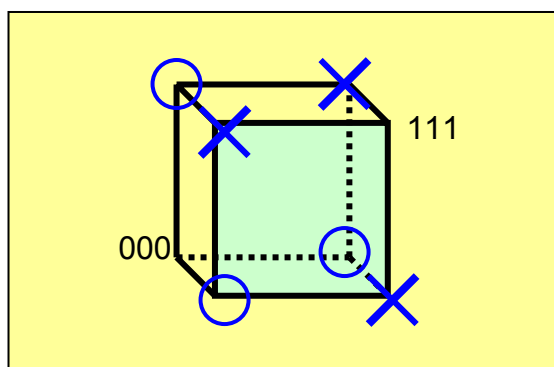
טענה 2

עבור קוד מושלם עם יכולת תיקון t , אם בונים כדור ברדיוס t סביב כל מילת קוד אזי הכדורים לא חופפים ומכסים את כל המרחב.

דוגמא 2.15 :

דוגמא של קוד מושלם:

קוד חזרות 3 עבורו יש שתי מילים $(0,0,0)$ $(1,1,1)$. $n=3$, $d=3$, $t=1$.
במרחב התלת מימדי נראה את שתי המילים באיור 2.12.
הכדור ברדיוס 1 מ- 000 מסומן ב- O ואילו הכדור ברדיוס 1 מ- 111 מסומן ב- X.



איור 2.11 שתי המילים החוקיות במרחב התלת מימדי

מתקיים:

$$2^3 = 2^1 \left(1 + \binom{3}{1} \right)$$

ולכן גם זה מושלם.

2.14 שאלות סיכום פרק

1. נתון קוד בעל מרחק 7. רשום בטבלה את מודי העבודה האפשריים המשלבים תיקון וגילוי שגיאות.

2. השוואה בין הקודים שבטור שמואל לבין הקודים שבטור ימין. ורשום איזה קוד עדיף בכול שורה. ($n_2 > n_1, k_2 > k_1, t_2 > t_1$)

קוד 1	קוד 2	עדיפות
n_1, k_1, t_1	n_1, k_1, t_2	
n_1, k_1, t_1	n_1, k_2, t_1	
n_1, k_1, t_1	n_2, k_1, t_1	
n_1, k_1, t_2	n_1, k_2, t_1	
n_1, k_1, t_1	$2n_1, 2k_1, t_1$	
n_1, k_1, t_1	$2n_1, k_1, 2t_1$	
n_1, k_1, t_1	$2n_1, 2k_1, 2t_1$	
n_1, k_1, t_1	n_2, k_2, t_2	

3. בהינתן קוד עם הפרמטרים הבאים (n, k, d, t) מה הפרמטרים של הקודים שמתקבלים בכל אחד מהאופנים הבאים:

3.1 מחיקת z ביטים מקודדים.

3.2 בקוד סיסטמטי קבע z מתוך k כאפס ומחק אותם אחרי הקדוד

3.3 הוסף ביט זוגיות כאשר d אי זוגי.

סכם את התוצאות בטבלה:

	n	k	d	t	הערות
1					
2					
3					

4. נתון קוד BCH $(63, k, t)$. קצר אותו לשם קבלת יחס קידוד $\frac{1}{2}$. בחר את k ו t הטובים ביותר. מה הם הפרמטרים k, t ו- n לאחר הקצור.

5. עבור קוד BCH (127, 64, 10) ועבור ערוץ BSC עם הסתברות שגיאה P . מה הסתברות לשגיאה בפענוח מילה? ומה הסתברות לשגיאה לביט (אחרי הפענוח)?

6. נתון קוד בלוקים (n, k) בעל מרחק המינג 7 במערכת עם החלטות קשות. מה יכולת התיקון וגילוי שלו? ציין את כל האפשרויות (בתשובה צריכים להיות 4 אפשרויות).

6.1 עבור כל מקרה תן נוסחה של הסתברות לתיקון השגיאה ולגילוי השגיאה. עבור שגיאה של ביט מקודד P .

6.2 חשב את הנוסחאות עבור $n=50$ ו $P=10^{-2}$.

6.3 עבור שיטת אפנון BFSK הסתברות השגיאה הינה $P = \frac{1}{2} e^{-\frac{1}{E/N_0}}$

כאשר E הינה האנרגיה של סימבול בערוץ. שרטט את הסתברות בשגיאה לביט של המערכת מקודדת ושל מערכת לא מקודדת כתלות ב E_b/N_0 כאשר $k=25$.

7. עבור שלושה קודי BCH (127, 92, 5), (127, 64, 10), (127, 36, 15).

7.1 רשום נוסחה של הסתברות שגיאה לביט עבור ערוץ BSC.

7.2 חשב את הסתברות השגיאה לביט כתלות ב- E_b/N_0 עבור שיטה אפנון BFSK.

7.3 מה שבח הקדוד הטוב ביותר עבור $P=10^{-5}$?

8. נתון $RS_8(204, 188)$ בשיטת אפנון BFSK.

8.1 שרטט סכמת מלבנים עקרונית של המשדר.

8.2 מהו יחס הקידוד R_c .

8.3 חשב הסתברות שגיאה למילה.

8.4 חשב הסתברות שגיאה לאות.

8.5 חשב הסתברות שגיאה לביט.

9.

9.1 הראה שאם בערוץ תקשורת יש תמיד שתי שגיאות המפענח לקוד המינג

(15,11) תמיד ישגה ולא יצליח לתקן נכון.

9.2 הראה כיצד ניתן להגדיל קוד המינג (15,11) לקוד המינג (16,11) אשר

מתקן שגיאות יחידות ומגלה שיאות כפולות, מהו המרחק המינימלי של הקוד החדש.

10. הינתן קוד RS עם $t=8$ ו $b=8$

- | | |
|---|------|
| מהם n, k | 10.1 |
| מהו R_c | 10.2 |
| מהו מספר ביטי המידע בבלוק (מילה מקודדת) | 10.3 |
| מהו מספר ביטי הקידוד בבלוק (מילה מקודדת) | 10.4 |
| כיצד ניתן להתאים את הקוד ל-mpeg2 עם 188 bytes . | 10.5 |

3 קודי בלוקים – קידוד ופענוח קודים ליניאריים בעזרת

מטריצות

3.1 הגדרות

הגדרת קוד ליניארי

א. קוד שמילת האפס כלולה בו $(0 \dots 0)$.
ב. סכום של כל שתי מילות קוד מהווה מילת קוד.

סכום וכפל בינארי

$$\text{חיבור לפי } \oplus \text{ XOR} \left\{ \begin{array}{ll} 0+0=0 & 0 \cdot 0=0 \\ 0+1=1 & 0 \cdot 1=0 \\ 1+0=1 & 1 \cdot 0=0 \\ 1+1=1 & 1 \cdot 1=1 \end{array} \right\} \begin{array}{l} 0 \cdot x=0 \\ 1 \cdot x=x \end{array}$$

חיסור בינארי:

$$0-0=0$$

$$0-1=1$$

$$1-0=1$$

$$1-1=0$$

רואים שטבלת החיסור שווה לטבלת החיבור וזה רק כמובן במבנה בינארי.

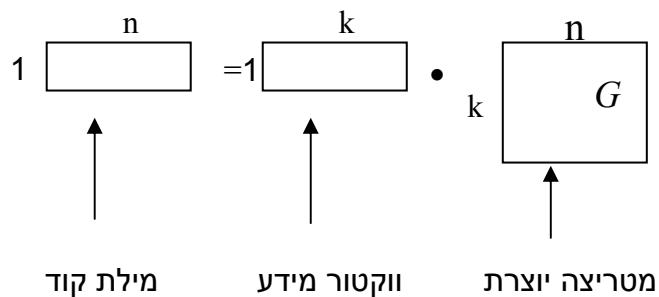
אם נתון: ווקטור שורה $a = [a_1 \dots a_n]$
ווקטור שורה $b = [b_1 \dots b_n]$

אזי: $a + b = [a_1 + b_1 \quad a_2 + b_2 \quad \dots \quad a_n + b_n]$ חיבור שני ווקטורים
כפל ווקטור בקבוע $\alpha a = [\alpha a_1 \quad \alpha a_2 \quad \dots \quad \alpha a_n]$
קבוע $\alpha =$

3.2 יצירת קוד בעזרת מטריצה יוצרת

מטריצה יוצרת (generator matrix) G גודלה $n \cdot k$ ומשמשת לייצר קודים בגודל (n, k) . היא בנויה מ- k שורות בלתי תלויות ליניאריות ו- n עמודות. על כן המימדים של הגדלים בהכפלה הן כמצויר. כל מילת קוד c היא צירוף ליניארי של k הוקטורים. המטריצה G משמשת ליצירת מילה מקודדת c מתוך מילת מידע לא מקודדת i . הקשר נתון על ידי:

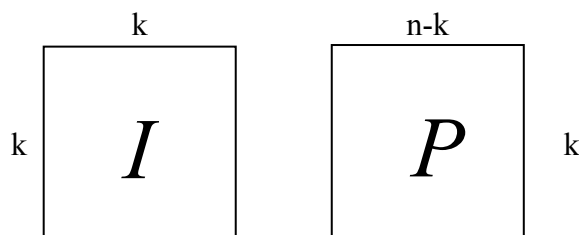
$$c = i \cdot G$$



אם חלק מהמטריצה G היא מטריצת יחידה אז מילת המידע i תישמר בתוך c ולכן הקוד סיסטמטי.

נסכם שקוד סיסטמטי כאשר:

$$G = [I \quad P]$$



נגדיר את הליניאריות:

$$[0 \quad \dots \quad 0] \cdot G = [0 \quad \dots \quad 0] \quad \text{א.}$$

$$(i_1 + i_2) \cdot G = i_1 \cdot G + i_2 \cdot G \quad \text{ב.}$$

3.3 מטריצה בודקת -H -parity check matrix

מטריצת H בעלת מימדים $(n-k) \cdot n$, שורות באורך n .

$$\begin{array}{c} n \\ \boxed{H} \\ n-k \end{array}$$

מילת קוד חוקית מוכפלת ב- H^T נותנת אפס.

$$c \cdot H^T = 0$$

המימדים בהכפלה הם:

$$1 \quad \begin{array}{c} n \\ \boxed{} \end{array} \cdot \begin{array}{c} n-k \\ \boxed{} \\ n \end{array} = 1 \begin{bmatrix} n-k \\ \end{bmatrix}$$

הקשר בין G ל- H:

$$G \cdot H^T = 0$$

טענה:

כל שורה של G היא למעשה מילת קוד.

הוכחה:

אם כל שורה של G מילת קוד אזי מתקיימת:

$$row_i G \cdot H^T = \begin{bmatrix} 0 & \dots & \dots & \dots & 0 \end{bmatrix} \quad 1 \quad n-k$$

$$G = \begin{bmatrix} row_1 G \\ row_2 G \\ \dots \\ row_n G \end{bmatrix} \Rightarrow G \cdot H^T = \begin{bmatrix} 0 & \dots & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix} \quad k \quad n-k$$

לסיכום המשוואות העיקריות הן:

$$c = i \cdot G \quad \text{קידוד}$$

$$c \cdot H^T = 0 \quad \text{בדיקת מילת קוד}$$

$$G \cdot H^T = 0 \quad \text{קשר בין G ו- H}$$

$$G = [I_n \quad P] \quad \text{G קוד סיסטמאטי}$$

$$\left. \begin{aligned} H^T &= \begin{bmatrix} -P \\ I_{n-k} \end{bmatrix} \\ H &= \begin{bmatrix} -P^T & I_{n-k} \end{bmatrix} \end{aligned} \right\} \text{H של קוד סיסטמאטי}$$

$$P = -P^T \quad \text{במקרה הבינארי}$$

$$I_n - \text{מסמנת מטריצת יחידה בעלת מימדים } n \times n$$

דוגמא 3.1 :

נתונה מטריצה יוצרת G בגודל 3×5 :

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

נתון גם ווקטור המידע i בגודל 1×3 :

$$i = [0 \quad 1 \quad 1]$$

המידע המקודד מתקבל על ידי הכפל של $i \cdot G$ ואז נקבל:

$$c = [0 \ 1 \ 1] \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{מטריצת יחידה}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} = [0 \ 1 \ 1 \ 1 \ 0]$$

ההכפלה היא סיכום ליניארי של השורות:

$$\begin{aligned} [0 \ 1 \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} &= 0 \cdot [1 \ 0 \ 0 \ 1 \ 0] + 1 \cdot [0 \ 1 \ 0 \ 0 \ 1] + 1 \cdot [0 \ 0 \ 1 \ 1 \ 1] \\ &= [0 \ 1 \ 1 \ 1 \ 0] \end{aligned}$$

נחשב את מטריצת H :

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

זוהי מטריצה בודקת המתאימה למטריצה יוצרת G:

$$[0 \ 1 \ 1 \ 1 \ 0] \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = [0 \ 0]$$

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}}_{-p^T} \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{I_{n-k}}$$

3.4 קוד המינג - Hamming

$$\begin{aligned}t &= 1 \\n &= 2^m - 1 \\k &= 2^m - 1 - m \\d &= 3\end{aligned}$$

m	n	k	הקוד המתקבל
3	7	4	$H(7,4)$
4	15	11	$H(15,11)$
5	31	26	$H(31,26)$

טבלה 3.1 קודי המינג בתלות ב- m

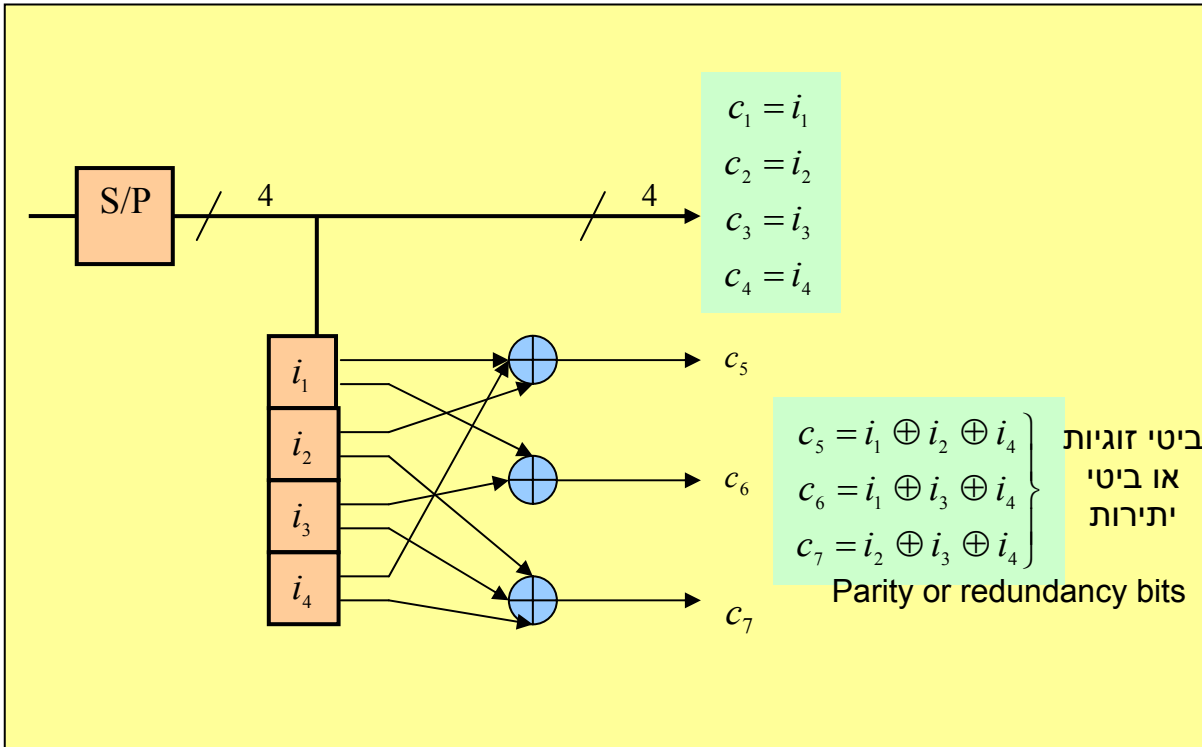
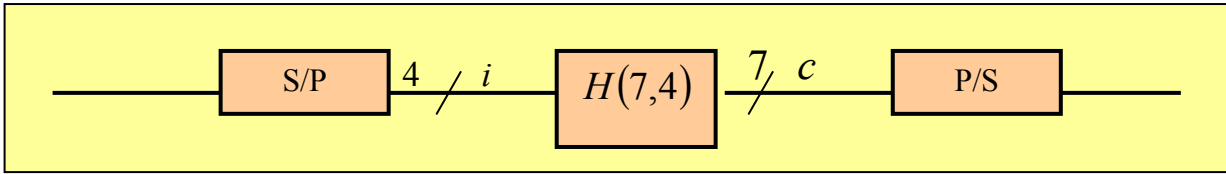
קוד המינג הוא קוד מושלם.

עבור $m = 3$ נבנה מטריצה יוצרת של $H(7,4)$

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right]$$

$$H = \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array} \right]$$

מימוש של המקודד:



איור 3.1 מימוש מקודד המינג (7,4)

$i = [i_1, i_2, i_3, i_4]$ (k ביטים) i_i ביט

$$i \cdot G = [i_1 \ i_2 \ i_3 \ i_4] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = [c_1 \ c_2 \ c_3 \ c_4 \ c_5 \ c_6 \ c_7]$$

3.5 פענוח קודים ליניאריים

שודר C

נקלט $v = C + e$

$C = [c_1, c_2, \dots, c_n] \Leftarrow$ ווקטור המייצג את מילת הקוד

$e = [e_1, e_2, \dots, e_n] \Leftarrow$ ווקטור המייצג את השגיאה

לדוגמא:

נתון: $e = [0010001]$

משמעות הדוגמא היא שביט 3 ו-7 שגויים.
כאשר נקלט C ללא שגיאה, הבדיקה מראה אפס.

$$C \cdot H^T = 0$$

יש לזכור שיש $n - k$ בדיקות ולכן וקטור אפס באורך $n - k$.
כאשר יש שגיאות מקבלים ווקטור המכונה סינדרום - Syndrome.

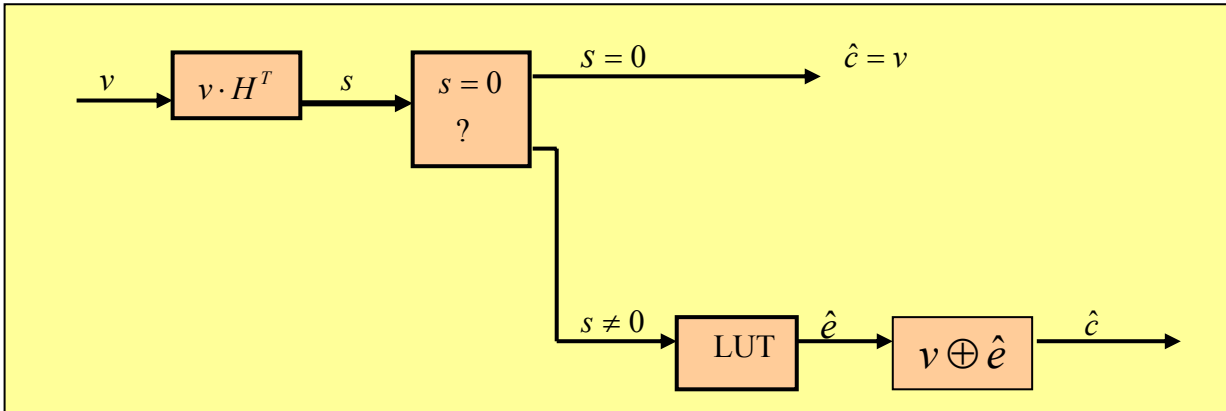
$$\underbrace{(C + e)}_v H^T = \underbrace{CH^T}_0 + \underbrace{eH^T}_s = s$$

אם $s = 0 \Leftarrow$ "אין שגיאות"
אם $s \neq 0 \Leftarrow$ "יש שגיאות"

תהליך הפענוח

- א. קולטים וקטור v
- ב. מחשבים את הסינדרום $s = v \cdot H^T \Leftarrow$
- ג. אם $s = 0$ אזי אין שגיאות.
- ד. אם $s \neq 0$, בודקים בתוך Look-up-table (LUT) - טבלת התאמה, מהי השגיאה המתאימה.
- ה. מתקן

מימוש המפענח נתון באיור 3.2 והוא מכונה מפענח סינדרום – Syndrome decoder



איור 3.2 מפענח סינדרום

לוקחים את המילה שנקלטת ועושים בדיקת Parity

מציאת LUT מתוך H

נמצא את LUT של $H(7,4)$ (של הדוגמא בסעיף 3.4)

$$H^T = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

נבנה את טבלת השגיאה בצורה הבאה:

נכפיל את H^T בווקטור של שגיאה בודדת.

נבנה את ווקטור השגיאה על ידי כך שנציב שגיאה במקום הראשון ובשאר אפסים, ולאחר מכן נציב שגיאה במקום השני ובשאר אפסים וכך הלאה לגבי כל האפשרויות.

לדוגמא:

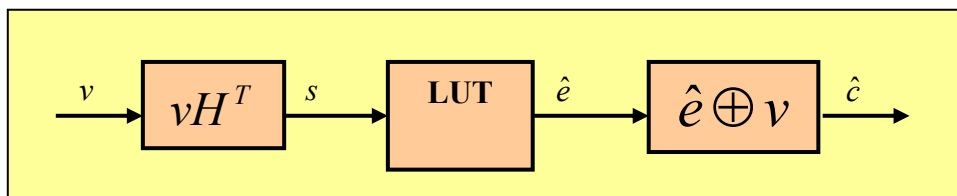
$$[1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \cdot \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [110]$$

רואים שבעצם השורה של H^T נשמרת

נסכם את ה LUT של קוד המינג (7,4) בטבלה 3.2:

S	e
110	1000000
101	0100000
011	0010000
111	0001000
100	0000100
010	0000010
001	0000001
000	0000000

טבלה 3.2 מיקום השגיאה בתלות במוצא הסינדרום של קוד המינג (7,4)



איור 3.3 מהלך פענוח מידע נקלט

נבחן מקרה של שתי שגיאות

לדוגמא במקומות ראשון ושני:

$$[1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0] \cdot \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [0 \ 1 \ 1]$$

רואים שהשגיאה שהמפענח מפענח היא במקום 3 כלומר לאחר ה"תיקון" שהמפענח מבצע, נוספה שגיאה.

3.6 הרחבה של קוד המינג.

פעולת ההרחבה הוסברה בסעיף 2.11.3

עבור קוד (n, k, d, t) , אפשר להוסיף לו ביט מקודד, על ידי בדיקת זוגיות של כל האחרים.

אזי עבור d אי זוגי הקוד החדש יהיה $(n+1, k, d+1, t)$.

הקוד החדש יכול לתקן t שגיאות כמו הקוד המקורי, אך בנוסף לזה הוא יכול לגלות $t+1$ שגיאות.

הוכחה: (הדגמה)

ניקח את הקוד (n, k, d, t)

המילה לאחר קידוד היא:

$$\left\{ \begin{array}{c} \underbrace{0 \ 0 \ \dots \ 0 \ 0}_n \\ \underbrace{0 \ 0 \ \dots \ 0}_{n-k} \ \underbrace{1 \ 1 \ \dots \ 1}_d \end{array} \right.$$

מילות קוד

אוסף המילים הכי קרובות
למילה המקורית היא

$$\text{יש } \binom{n}{d} \text{ מילים במרחק } d \text{ מ- } 0\dots 0$$

כעת נראה מה יקרה כאשר נעשה הרחבה, כלומר נוסיף עוד ביט בדיקת הזוגיות:

$$\begin{array}{r} 00\dots 000 \ 0 \\ 00\dots \underbrace{11\dots 1}_d \ 1 \end{array}$$

הוספת הביט ע"י חיבור XOR של הביטים האחרים. בשורה השנייה מוסיפים 1 כי d הוא אי זוגי. מה שבעצם קיבלנו בהרחבה זה מרחק קוד $d+1$.

3.7 שאלות סיכום פרק

- 1 נתון קוד לתיקון שגיאות מסוג הימנג בעל $2^m - 1$ ביטים עם m ביטי זוגיות.
 - 1.1 כתוב (k, n) לחמשת קודי המינג הראשונים החל מ- $m=3$.
 - 1.2 חשב את יחסי הקידוד של כול אחד מהקודים שמצאתה ב-1.1.
 - 1.3 מה היא הסתברות לשגיאה P_e , כאשר משתמשים בערוץ בינארי עם הסתברות שגיאה לביט p . כיצד הסתברות לשגיאה P_e , תלויה ב- n ?
- 2 נתון קוד המינג $m=3$ $(7,4)$ כאשר סיביות הזוגיות נתונות כך:
$$p_1 = i_1 \oplus i_2 \oplus i_3$$
$$p_2 = i_2 \oplus i_3 \oplus i_4$$
$$p_3 = i_1 \oplus i_2 \oplus i_4$$
 - 2.1 תאר את המטריצה היוצרת G .
 - 2.2 תאר את המטריצה בדיקת הזוגיות H .
 - 2.1 מצא LUT שקושר בין הסינדרום למקום השגיאה.
 - 2.2 רשום תרשים בלוקים סכמתי של תהליך הקידוד והפענוח.

3. נתון קוד המינג $m=3$ (7,4) כאשר סיביות הזוגיות נתונות כך:

$$p_1 = i_1 \oplus i_2 \oplus i_4$$

$$p_2 = i_1 \oplus i_3 \oplus i_4$$

$$p_3 = i_2 \oplus i_3 \oplus i_4$$

3.1 תאר את המטריצה היוצרת G .

3.2 תאר את המטריצה בדיקת הזוגיות H .

3.3 מצא LUT שקושר בין הסינדרום למקום השגיאה.

3.4 רשום תרשים בלוקים סכמתי של תהליך הקידוד והפענוח.

3.5 כאשר בערוץ תקשורת תמיד השגיאות מופיעות בזוגות הסבר מתי המפענח לקוד המינג (15,11) ישגה ומתי יצליח לתקן נכון.

3.6 הראה כיצד ניתן להגדיל את קוד המינג הנתון (15,11) לקוד

המינג (16,11) אשר מתקן שגיאות יחידות ומגלה שגיאות כפולות. מהו

המרחק המינימלי של הקוד החדש? מה המטריצה היוצרת G ?

4 קודי בלוקים - קודים ציקליים Cyclic code

4.1 ציקליות

קודים ציקליים הם קודי בלוקים ליניאריים שמקיימים את תכונת הציקליות ז"א אם $u = (u_1 \dots u_n)$ מילת קוד, אזי גם $(u_n, u_1, \dots, u_{n-1})$ מהווה מילת קוד. בתהליך הקידוד והפענוח עושים שימוש בייצוג בעזרת מילות הקוד - פולינומים.

מילת קוד כתובה כווקטור שורה :

$$\bar{u} = (u_0, u_1, \dots, u_{n-1}) \Rightarrow \text{ווקטור שורה}$$

יש אפשרות לכתובה כפולינום מסדר $n-1$:

$$u(x) = u_0 + u_1x + u_2x^2 + \dots + u_{n-1}x^{n-1}$$

כאמור השם של משפחת הקודים נובע מהתכונה שכל הזזה ציקלית של מילת קוד גם היא מילת קוד.

הסבר הזזה ציקלית:

אם נתון $u_0 \ u_1 \ u_2 \dots u_{n-1}$ אזי גם $u_{n-1} \ u_0 \ u_1 \dots u_{n-2}$ היא מילת קוד. (ביצענו הזזה ציקלית ימינה ב-1)

לאחר הזזות ציקליות מקבלים את מילת הקוד מיוצגת על ידי הפולינום.

$$u_{n-1} + u_0x + u_1x^2 + \dots + u_{n-2}x^{n-1}$$

טענה :

הזזה ציקלית של הפולינום ניתנת לחישוב על ידי:

$$u(x)^{(1)} = x \cdot u(x) \bmod (x^n + 1)$$

$$u(x)^{(i)} = x^i \cdot u(x) \bmod (x^n + 1)$$

הסבר לפעולת מודול של פולינום:

ללא פעולת מודולי סדר הפולינום גדל עם הזזה.

$$\begin{aligned}x \cdot u(x) &= x(u_0 + u_1x + \dots + u_{n-1}x^{n-1}) \\&= u_0x + u_1x^2 + \dots + u_{n-2}x^{n-1} + u_{n-1}x^n\end{aligned}$$

פעולת מודולי $(x^n + 1)$ נותנת:

$$\begin{aligned}xu(x) \bmod (x^n + 1) \\&= u_0x + u_1x^2 + \dots + u_{n-2}x^{n-1} + u_{n-1}x^n + u_{n-1}(x^n + 1) \\&\text{אבל } 0+0=0, 1+1=0 \text{ כי בחיבור בינארי } u_{n-1}x^n + u_{n-1}x^n = 0\end{aligned}$$

ולכן התשובה הסופית היא:

$$= u_{n-1} + u_0x + u_1x^2 + \dots + u_{n-2}x^{n-1}$$

דוגמא 4.1 :

$$\begin{aligned}u_a &= (0110) \\u_b &= (0111)\end{aligned} \quad \text{נתונות שתי מילים:}$$

נרצה לעשות הזזה ציקלית אחת ימינה בוקטורים.

$$\begin{aligned}u_a^{(1)} &= (0011) \\u_b^{(1)} &= (1011)\end{aligned}$$

כעת נבצע הזזה ציקלית בפולינום מסדר 3.

$$u(x) = u_0 + u_1x + u_2x^2 + u_3x^3$$

נציב ונקבל:

$$\begin{aligned}u_a(x) &= 0 + 1 \cdot x + 1 \cdot x^2 + 0 \cdot x^3 = x + x^2 \\u_b(x) &= 0 + 1 \cdot x + 1 \cdot x^2 + 1 \cdot x^3 = x + x^2 + x^3\end{aligned}$$

כעת נבצע הזזה על ידי הכפלה ב x מודולי הפולינום : $x^4 + 1$ הסדר 4 הוא גודל המילה המקורית

$$\begin{aligned} x \cdot u_a(x) \bmod (x^4 + 1) &= x^2 + x^3 \bmod (x^4 + 1) \\ &= x^2 + x^3 + 0(x^4 + 1) \end{aligned}$$

$\uparrow$
 u_3

$$\begin{aligned} & \begin{matrix} x + x^2 + x^3 \\ \Downarrow \end{matrix} \\ xu_b(x) \bmod (x^4 + 1) &= x^2 + x^3 + x^4 + \underbrace{x^4 + 1}_{u_3(x^4+1)} = 1 + x^2 + x^3 \end{aligned}$$

$u_3=1$

נראה שתי שיטות לקידוד ציקלי. הראשון פשוט יותר להבנה אך לא סיסטמטי השני הוא סיסטמטי הנפוץ.

4.2 שיטת קידוד ציקלי (לא סיסטמטית):

תהליך קידוד של קוד ציקלי (n, k) כולל הכפלת פולינום המידע בפולינום יוצר ומקבלים את המילה המקודדת. הקוד הוא לא סיסטמטי.

- פולינום יוצר:

$$g(x) = g_0 + g_1x + \dots + g_px^p$$

כאשר $p = n - k$

- המידע כווקטור : $(m_0, m_1, \dots, m_{k-1})$

$$\text{וכפולינום: } m_0 + m_1x + m_2x^2 + \dots + m_{k-1}x^{k-1}$$

- הקידוד מתבצע ע"י מכפלה של אחד בשני:

$$u(x) = g(x) \cdot m(x) \quad \text{סדר } n-1$$

$\nwarrow$ $\nearrow$
 פולינום מידע פולינום יוצר

פולינום של המילה המקודדת

דוגמא 4.2 :

קוד $(7,4)$

פולינום יוצר $g(x) = 1 + x + x^3$

$k = 4$ הביטים של המידע הם 1011 :

ולכן פולינום המידע: $m(x) = 1 + x^2 + x^3$

$$\begin{aligned} u(x) &= g(x) \cdot m(x) = 1 + x^2 + x^3 + x + x^3 + x^4 + x^3 + x^5 + x^6 \\ &= 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 \end{aligned}$$

הייצוג הווקטורי של המילה המקודדת הוא: $(1,1,1,1,1,1,1,1)$.

4.3 שיטת קידוד ציקלי שניה (סיסטמטית)

השיטה השנייה פותחה על מנת להשיג קוד סיסטמטי.

נתונים:

1. פולינום של המידע: $m(x)$ מסדר $k-1$

2. פולינום יוצר $g(x)$ מסדר $p = n - k$

השארית שמקבלים מחלוקת $x^{n-k}m(x)$ ב- $g(x)$ היא $p(x)$.

$$p(x) = x^{n-k}m \mod g(x)$$

$$u(x) = \underbrace{p(x)} + \underbrace{x^{n-k}m(x)} \quad \text{היא: } u(x) \text{ המקודדת}$$

ביטי מידע ביטי יתירות

טענה:

$u(x)$ זו כפולה של $g(x)$

הוכחה:

$$u(x) = p(x) + q(x) \cdot g(x) + p(x) = q(x)g(x)$$

דוגמא 4.3 :

קוד (7,4) עם פולינום יוצר

$$g(x) = 1 + x + x^3$$

המידע:

$$\bar{m} = (1011)$$

$$m(x) = 1 + x^2 + x^3$$

תהליך הקידוד:

א. חישוב $p(x)$

$$\begin{aligned} p(x) &= x^{n-k} m(x) \bmod g(x) \\ &= x^3 m(x) \bmod g(x) \\ &= x^3 (1 + x^2 + x^3) \bmod g(x) \\ &= x^3 + x^5 + x^6 \bmod (1 + x + x^3) \end{aligned}$$

$$\begin{array}{rcl} & x^3 + x^2 + x + 1 & \\ x^3 + x + 1 & \overline{) x^6 + x^5 + x^3} & \Leftarrow x^3 m(x) \\ & x^6 + x^4 + x^3 & \Leftarrow x^3 g(x) \\ & \hline & x^5 + x^4 & \\ & x^5 + x^3 + x^2 & \Leftarrow x^2 g(x) \\ & \hline & x^4 + x^3 + x^2 & \\ & x^4 + x^2 + x & \Leftarrow x g(x) \\ & \hline & x^3 + x & \\ & x^3 + x + 1 & \Leftarrow g(x) \\ & \hline & 1 & \Leftarrow p(x) \end{array}$$

ב. חישוב מילת קוד $u(x)$

$$u(x) = p(x) + m(x)x^{n-k}$$

$$u(x) = p(x) + m(x) \cdot x^3 = 1 + x^3 + x^5 + x^6$$

$$\bar{u} = (1001011)$$

פענוח:

נקלט:

$$z(x) = u(x) + e(x)$$

פולינום מילת קוד פולינום השגיאה

תהליך הפענוח:

א. מחלקים את $z(x)$ ב- $g(x)$

ב. אם השארית 0 אזי אין שגיאות, המילה של המידע נמצאת ב- k המקדמים של החזקות הגבוהות. כלומר בדוגמא עבור $\bar{u} = (1001011)$, $k = 4$.
לוקחים את $k = 4$ החזקות הגבוהות כמידע מפוענח כלומר את 1011
או בצורה אחרת מאפסים את $n - k$ המקדמים הנמוכים ולקבלת $m(x)$ מחלקים ב- x^{n-k} כלומר:

$$\begin{aligned} u(x) &= 1 + x^3 + x^5 + x^6 & / -1 \\ u(x) - 1 &= x^3 + x^5 + x^6 & / : x^3 \\ m(x) &= 1 + x^2 + x^3 \end{aligned}$$

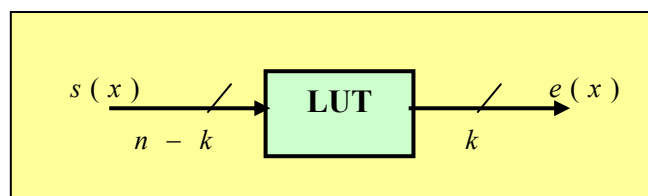
ובאופן כללי

$$\begin{aligned} z(x) &= 1 + 0 \cdot x + 0 \cdot x^2 + 1 \cdot x^3 + 0 \cdot x^4 + 1 \cdot x^5 + 1 \cdot x^6 \\ \frac{z(x) - 1}{x^3} &= \frac{x^3 + x^5 + x^6}{x^3} = 1 + x^2 + x^3 = m(x) \end{aligned}$$

ג. אם השארית היא $s(x)$ שונה מאפס כלומר:

$$z(x) = g(x)q(x) + s(x)$$

אזי ניתן להיעזר ב- LUT כדי למצוא את ווקטור השגיאה.



איור 4.1 דיאגרמת פענוח פולינום השגיאה משארית

$$z(x) + e(x) = \hat{u}(x) \quad \text{מתקנים}$$

מילה מקודדת לאחר תיקון

לאחר התיקון אין כבר שארית.
כדי למצוא את המידע בעזרת המילה המקודדת המתוקנת ניתן לבצע את סעיף ב'.

4.4 סיכום שתי השיטות

• מידע לא מקודד מסדר $k-1$ $m(x) = m_0 + m_1x + m_2x^2 + \dots + m_{k-1}x^{k-1}$

• פולינום יוצר: סדר $n-k$ $g(x) = g_0 + g_1x + g_2x^2 + \dots + g_{n-k}x^{n-k}$

קידוד לא סיסטמטי. יש פולינום יוצר היוצר את המילה המקודדת $u(x)$

סדר $n-1$ מידע מקודד $u(x) = m(x)g(x)$

קידוד סיסטמטי : $u(x) = p(x) + m(x)x^{n-k}$
סיסטמטי כאשר $p(x)$ הינה השארית בחלוקה של $m(x)x^{n-k}$ ב- $g(x)$

$$m(x)x^{n-k} = q(x)g(x) + p(x)$$

הערה: בשתי השיטות $u(x)$ מתחלק ב- $g(x)$.

4.5 מימוש של מקודד ומפענח של קוד ציקלי

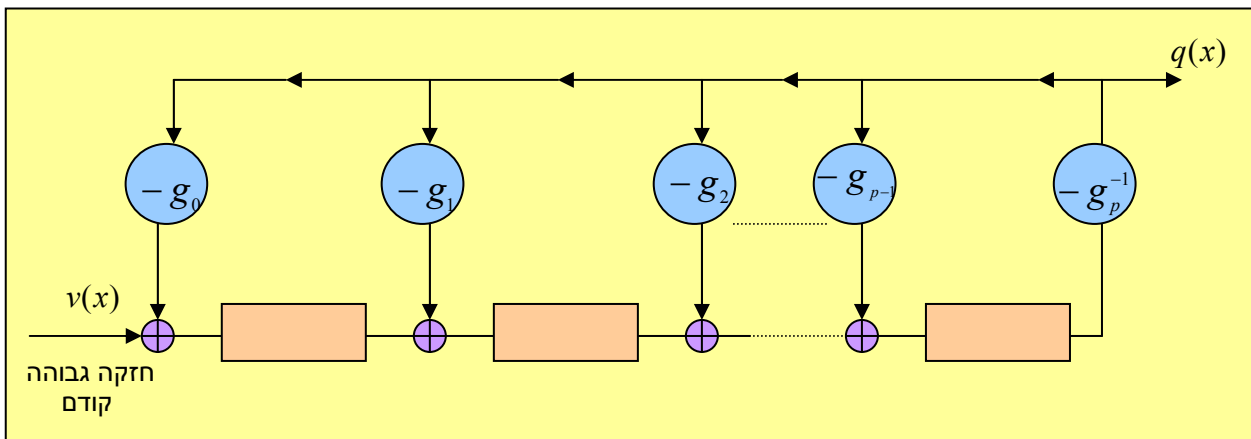
מעגל חלוקה של $v(x)$ ב- $g(x) = g_0 + g_1x + \dots + g_px^p$ (לאו דווקא בינאריים)

$$v(x) : g(x)$$

$$v(x) = q(x)g(x) + p(x)$$

$q(x)$ - מנה

$p(x)$ - שארית



איור 4.2 מעגל החלוקה

העיגול מסמן הכפלה והמלבן מסמן תא זיכרון. העיגול $\oplus$ מסמן XOR. בסיום $p(x)$ נמצא ברגיסטר.

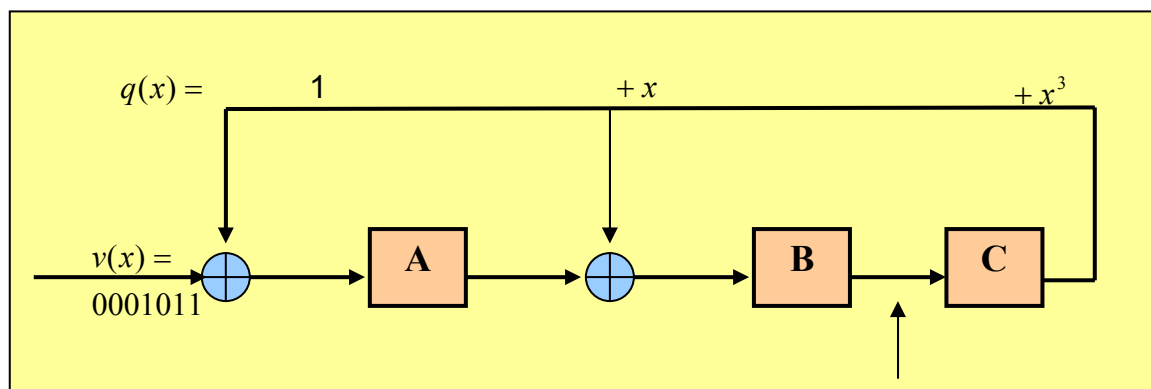
דוגמא 4.4 :

$$v(x) = x^3 + x^5 + x^6$$

נתון:

$$g(x) = 1 + x + x^3$$

מספר הריבועים (השהיות) יהיה לפי הסדר של $g(x)$:



פה אין חיבור ל $g(x)$ כי אין איבר של x^2

איור 4.3 דיאגרמת מעגל חלוקה דוגמא 4.4

נכניס בכניסה את $v(x)$ ונבנה טבלה:

זמנים	תור בכניסה	רגיסטר ABC	יציאה ומשוב
0	0001011	000	-
1	000101	100	0
2	00010	110	0
3	0001	011	0
4	000	011	1
5	00	111	1
6	0	101	1
7		100	1

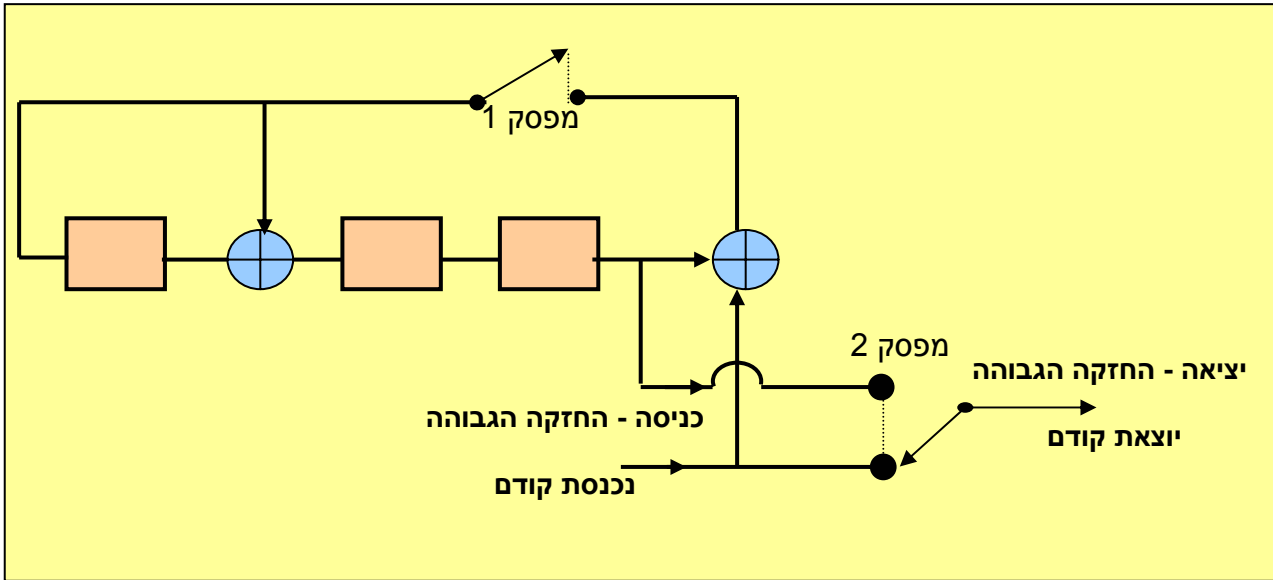
שארית $p(x)$

$$q(x) = 1 + x + x^2 + x^3$$

טבלה 4.1 פרוט מהלך האות במעגל החלוקה שבדוגמא

מימוש מקודד

עבור קוד $(7,4)$ עם פולינום יוצר $g(x) = 1 + x + x^3$

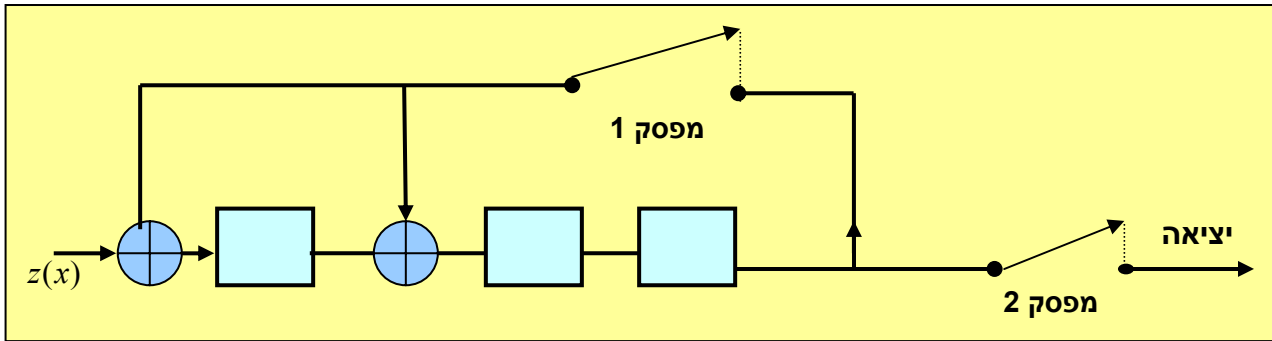


איור 4.4 מימוש מקודד עם פולינום יוצר $g(x) = 1 + x + x^3$

- שלב ראשון - מפסק 1 סגור משך k הזזות הראשונות, כדי להכניס את המידע לתוך $n - k$ תאים של הרגיסטר.
- שלב שני - מפסק 2 נמצא במצב נמוך (תחתון), כדי לאפשר שידור החוצה של ביטי המידע. (כי זהו קוד סיסטמטי). הוא צריך להישאר נמוך למשך k הזזות.
- שלב שלישי - אחרי שידור של k ביטי המידע (k הזזות) משנים את המפסקים : מפסק נפתח ומפסק 2 עובר למצב גבוה (עליון).
- שלב רביעי - $n - k$ הזזות נוספות מרוקנות את הרגיסטר שבו נמצאים ביטי היתירות (הכוונה היא $p(x)$).

מימוש מפענח

עבור קוד (7,4) עם פולינום יוצר $g(x) = 1 + x + x^3$ המעגל המחשב את הסינדרום המתואר באיור 4.5.

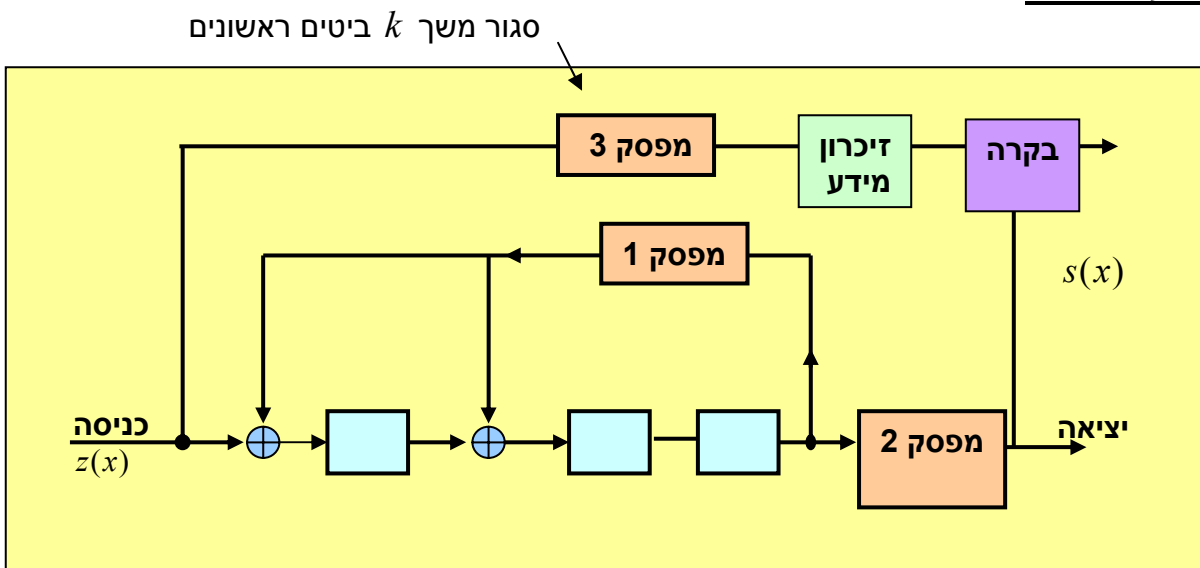


איור 4.5 מימוש מפענח עם פולינום יוצר $g(x) = 1 + x + x^3$

- שלב ראשון - מפסק 1 סגור, מפסק 2 פתוח.
מכניסים m ביטים נקלטים (קודם החזקה הגבוהה ביותר נכנסת). מתבצעת חלוקה, אחרי n הזזות ברגיסטר יש שארית.
- שלב שני - מפסק 1 נפתח, מפסק 2 נסגר.
משך $n - k$ הזזות מוצאים את הסינדרום - שארית $s(x)$ החוצה, כאשר החזקה הגבוהה ביותר ראשונה.
- שלב סיום - אם $s(x) = 0$ אין שגיאות.
אם $s(x) \neq 0$ יש שגיאות.

אם זהו קוד לגילוי שגיאות אז מספיק לנו לדעת רק מהו $s(x)$, אבל אם זהו קוד לתיקון שגיאות אזי צריך לתקן במידת הצורך ולהמשיך בפענוח כמו שהוסבר בסעיף 4.4.

המעגל הכולל

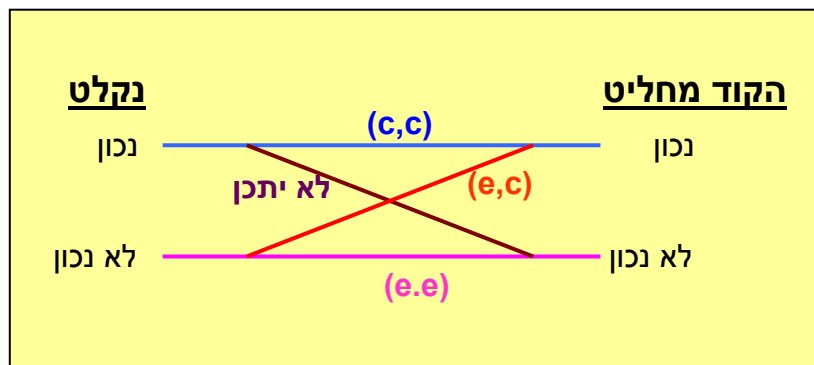


איור 4.6 מימוש המעגל הכולל

הבקרה מעבירה הלאה רק מידע תקין כאשר $s(x) = 0$

אם מעגל הבקרה מזהה $s(x) \neq 0$. הוא מסתכל ב-LUT (איור 4.1) ומוצא את וקטור השגיאה $e(x)$ המתאים (לגבי המידע בלבד) ומתקן.

4.6 ביצועים של קוד לגילוי שגיאות



איור 4.7 גלאי שגיאות

c=Correct
e=Error

מטרת הקוד לגילוי שגיאות הינה לזהות קיום שגיאה. אם האות נקלט נכון, אזי הקוד מזהה זאת (אירוע c, c). כאשר ישנה שגיאה הקוד יכול לזהות זאת (אירוע e, e), אך יכול גם לטעות (אירוע e, c).

הערות:

- לא יתכן שאם נקלטה מילה נכונה הקוד יחליט שהמילה לא נכונה (c, e).
- האירועים (c, c) ו (e, e) מהווים **פעולה נכונה** של המפענח לגילוי שגיאות.
- האירוע (e, c) הוא מהווה **תקלה חמורה**.

נשאלת השאלה מהי ההסתברות ל - P_{ec} ?

נדון בשני חסמים ל- P_{ec} עבור הקוד הוא (n, k, d) .

א. חסם עבור הסתברות שגיאה גבוהה.

עבור $p = \frac{1}{2}$ האות הנקלט לא תלוי באות המשודר.

סה"כ יש 2^n מילים. ישנם 3 מצבים אפשריים:

א. $2^n - 2^k$ מילים לא חוקיות

ב. 1 מילה חוקית ונכונה

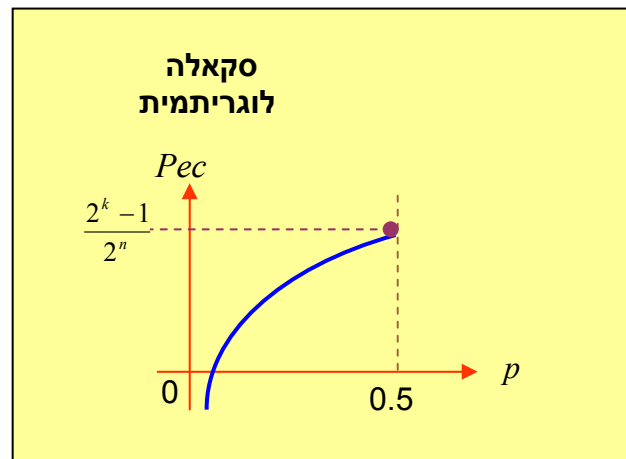
ג. $2^k - 1$ חוקיות ולא נכונות.

ההנחה מאחורי הניתוח:

אם נקלטת **מילה חוקית**, הקוד לא מגלה שגיאות ומכריז **נכון**.
אם נקלטת **מילה לא חוקית**, אז הקוד מחליט **לא נכון**.

$$Pec \leq \frac{2^k - 1}{2^n} \text{ ולכן הסתברות השגיאה בביט נקלט } Pec \leq \frac{2^k - 1}{2^n}$$

$$\approx 2^{-(n-k)}$$



איור 4.9 גרף הסתברות פענוח מילה לא נכונה כמילה נכונה.

ב. חסם עבור הסתברות שגיאה נמוכה.

חסם נוסף עבור הסתברויות קטנות לשגיאה של הביטים הנקלטים. נחשב את Pec עבור שגיאות p קטנות.

ההסתברות שיש בדיוק i שגיאות היא:

$$\binom{n}{i} p^i (1-p)^{n-i}$$

אם המרחק d אזי הוא יכול בוודאות לגלות $d-1$ ללא תיקון ואז:

$$Pec < \sum_{i=d}^n \binom{n}{i} p^i (1-p)^{n-i}$$

דוגמא 4.5 :

k - מידע לא מקודד

$n - k = 16$ יתירות

2^n מילים בסך הכול

2^k מילים נכונות

אם מקבלים מילה באקראי ונגדיר אותה כלא נכונה אזי הסיכוי שהיא תהיה מילה נכונה היא:

$$P_{ec} = 2^{-16} = 2^{k-n} = \frac{2^k - 1}{2^n}$$
$$\approx \frac{1}{65,000} \cong 1.5 \cdot 10^{-5} \quad P_{FA}$$

4.7 שאלות סיכום פרק

1. התיחס לקוד ציקלי (15,5) סיסטמטי הנוצר ע"י פולינום יוצר $g(x)=1+x+x^2+x^5+x^8+x^{10}$.

1.1 תאר מקודד ומפענח

1.2 תאר את תהליך הקדוד של מילת המידע הנתונה ע"י הפולינום $m(x)=1+x^2+x^4$

1.3 האם המילה שנקלטת הנתונה ביצוג פולינומיאלי ע"י $u(x)=1+x^4+x^6+x^8+x^{14}$

היא מילה חוקית? מה המלה המפוענחת עבור מילה נקלטת זאת.

2. התיחס לקוד ציקלי (15,5) סיסטמטי הנוצר ע"י פולינום יוצר $g(x)=1+x+x^2+x^5+x^8+x^{10}$.

2.1 תאר מקודד ומפענח

2.2 תאר את תהליך הקדוד של מילת המידע הנתונה ע"י הפולינום $m(x)=1+x^2+x^4$

2.3 האם המילה שנקלטת הנתונה ביצוג פולינומיאלי ע"י $u(x)=1+x^4+x^6+x^8+x^{14}$

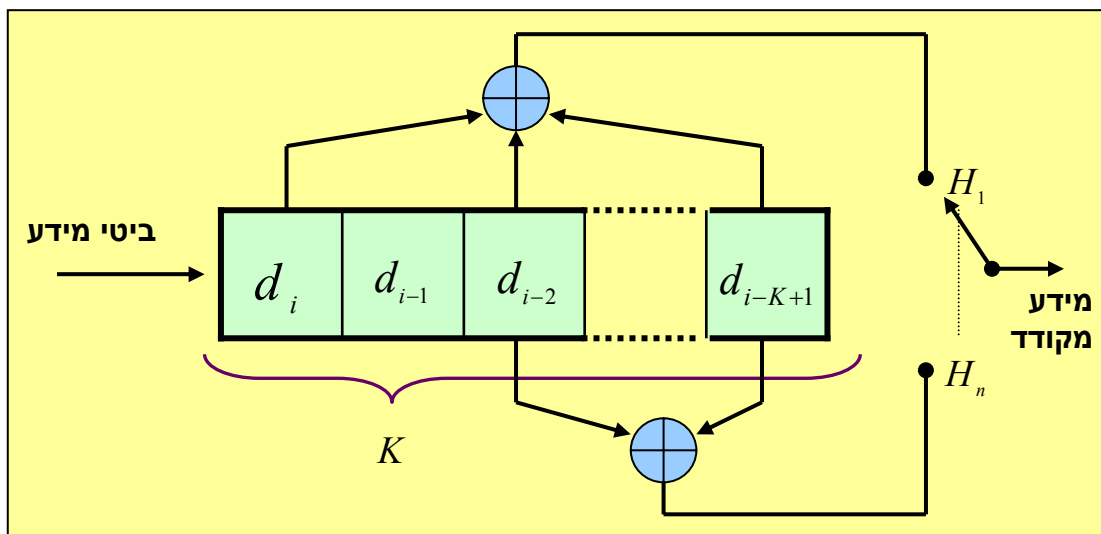
היא מילה חוקית? מה המלה המפוענחת עבור מילה נקלטת זאת.

5 קודי קונבולוציה

קודי קונבולוציה התחילו להיות שימושיים לאחר המצאת אלגוריתם ויטרבי לפענוח בסביבות 1970. כיום הם הנפוצים ביותר עם מיליארד מקודדים ומפענחים בכל טלפון סלולארי מהדור השני והשלישי. באופן עקרוני הקידוד אינו נעשה על בלוקים של מידע כמו בקודי בלוקים אלא נעשה על רצף של מידע. כאשר המידע מחולק לבלוקים ניתן לבצע הליך של התחול ושל סיום לשם הפיכת קוד קונבולוציה לקידוד בלוקים.

5.1 קידוד ביחס $\frac{1}{n}$

ביטי המידע נכנסים לרגיסטר באורך K . עבור כל ביט שנכנס מחושבים n ערכי מוצא כל אחד הינו XOR של צרף K הערכים האחרונים של המידע.



איור 5.1 מקודד קונבולוציה $\frac{1}{n}$

K -Constraint Length
 $\mathcal{G} = K - 1$ - אורך הזיכרון

במערכות יישומיות: $K = 3, \dots, 9$, ככל ש- K יותר גדול הביצועים טובים יותר. מספר הכניסות לכל XOR יכול להיות שונה.

מה שכן חייבים זה שימוש לפחות פעם אחת בבלוק d_i ובבלוק d_{i-K+1} וזאת על מנת שינוצל אורך הזיכרון K .

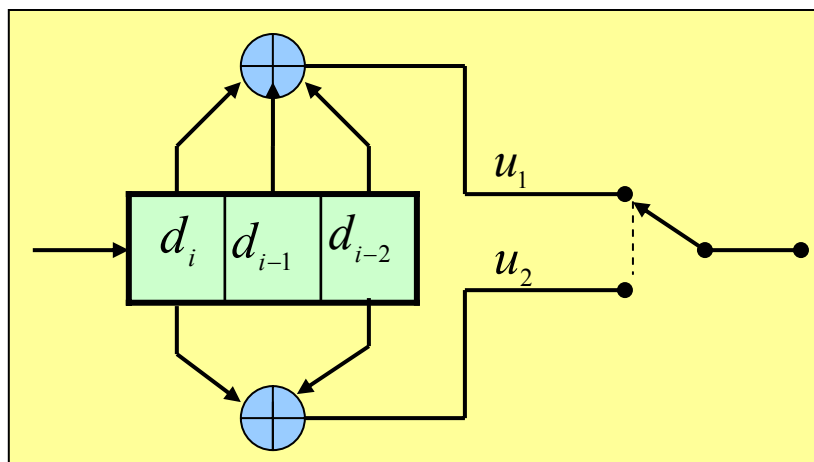
יחס הקידוד יהיה: ביט הנכנס לעומת כמות ה XORs שיוצאים כלומר $R_c = \frac{1}{n}$.

המעגל יכול לשמש ליצירת יחס קידוד $R_c = \frac{k}{n}$, על ידי הכנסת k ביטים בו זמנית

והוצאת n היציאות אך לא משתמשים בשיטה זו ליצירת יחס קידוד $\frac{k}{n}$ אלא בשיטת הניקוב שתוסבר בהמשך.

דוגמא 5.1 :

נתון $K = 3, R_c = \frac{1}{2}$



איור 5.2 מקודד קונבולוציה בעל $K = 3$ יחס קידוד $\frac{1}{2}$

בדרך כלל מתחילים ברגיסטר מאופס (000)

	כניסה	מצב קודם	מצב הבא	יציאות	
				u_1	u_2
1	1	00	10	1	1
2	0	10	01	1	0
3	1	01	10	0	0
4	1	10	11	0	1
מידע	0	11	01	0	1
סיום	0	01	00	1	1

טבלה 5.1 פרוט הכנסת מידע לרגיסטר ומוצא המקודד

ישנם שני אופני עבודה:

1. חצי אינסופי - מאפסים את הרגיסטר, ועבור כל ביט בכניסה מוציאים n ביטי יציאה עד אין סוף.
2. סופי - מאפסים את הרגיסטר, ועבור כל ביט בכניסה מוציאים n ביטי יציאה, לאחר סיום ביטי המידע, ממשיכים להכניס אפסים לרגיסטר ולשדר כל עוד יש מידע ברגיסטר.
בסוף השידור מוסיפים $K - 1$ ביטי איפוס.

שאלה: עבור m ביטי מידע ו $R_c = \frac{1}{n}$, כמה ביטים מקודדים מוציאים עבור אופן עבודה של מידע סופי?

תשובה: מספר הביטים המקודדים $= (m + K - 1) \cdot n$.

עבור הודעה באורך m אפשר לסיים עם $K - 1$ אפסים את הקידוד ואז משדרים: $(m + K - 1) \cdot \frac{1}{R_c}$, ואז בעצם יחס הקידוד R_c האפקטיבי קטן במקצת.

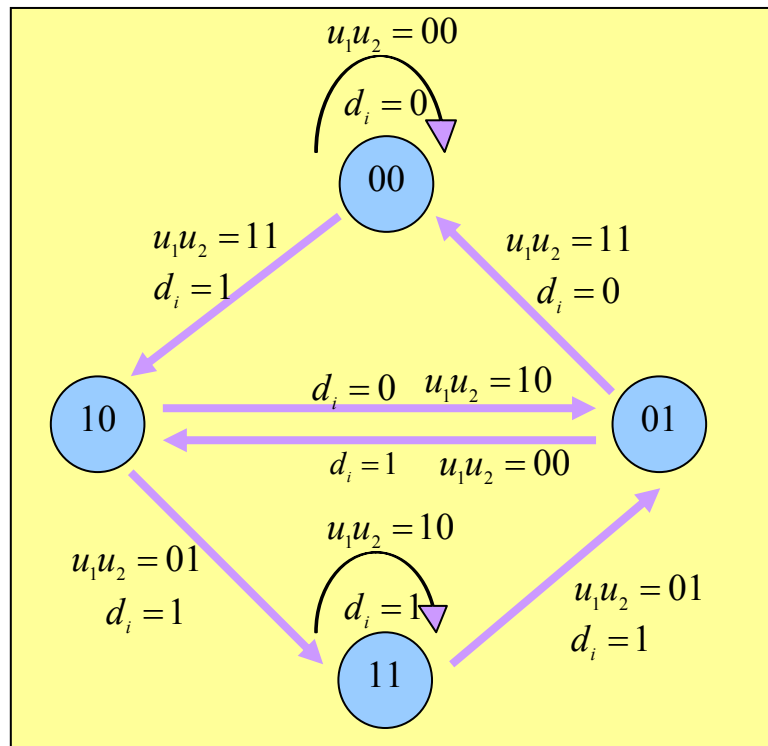
$$R_{eff} = \frac{m}{(m + K - 1) \cdot \frac{1}{R_c}} = R_c \cdot \frac{m}{m + K - 1}$$

יחס קידוד אפקטיבי: R_{eff}

ניתן לראות שכאשר $m \Rightarrow \infty$ מקבלים בעצם R_c

5.2 תיאור הקוד בעזרת דיאגרמת מצבים

אם נתייחס למקודד כמכונת מצבים אזי מספר המצבים הוא 2^{K-1} . כי ברגיסטר של K תאים התא הראשון הוא אות כניסה ו $K - 1$ האחרים הם זיכרון.

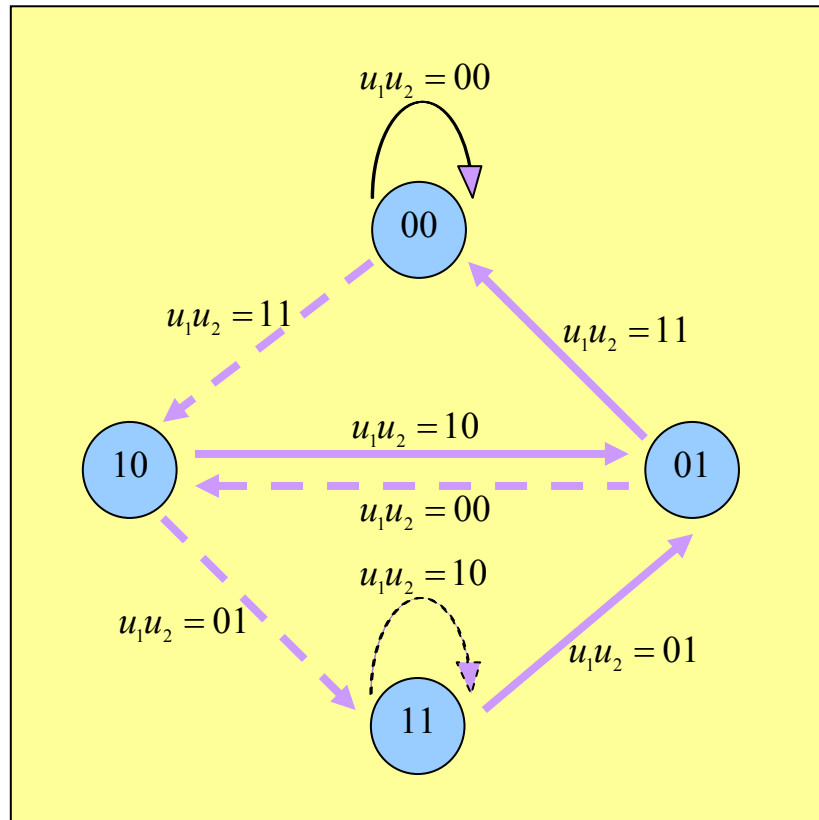


איור 5.3 דיאגרמת מצבים למקודד קונבולוציה בעל $K = 3$

נראה הצגה נוספת של דיאגרמת המצבים כאשר נגדיר את d_i על ידי שני מצבים:

$d_i = 0$ יתואר על ידי קו רציף.

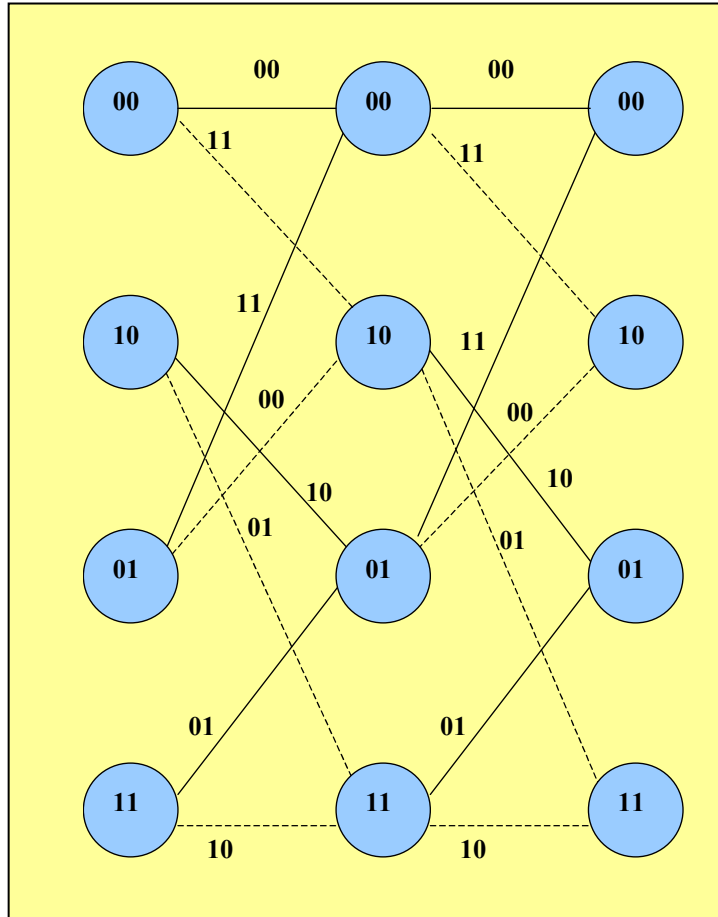
$d_i = 1$ יתואר על ידי קו מקווקו.



איור 5.4 דיאגרמת מצבים למקודד קונבולוציה בעל $K = 3$ ע"י הבדלת d_i

5.3 תיאור הקוד בעזרת טרליס

הוספת מימד הזמן לדיאגרמת מצבים הופכת אותה לדיאגרמת טרליס.



איור 5.5 הוספת מימד הזמן לדיאגרמת מצבים – טרליס

באיור 5.5 מתואר קטע של טרליס.

ישנם: 2^{K-1} מצבים.
 2^K ענפים (branch). המחברים מצבים עוקבים.
 מכל מצב יוצאים שני ענפים ולכל מצב נכנסים שני ענפים.
 לכל ענף יש סימן של ביט המידע וביטי המוצא שמתאימים לו.
 עבור מוד עבודה סופי מתאים טרליס סופי.

$$R_c = \frac{1}{3} \quad K = 9$$

לדוגמא נתון

$$2^{K-1} = 2^{9-1} = 2^8 = 256$$

מספר המצבים:

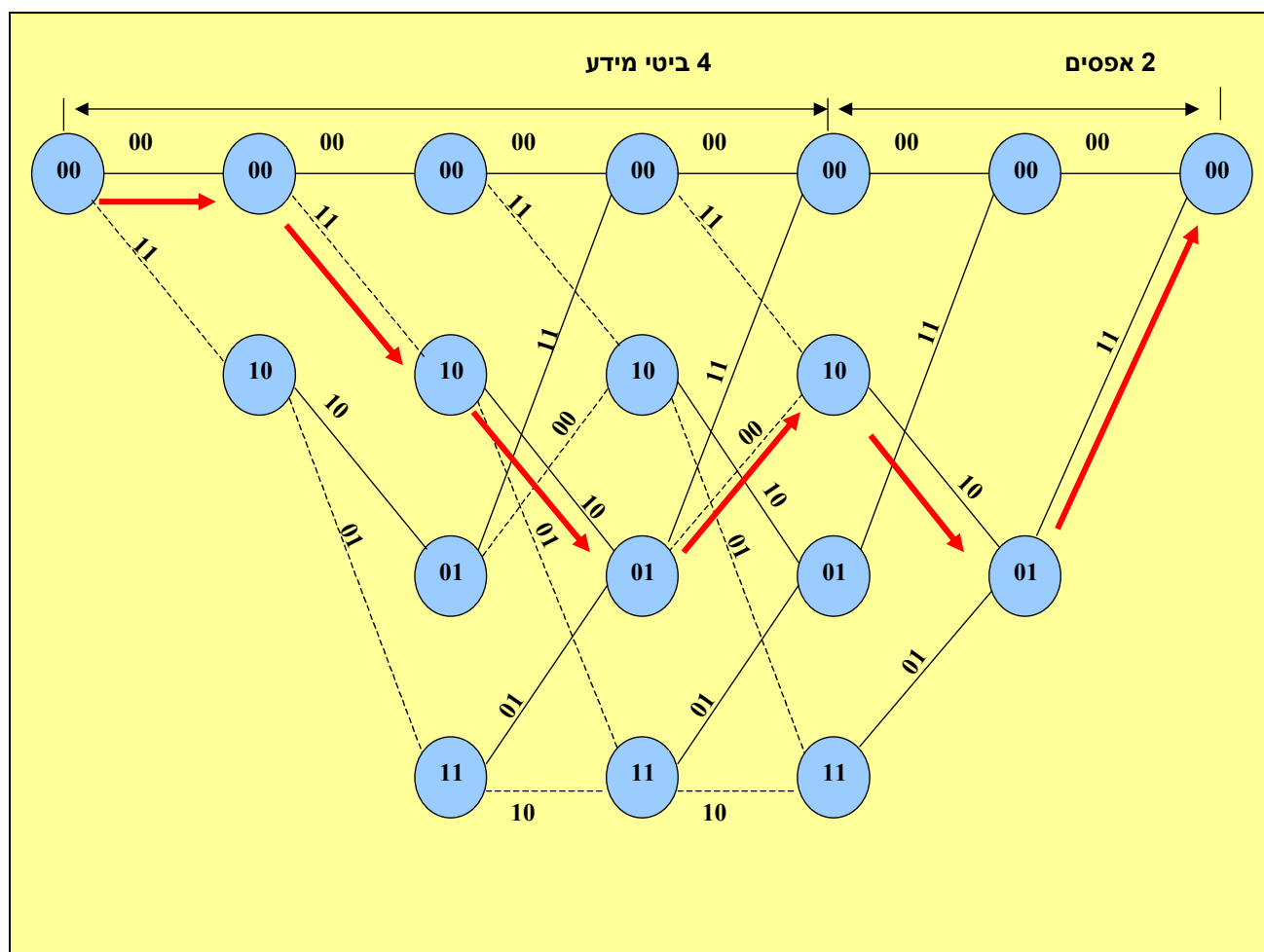
$$2^K = 2^9 = 512$$

מספר הענפים:

$$R_c = \frac{1}{n} = \frac{1}{3} \Rightarrow n = 3 \quad 2^n = 2^3 = 8$$

כמה ענפים שונים ישנם באות המוצא:

באיור 5.6 מתואר טרליס סופי עבור מקודד קונבולוציה $K = 3$, מדוגמא 5.1 להודעה באורך של 4 ביטי מידע.



איור 5.6 שרטוט טרליס סופי עבור מקודד קונבולוציה $K = 3$ והודעה באורך 4 ביטים.

המסלול **האדום** הינו המסלול המתאים למידע 0101. על פי מעקב מסלול זה ניתן לקבל את המידע המקודד המשודר: 001110001011. **מסלול (Path)** – אוסף של ענפים ברצף. אוסף המצבים והענפים המתארים סדרת מידע משודרת.

5.4 פענוח

בתהליך הקליטה יש למצוא את המסלול שמתאים לאות המשודר בעזרת האות הנקלט. תהליך הקליטה כולל בתוכו סדרה של החלטות המופקות על ידי גלאי האפנון ומעוברת למפענח. לכל אחד מהביטים המקודדים המשודרים מותאמת בקליטה החלטה. ההחלטה יכולה להיות קשה Hard Decision או רכה Soft Decision כפי שיוסבר בסעיף 5.6. המסלול שנבחר בתהליך הפענוח הוא המסלול המפוענח והוא הקרוב ביותר לאות הנקלט. לשם מדידת מידת הקרוב או הריחוק של מסלול לאות הנקלט מוגדרות מטריקות.

מטריקה של ענף (Branch Matric) היא מידת הקרוב או הריחוק של ענף לאות הנקלט.

מטריקה של מסלול (Path Matric) - סכום המטריקות של הענפים לאורך המסלול.

כדי לדעת מהו המסלול הקצר ביותר אפשר לחשב את כל מטריקות המסלולים, כלומר K^n מסלולים ולבחור את המסלול עם המטריקה הטובה ביותר, אבל ישנה דרך אשר בה ניתן לבדוק מהו המסלול הקצר ביותר בפחות חישובים. בתהליך הפענוח צריך למצוא את המסלול עם המטריקה הטובה ביותר כלומר הקצרה ביותר אם המטריקה מתארת ריחוק או הארוכה ביותר אם המטריקה מציינת קירבה.

5.5 פענוח בעזרת אלגוריתם ויטרבי

אלגוריתם למציאת מסלול אופטימאלי עבור טרליס בכמות חישובים של תלות ליניארית עם כמות ביטי המידע.

אלגוריתם ויטרבי הוא רקורסיבי ומתבצע תוך התקדמות בשלבים משלב $i-1$ לשלב i . כל פעם מחשבים עבור כל אחד מהמצבים בשלב i את מטריקת המצב ואת המסלול השורד של המצב כאשר :

המסלול השורד – הינו המסלול הטוב ביותר המגיע למצב הזה ניתן לבנות אותו כהיסטוריה.

היסטוריה – תיאור של המסלול הנבחר. זהו בעצם רצף המצבים או רצף הענפים או רצף ההחלטות להיכן לפנות.

מטריקה של מצב (State Matric) – מטריקה של המסלול השורד, סכום המטריקות של הענפים שהובילו למצב זה (מטריקה של המסלול הנבחר להגעה למצב).

נראה תחילה אלגוריתם ויטרבי עבור גלאי אפנון עם החלטות קשות ואחר כך נראה השפעת החלטות רכות.

דוגמא 5.2 : אלגוריתם ויטרבי לפי החלטה קשה

המפענח מקבל מהגלאי אות HD ואנו רוצים לדעת מהו המידע.
 נניח קוד קונבולוציה $K = 3$ $m = 4$.
 לפי טרליס סופי שודרו: $(m + K - 1) \cdot n = (4 + 3 - 1) \cdot 2 = 12$
 לפי החלטה קשה נקלטו 12 ביטים מקודדים ויתכן שבחלקם יש שגיאות.

ראינו כי עבור מידע 0101 עלינו לשדר את המידע המקודד הבא : 001110001011
 נעסוק בדוגמא בה המפענח קולט מידע בעל שתי שגיאות.

שלבי פתרון:

1. המידע הנקלט

001111011011

↑
מיקום השגיאות

באיור 5.8 משורטט טרליס ומעליו מצוינים ערכי הביטים שנקלטו בזוגות, כל זוג מעל ענף.
 המטרה היא למצוא בין כל המסלולים בטרליס את המסלול הכי קרוב, "דומה" למסלול הנקלט. המסלול שיבחר הוא שייתן לנו את המידע המפוענח.

2. נחשב מטריקה של כל ענף. על כל ענף נסמן את המרחק בין מה שמסומן על הענף לבין המידע הנקלט. חישוב המרחק נקרא מטריקת המינג (מרחק המינג)

נקלט $r_1, r_2 \in 0,1$

ענף $u_1, u_2 \in 0,1$

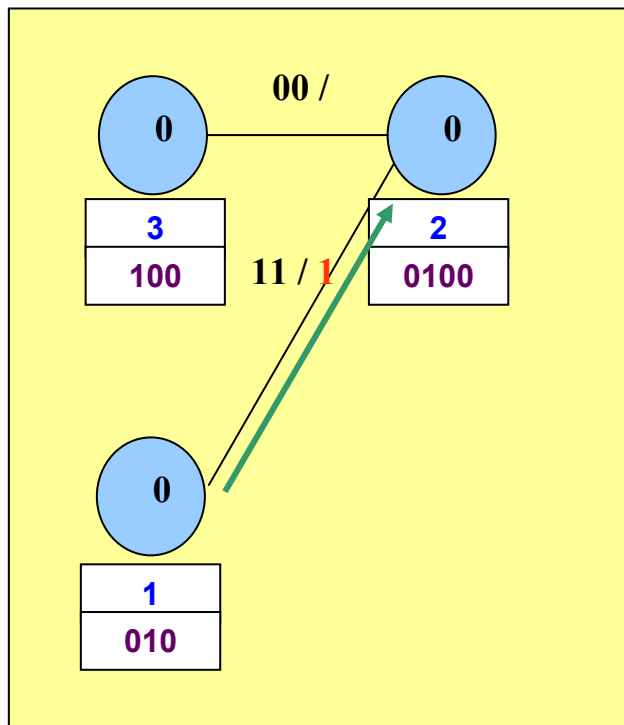
$r_1 \oplus u_1 + r_2 \oplus u_2$

אנו עובדים ב HD (החלטה קשה) לכן מקבלים $r_1 = "1", r_2 = "0"$.
 כאשר המידע הנקלט 10 , המרחקים של הענפים השונים יהיו:

מרחק המינג	מטריקה של הענף
1	00
2	01
0	10
1	11

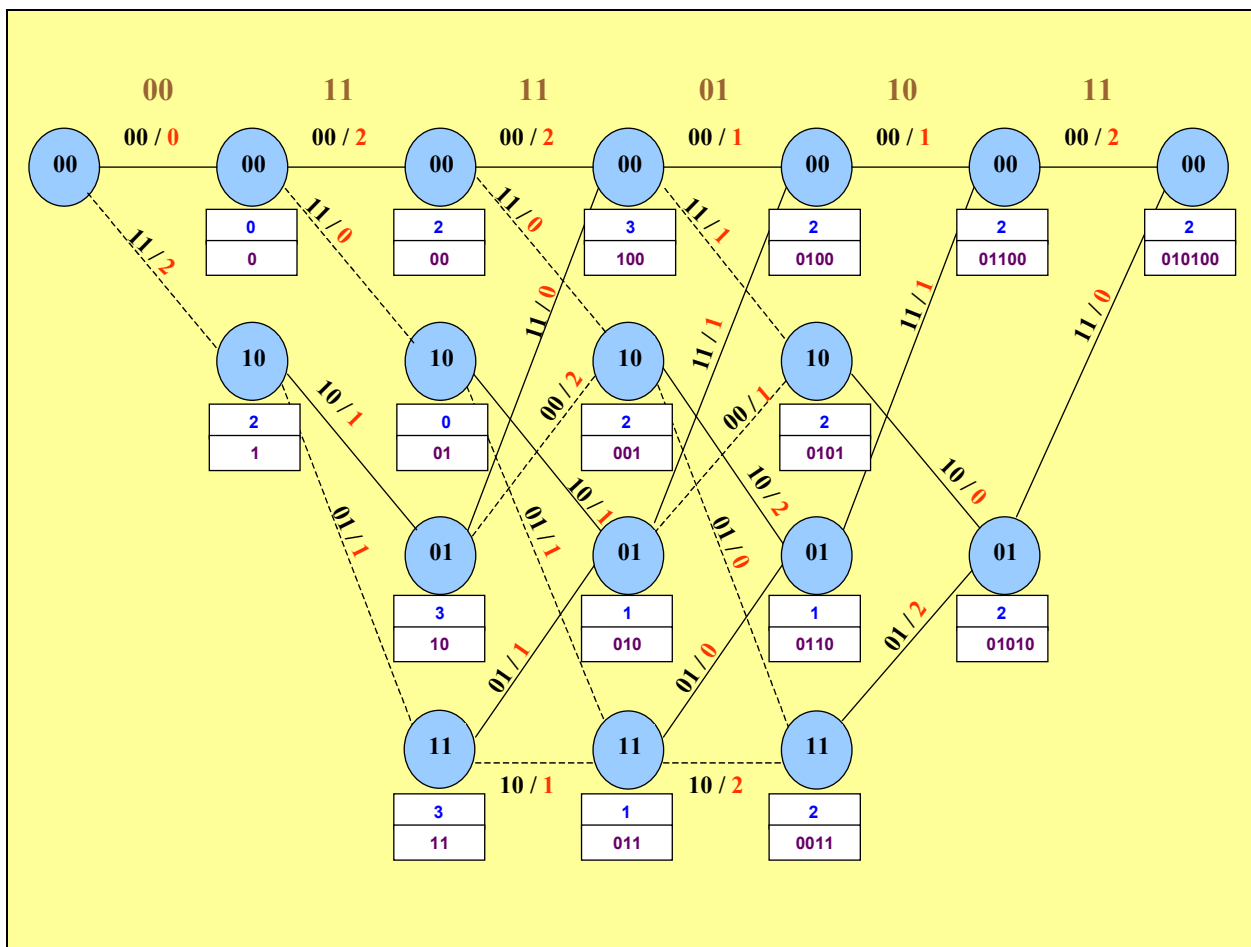
3. חישוב המסלול השורד ומטריקת מצב בעזרת טכניקת ACS = Add Compare Select (חבר, השווה, בחר). עבור שני ענפים המגיעים למצב מסוים מחברים את מטריקת הענף עם מטריקת המצב הקודם, משווים (Compare) בין שני הסכומים ובחרים (Select) הטוב ביותר (קטן ביותר עבור HD) ועל פיו בוחרים מסלול שורד.
 מטריקת המצב שווה לסכום שבחרנו.
 המסלול השורד זהו המסלול השורד של המצב הקודם ועוד ביט מידע של הענף הנבחר.

ניקח לדוגמא מצב בו מגיעים שני ענפים למצב הבא (**בצבע תכלת**) לכל ענף יש מטריקת מצב קודמת (**בתוך הריבוע בצבע כחול**) נחבר כל מטריקת מצב קודמת עם מרחק המינג של כל ענף (**בצבע אדום על הענף**).
המסלול המוביל למצב בעל הסכום הנמוך ביותר, יבחר (**בצבע ירוק**).



איור 5.7 חישוב מטריקות

4. בסיום מגיעים למצב האחרון. עבור טרליס סופי יש מצב סופי אחד בלבד. המסלול השורד הסופי כולל את המידע ועוד $K - 1$ אפסים שניתן לזרוק.



איור 5.8 טרליס הפענוח של דוגמא 5.2 עם החלטה קשה

לכל מצב מחושב:

מסלול שורד

מטריצת מצב

מרחק המינג מסומן על כל ענף.

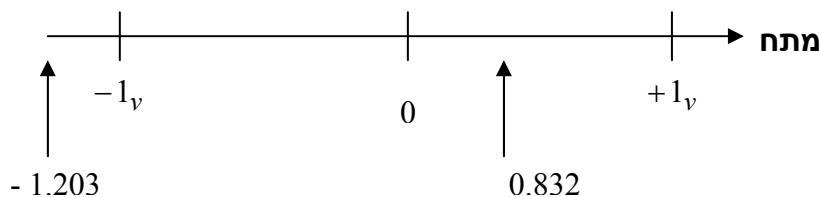
5.6 פענוח עם החלטות רכות

נתייחס להחלטה רכה ללא כימות ועם כימות. נראה חישוב מטריקה וכמו כן חישוב מופשט. בסעיף זה נראה מטריקות ענף נוספות המתאימות להחלטה רכה.

המידע המקודד עובר בשידור המרה:

$$\begin{aligned} "0" &\longrightarrow -1_v \\ "1" &\longrightarrow 1_v \end{aligned}$$

החלטה רכה SD (Soft Decision) אידיאלית - המתח עובר מהגלאי למפענח ללא חלוקה לרמות מתח:



עבור קוד ביחס קידוד $\frac{1}{2}$ עבור שני הביטים המקודדים u_1, u_2 נקלטים r_1, r_2 שהינם מספרים ממשיים המתארים את ההחלטה הרכה מהגלאי.

מרחק אוקלידי

במצב של החלטה רכה SD המרחק יהיה שווה ל:

$$\text{מרחק אוקלידי בריבוע} = (r_1 - u'_1)^2 + (r_2 - u'_2)^2$$

כאשר הערכים של u' הם בעצם: $"0" \longrightarrow -1_v$, $"1" \longrightarrow 1_v$

נניח שקלטנו שני ביטים: $r_1 = 0.832$, $r_2 = -1.203$. נציב בטבלה את המתחים r_1 ו- r_2 לפי הנוסחה ונקבל טבלה:

מטריקה של ענף	מטריקה SD מרחק אוקלידי בריבוע
00	3.4
01	8.2
10	0.07
11	4.8

למצב של 0,0 נקבל: $(0.832 - (-1))^2 + (-1.203 - (-1))^2 = 1.832^2 + 0.203^2 = 3.4$

למצב של 0,1 נקבל: $(0.832 - (-1))^2 + (-1.203 - 1)^2 = 1.832^2 + 2.203^2 = 8.2$

למצב של 1,0 נקבל: $(0.832 - 1)^2 + (-1.203 - (-1))^2 = 0.168^2 - 0.203^2 = 0.07$

למצב של 1,1 נקבל: $(0.832 - 1)^2 + (-1.203 - 1)^2 = 0.168^2 + 2.203^2 = 4.8$

מטריקת התאמה

פישוט המטריקה עבור המקרה הבינארי:

$$(r_1 - u'_1)^2 + (r_2 - u'_2)^2 =$$

נפשט את המשוואות ונקבל:

$$r_1^2 - 2r_1u'_1 + u_1'^2 + r_2^2 - 2r_2u'_2 + u_2'^2$$

- נצמצם את r_1^2 ואת r_2^2 (בכחול) מכיון שהם קבועים לכל מרחב התוצאות ועבור כל מטריצות הענפים במעבר מסוים.
- נצמצם את $u_1'^2$ ואת $u_2'^2$ (באדום) מכיון שגם במצב של -1 וגם במצב של +1 נקבל 1.
- נצמצם גם את 2 מכיון והוא קבוע (שחור).

לאחר צמצום:

$$\cancel{r_1^2} - \cancel{2r_1}u'_1 + \cancel{u_1'^2} + \cancel{r_2^2} - \cancel{2r_2}u'_2 + \cancel{u_2'^2}$$

נכפיל את התוצאה ב -1 - וכעת הפכנו סימן, לכן כעת נחפש מטריקה מקסימאלית ולא מינימאלית כמו שחיפשנו עד עכשיו. לאחר צמצום הנוסחה ולאחר ההכפלה נקבל את המטריקה המקובלת ביותר:

$$r_1u'_1 + r_2u'_2$$

לאחר הצמצום נבנה טבלה למצבים ונקבל:

מטריקה של ענף	מטריקה SD פשוט קורלצית התאמה
00	0.371
01	-2.035
10	2.035
11	-0.371

למצב של 0,0 נקבל: $0.832(-1) + (-1.203)(-1) = -0.832 + 1.203 = 0.371$

למצב של 0,1 נקבל: $0.832(-1) + (-1.203)(1) = -0.832 - 1.203 = -2.035$

למצב של 1,0 נקבל: $0.832(1) + (-1.203)(-1) = 2.035$

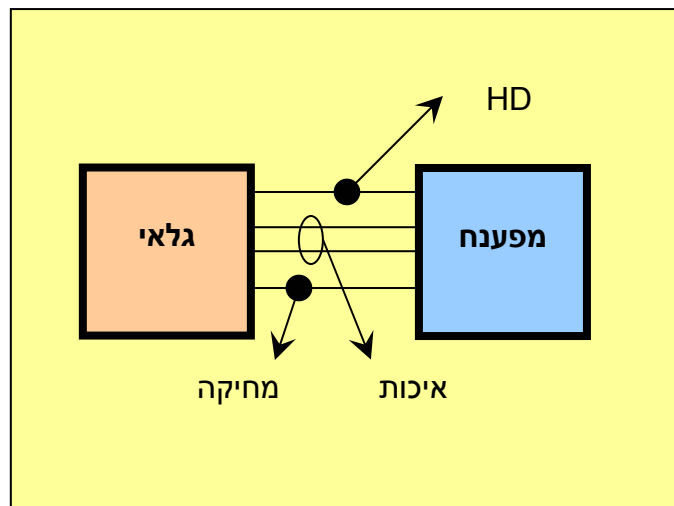
למצב של 1,1 נקבל: $0.832(1) + (-1.203)(1) = -0.371$

במקרה בינארי נשתמש במטריקה זו.

מטריקה עבור SD עם כימות

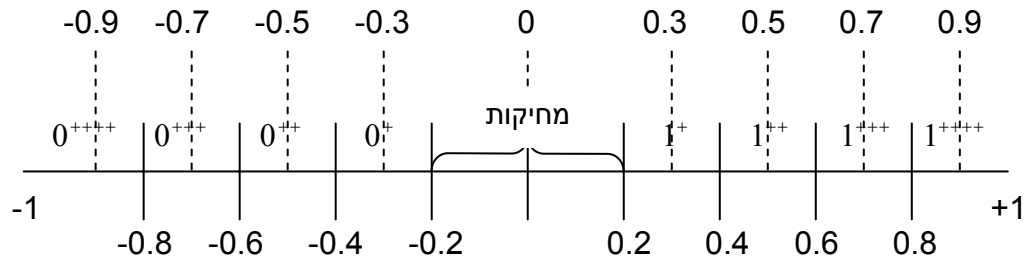
בדרך כלל הגלאי נותן את ה SD אחרי כימות במקרה זה מטריקה עבור החלטה רכה עם כימות.

החלטה רכה של 3 ביטים + מחיקה:



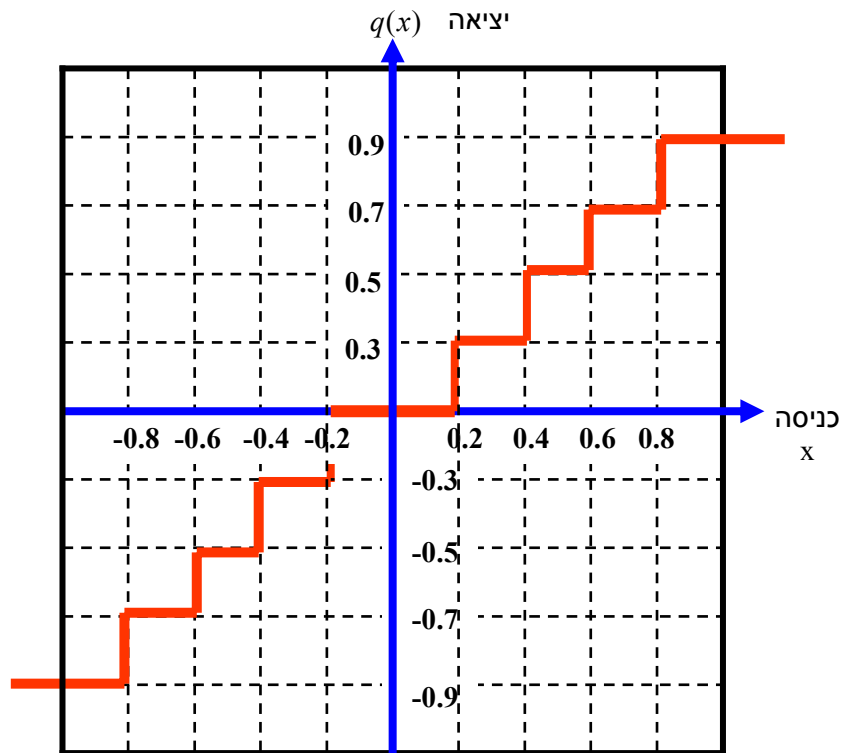
איור 5.9 החלטה רכה של 3 ביטים ומחיקה

ישנם שמונה ערכים המתוארים באיור 5.9 .

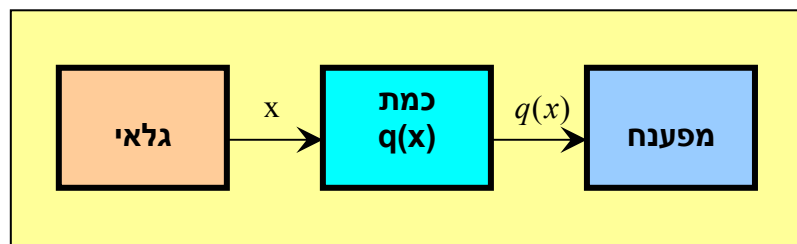


כאשר נקלוט מידע. אנחנו נקבל ערכים של מתחים. נבצע כימות לכל ערך של המתח על ידי בחירת אמצע התחום שבו קיים המתח ואת ערכים אלו נעביר הלאה. עקום כימות אופייני מתואר באיור 5.10. המתח מותאם ל 9 רמות - $0, \pm 0.3, \pm 0.5, \pm 0.7, \pm 0.9$

כימות



איור 5.10 דוגמא לעקום כימות



איור 5.11 מיקום הכמת במקלט

המטריקה שבה נשתמש היא: $q(r_1)u'_1 + q(r_2)u'_2$ כאשר $q(r_1)$ זהו הכימות של המתח r_1 שנקלט.

לשם המחשה נחזור על הדוגמא הקודמת שבה קלטנו r_1, r_2 .

$$(u'_1, u'_2) = (1, -1)$$

$$(g_1, g_2) = (-0.168, -0.203)$$

$$r_1, r_2 = (u'_1 + g_1, u'_2 + g_2) = (0.832, -1.203)$$

$$(q(r_1), q(r_2)) = (0.9, -0.9)$$

המטריקות יהיו:

מטריקה של ענף	מטריקה SD פשוט קורלצית התאמה
00	0
01	- 1.8
10	1.8
11	0

דוגמא 5.3 : פענוח ויטרבי עם מטריקת החלטה רכה עם כימות

נתון מידע: 0101

המידע מקודד ולאחר קידוד נשלח ומגיע למקלט.
ניקח את התוצאה שקיבלנו ב HD לשם השוואה.
המידע שקלטנו יהיה : 00 11 10 00

כאשר נפתור את השאלה בצורת SD נניח כי המידע הנקלט יהיה ברמות המתח כדלקמן:

$$-0.9, 0.1, 0.6, 0.5, 1.5, 0.3, -0.9, 0.4, 1, -0.7, 0.8, 0.8$$

נבנה טבלה שבה נבדוק היכן השגיאות ברמות המתח של המידע הנקלט לעומת רמות המתח של המידע ששודר:

מידע	0	1	0	1	0	0
מידע מקודד לוגי	00	11	10	00	10	11
מידע מקודד לרמות מתח	-1 -1	1 1	1 -1	-1 -1	1 -1	1 1
מידע נקלט	-0.9 0.1	0.6 0.5	1.5 0.3	-0.9 0.4	1 -0.7	0.8 0.8

ניתן לראות כי ישנן שלוש שגיאות לפי החלטה קשה

נבצע כימות לפי עקום הכימות ולאחר מכן נציין על הטריס את המידע המתקבל מהמפענח ואת מרחקי הענפים.

נכמת את המידע המתקבל במפענח:

מידע לפני כימות	מידע לאחר כימות
-0.9	-0.9
0.1	0
0.6	0.7
0.5	0.5
1.5	0.9
0.3	0.3
-0.9	-0.9
0.4	0.5
1	0.9
-0.7	-0.7
0.8	0.9
0.8	0.9

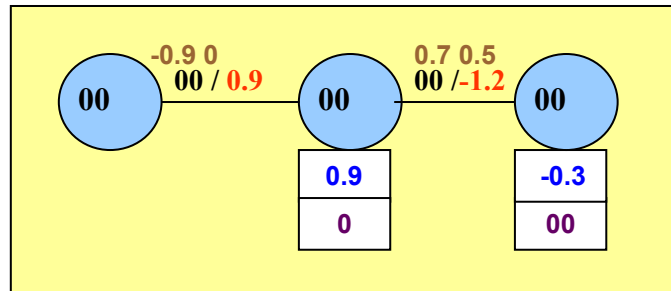
נשרטט מעל הטריס את המידע המתקבל לאחר פענוח (**בשרטוט הטריס בצבע חום**).

נכפיל מטריקות שקיבלנו בענפים ונציין את המרחק על כל ענף (**בשרטוט הטריס בצבע אדום**).

לדוגמא: על הענף הראשון בשרטוט הטריס המידע הינו 00, בהמרה לרמות מתח נקבל $-1 - 1$. נבצע חישוב לעומת המידע הנקלט על ידי הנוסחא $r_1 \cdot u_1 + r_2 \cdot u_2$.
 $(-0.9 \cdot -1) + (0 \cdot -1) = 0.9 + 0 = 0.9$

לאחר מכן מתחת לכל עיגול "מצב הבא" (**בשרטוט הטריס בצבע תכלת**) נבנה קובייה ובה נציין את המצבים הנכנסים (מטריקה) (**בשרטוט הטריס בצבע כחול**) והיסטוריה (חיבור מסלול שורד) (**בשרטוט הטריס בצבע סגול**).

ננתח את התחלת הטרליס בתור דוגמא: במצב הבא הראשון (כדור מצד ימין) כאשר נחשב את המרחק, הסכום שנקבל (**בשרטוט בצבע אדום**) יהיה 0.9. במצב ההתחלתי (הכדור מצד שמאל) סכום המרחקים הינו 0 (התחלת הטרליס) לכן נחבר $0 + 0.9$ ונקבל 0.9, ערך זה מצוין בקובייה העליונה מאחר והוא מצב נכנס (**בשרטוט בצבע כחול**). ההיסטוריה נקבעת על פי ערך ביט המידע בענף ("0" או "1"), בענף זה ביט המידע הינו "0" ולכן ההיסטוריה תהיה "0" אותה נציין בקובייה התחתונה (**בשרטוט בצבע בסגול**).



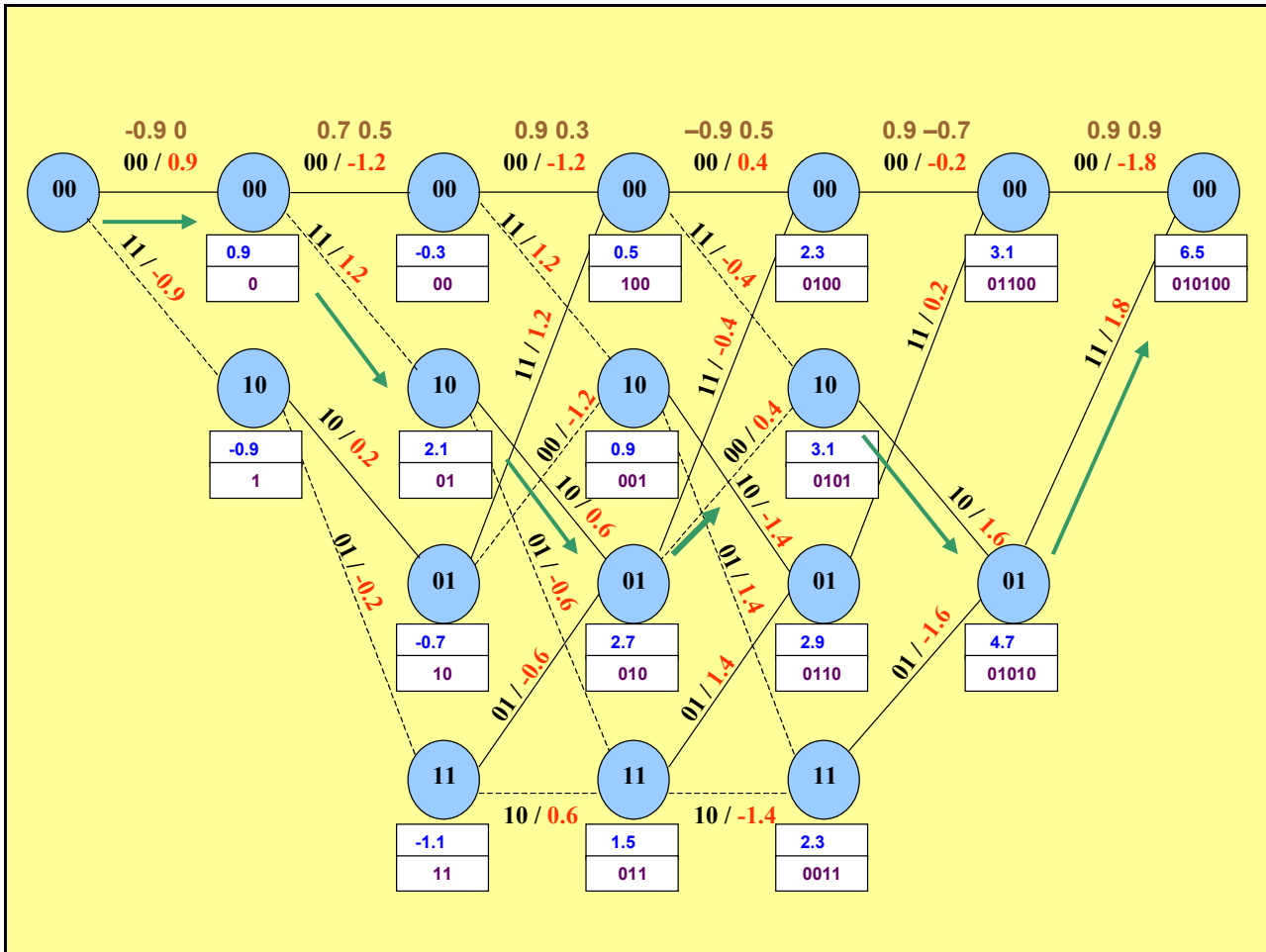
במצב הבא כאשר נכניס ביט מידע "0" נחשב את המרחק בענף ונקבל -1.2 בחיבור עם מצב נכנס קודם (סכום המרחקים הקודמים) נקבל $-0.3 = 0.9 + (-1.2)$ ההיסטוריה תהיה 00.

מעיון במצב הרביעי (איור 5.12) ניתן לראות שישנם שני מצבים נכנסים, לכן יש צורך בביצוע A.C.S אך יש לשים לב שבהחלטה רכה אנו בוחרים את המסלול בעל הערך הגבוה ביותר (בהחלטה קשה לקחנו את הערך הקטן). לכן נבחר בכניסה התחתונה מכיוון שסכום המרחקים בו גבוה יותר מהכניסה העליונה.

נשרטט באיור 5.12 את הטרליס הסופי לאחר כל החישובים ונציין בו את המסלול השורד.

הטרליס הסופי שנקבל לאחר החישובים מתואר באיור 5.12:

נציין את המסלול השורד (בשרטוט בצבע ירוק).



איור 5.12 שרטוט טרליס סופי וציון מסלול שורד של דוגמא 5.3

5.7 ניקוב - Puncturing

לצורך יצירת יחס קידוד יותר גדול ניתן להשתמש בקוד מנוקב – Punctured code. בשיטה זו לא משדרים חלק מוסכם מהביטים המקודדים. במקלט משחזרים את הביטים שנוקבו על ידי מחיקות. נהוג לבצע ניקוב רק בקודי קונבולוציה וטורבו.

זו הטכניקה המקובלת היחידה ליצירת יחסי קידוד $\frac{k}{n}$ ולא $\frac{1}{n}$ (k שונה מ-1).

הסיבה היא פשטות - משתמשים באותו משדר ואותו מקלט עבור כל יחסי הקידוד.

כאשר נתון יחס קידוד בסיסי $R_c = \frac{1}{2}$ ניתן להגיע על ידי ניקוב ליחס קידוד $\frac{1}{2} < R_c < 1$.

כאשר נתון יחס קידוד בסיסי $R_c = \frac{1}{3}$ ניתן להגיע על ידי ניקוב ליחס קידוד $\frac{1}{3} < R_c < 1$.

כאשר נתון יחס קידוד בסיסי $R_c = \frac{1}{3}$ וגם $R_c = \frac{1}{2}$ ניתן להגיע על ידי ניקוב

ליחס קידוד $\frac{1}{3} < R_c < \frac{1}{2}$ ו- $\frac{1}{2} < R_c < 1$.

דוגמא לביצוע ניקוב במידע בעל 4 ביטים ועוד שני ביטי איפוס רגיסטר סה"כ 6 ביטים.

עקב מקודד הקונבולוציה נקבל לאחר קידוד: $[m + (K - 1)] \frac{1}{R_c} = [6 + 2]2 = 12$

ישנה דרישה להגיע ליחס קידוד $\frac{3}{4}$. כלומר על כל 3 ביטי מידע לאחר קידודם נבצע ניקוב בכדי להגיע לשליחת 4 ביטים בלבד.

המידע לאחר הוספת אפסים הינו שישה ביטים:

1	2	3	4	5	6
---	---	---	---	---	---

המידע לאחר מקודד קונבולוציה ביחס קידוד $\frac{1}{2}$ הם 12 ביטים מקודדים.

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

לאחר קידוד מידע ביחס $\frac{1}{2}$ קיבלנו על כל 3 ביטים 6 ביטים מקודדים, לכן כדי להגיע ל-4 ביטים עלינו לנקב 2 מתוכם בשיטה מוסכמת.

המשדר שלנו מנקב כל ביט שלישי לכן המידע המנוקב יהיה:

1	2	X	4	5	X	7	8	X	10	11	X
---	---	---	---	---	---	---	---	---	----	----	---

כאשר X מסמן אי שידור.

המידע שישודר לאחר ניקוב ביחס קידוד $\frac{3}{4}$:

1	2	4	5	7	8	10	11
---	---	---	---	---	---	----	----

ניתן לראות כי כמות הביטים המשודרת הינה 8.

המקלט צריך לשחזר את המידע ליחס קידוד $\frac{1}{2}$. לכן המקלט יעבוד כמו מנקב אך במקום לנקב יכניס מחיקות (0 volt - מתח). המידע יראה כך:

1	2	0V	4	5	0V	7	8	0V	10	11	0V
---	---	----	---	---	----	---	---	----	----	----	----

סיכרון

יש צורך בסנכרון לקבוצות. בדוגמא גודל קבוצה הוא 3 bit כיצד מסנכרנים את המידע? נתאר שלוש אפשרויות.

- דואגים להכניס Header או Preamble לפני המידע בכדי לסמנו או לחילופין משדרים בתחילה מילה מוסכמת וידוע מראש גם למקלט ולמשדר.
- בודקים כל אחת מ- n האפשרויות (ההיפותזות). עבור אפשרות שאינה נכונה נקבל המון שגיאות.
- מכונה לספירת שגיאות (מדידת כמות שגיאות)
 - מקודדים מחדש את המידע המפוענח
 - משווים עם מה שנקלט.

דוגמא 5.4 :

מעבר יחס קידוד על ידי ניקוב מ $\frac{1}{2}$ ל $\frac{7}{8}$

מידע:

1	2	3	4	5	6	7
---	---	---	---	---	---	---

מידע מקודד ביחס $\frac{1}{2}$:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	----	----	----	----	----

מידע מקודד עם ניקוב:

1	X	3	X	5	X	7	8	X	10	X	12	X	14
---	---	---	---	---	---	---	---	---	----	---	----	---	----

מידע מקודד ביחס $\frac{7}{8}$

1	3	5	7	8	10	12	14
---	---	---	---	---	----	----	----

פעולת המקלט הוספת מחיקות:

1	0V	3	0V	5	0V	7	8	0V	10	0V	12	0V	14
---	----	---	----	---	----	---	---	----	----	----	----	----	----

דוגמא 5.5 :

שליחת מידע ע"י ניקוב ליחס קידוד של $\frac{3}{4}$ בתצורת SD

המידע שאנו רוצים לשדר הוא 0101

המידע המקודד לפני תהליך הניקוב הוא :

00 11 10 00 10 11

נבצע ניקוב במידע המקודד בכדי להגיע ליחס קידוד $\frac{3}{4}$ (ננקב כל ביט מידע שלישי) :

00 **X**1 1**X** 00 **X**0 1 **X**

המידע המנוקב לשידור :

00 11 00 01

נמיר את המידע לשידור בתצורת SD (± 1) :

-1 -1 , 1 1 , -1 -1 , -1 1

האות נשלח ונקלט (עם רעש) בתצורה הבאה:

-0.9 0.1 0.5 0.9 -0.9 0.55 -0.7 1.5

נבצע כימות לפי עקום הכימות.

לאחר כימות נקבל את האות:

-0.9 0 0.5 0.9 -0.9 0.5 -0.7 0.9

כעת יש להוסיף מחיקות במקום הניקובים שביצענו במשדר ולכן לאחר כל שני ביטים נוסיף מחיקה.

האות לאחר הוספת אפסים יהיה:

-0.9 0 **0** 0.5 0.9 **0** -0.9 0.5 **0** -0.7 0.9 **0**

נשרטט מעל הטרליס את המידע המתקבל לאחר הכימות והוספת האפסים.
(**בשרטוט הטרליס בצבע חום**)

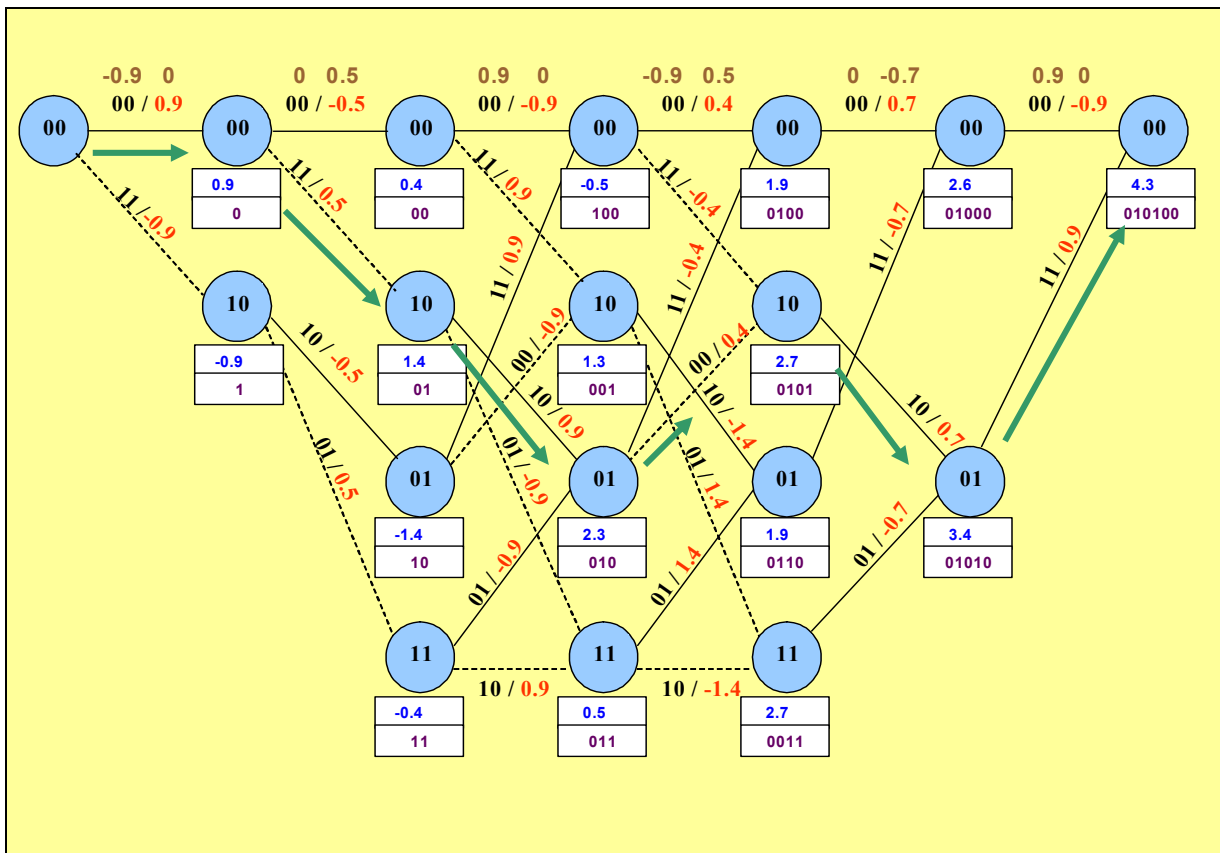
נכפיל מטריקות שקיבלנו בכל ענף וענף ונציין את המרחק על כל ענף.
(**בשרטוט הטרליס בצבע אדום**)

לדוגמא: על הענף הראשון בשרטוט הטרליס המידע הינו 00, בהמרה לרמות מתח נקבל $-1 -1$. נבצע חישוב לעומת המידע הנקלט על ידי הנוסחה $r_1 \cdot u_1 + r_2 \cdot u_2$

$$(-0.9 \cdot -1) + (0 \cdot -1) = 0.9 + 0 = 0.9$$

לאחר מכן מתחת לכל עיגול "מצב הבא" (בשרטוט הטרליס בצבע תכלת) נבנה קובייה ובה נציין את המצבים הנכנסים (מטריקה) (בשרטוט הטרליס בצבע כחול) והיסטוריה (חיבור מסלול שורד) (בשרטוט הטרליס בצבע סגול).
לסיום נציין את המסלול השורד (בשרטוט הטרליס בצבע ירוק).

שרטוט הטרליס הסופי שנקבל לאחר כל החישובים הוא:



איור 5.13 שרטוט טרליס סופי וציון מסלול שורד של דוגמא 5.5

תיקו – שיוויון

יש לשים לב שבדוגמא הנ"ל במצב של כניסת ביט מידע שלישי (עמודת כדורים רביעית משמאל) שני המצבים הקיצוניים (כדור עליון ותחתון) הם בעלי ACS אשר תוצאות שני הענפים הנכנסים שוות. במצב של שוויון יש לבחור שרירותית בנתיב נבחר ולהמשיך על פיו. במקרה שלנו המסלול השורד אינו עובר דרך מצבים אלו לכן אין להם השפעה.

5.8 שיקולים לבחירת קוד

מה היתרונות והחסרונות ביחסי קידוד שונים?

ככל שיחס הקידוד קטן יותר אז אם קצב הביטי המידע R_b וקצב ביטים מקודדים הינו $R_c = \frac{R_b}{R_c}$, ניתן לראות כי עקב כך רוחב הסרט גדל (עקב תלות יחסית ב R_{eb}) וזה לא טוב. לעומת זאת שבח הקידוד (CG) גדל יותר וזה טוב.

בטבלה נתונים שבח הקידוד והגידול ברוחב הסרט בתלות יחס הקידוד עבור החלטה רכה SD $P_b = 10^{-5}$, $K = 7$,

R_c	CG	$\left(\frac{R_{eb}}{R_b} - 1 \right) \cdot 100$
$\frac{1}{2}$	5dB	100%
$\frac{3}{4}$	4dB	33.3%
$\frac{7}{8}$	3dB	14.2%

הערות:

1. חיפוש מקסימום/ מינימום

מה המשמעות של חיפוש לפי מקסימום או לפי מינימום?
באלגוריתם ACS נבצע בחירה לפי Maximum או לפי Minimum. המטריקות שהצגנו
מטריקה המינג – מינימום
מטריקה אוקלידית – מינימום
מטריקה מפושטת - מקסימום

2. מה קורה כאשר הטריס הוא אינסופי?

כאשר משדרים מידע ארוך נקבל טריס כמעט אינסופי ואז יש כמה בעיות:

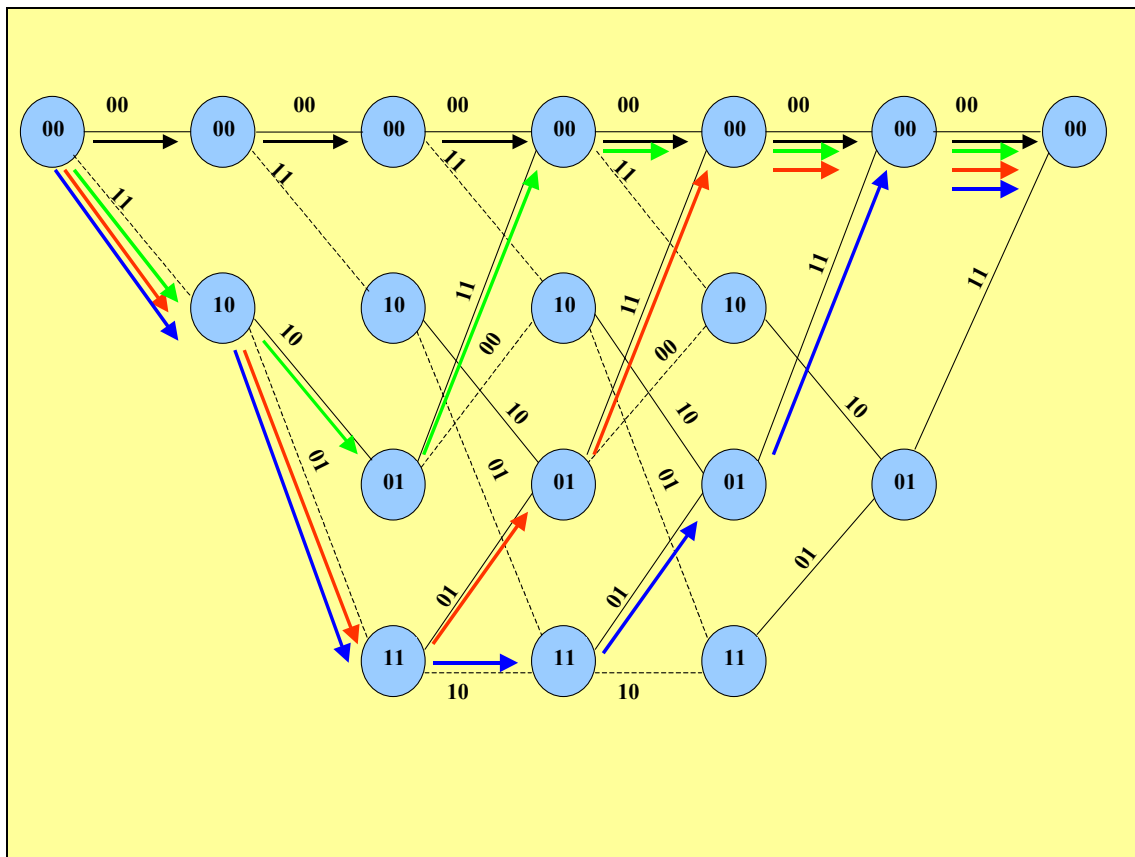
- מטריקות גדלות בלי סוף – הפתרון – מידי פעם מורידים את כל מטריקות המצבים במספר מסוים.
- החלטה סופית באין סוף הפתרון – מגבילים את ההיסטוריה (רישום של המסלול) ל – 35 $(5 \cdot 7 = 35 = 5k)$ אזי הכי ישן שזורקים אותו בהיסטוריה.
- המסלול השורד בטריס גדל לאינסוף. הפתרון- בזמן שמגיעים לביט i מחליטים לגבי ביט בזמן $i - 35$

5.9 ביצועי קוד קונבולוציה

אירוע שגיאה – Error event הינו קטע של מסלול לא נכון מרגע שהוא עוזב את המסלול הנכון ועד שהוא חוזר אליו.

המאפיין אירוע זה:

מספר ענפים – l
 מספר השגיאות – n
 מרחק – d



איור 5.14 אירועי שגיאות בקודי קונבולוציה

5.14 ננתח את איור

המסלול השחור הינו המסלול התקין.
 המסלול הירוק מסמן אירוע שגיאה ראשונה.
 המסלול האדום מסמן אירוע שגיאה שנייה.
 המסלול הכחול מסמן אירוע שגיאה שלישית.

המרחקים d הם ההבדלים בביטים המקודדים במרחק ביניהם. נחבר את כל המרחקים. לדוגמא במצב הראשון (**מסלול ירוק**) בענף הראשון המרחק הינו 2 ובענף השני גם 2 ובשלישי גם 2 לאחר מכן אין הבדל בענפים לכן נקבל $d = 6$.
 n הינו הבדל בביטי המידע כלומר מספר השגיאות. רואים שהמידע המקורי הוא 00000 והמידע עקב שגיאה הוא 10000 לכן מספר השגיאות $n = 1$.
 l שהינו מספר הענפים של השגיאה הוא 3.

ניתן לומר אם כך כי **במסלול האדום**

$$d = 6 \quad n = 2 \quad l = 4$$

ובמסלול הכחול:

$$d = 7 \quad n = 3 \quad l = 5$$

הגדרה:

המרחק הקטן ביותר של כל אירועי השגיאה עבור קוד מסוים נקרא מרחק הקוד:

$$d_f = \text{free distance} \quad \text{נסמנו:}$$

על פניו נראה בטרליס באיור 5.14 כי המסלול עם המרחק הקצר ביותר ($d_f = 5$) שייך למסלול הירוק שהוא בעל מספר הענפים הקצר ביותר, אך ישנם קודים בהם המסלול עם המרחק הקצר ביותר d_f אינם בעלי מספר הענפים הקטן ביותר.
עובדה ידועה זה שעבור $k = 7$ נקבל $d_f = 10$ מתוך טבלה 5.2 שבעמוד הבא.

k	Generators (octal)	d_f	$(a_{df+t}, \dots)$ $[b_{df+t}, \dots]$ $l = 0, 1, 2, \dots, l-1$	d_0
3	5,7	5	(1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072) [1, 4, 12, 32, 80, 192, 448, 1024, 2304, 5120, 11264, 24576, 53248, 114688, 245760, 524288, 1114112, 2359296]	1/2
4	15,17	6	(1, 3, 5, 11, 25, 55, 121, 267, 589, 1299, 2865, 6319, 13937, 30739, 67797, 149531, 329801, 727399) [2, 7, 18, 49, 130, 333, 836, 2069, 5060, 12255, 29444, 70267, 166726, 393635, 925334, 2166925, 5057286, 11767305]	1/2
5	23,35	7	(2, 3, 4, 16, 37, 68, 176, 432, 925, 2156, 5153, 11696, 26868, 62885, 145085, 334024, 774966, 1793363) [4, 12, 20, 72, 225, 500, 1324, 3680, 8967, 22270, 57403, 142234, 348830, 867106, 2134239, 5205290, 12724352, 31022962]	4/11
6	53,75	8	(1, 8, 7, 12, 48, 95, 281, 605, 1272, 3334, 7615, 18131, 43197, 99210, 237248, 559238, 1312675, 3108350) [2, 36, 32, 62, 332, 701, 2342, 5503, 12506, 36234, 88576, 225685, 574994, 1400192, 3554210, 8845154, 21841106, 54350946]	8/23
7	133,171	10	(11, 0, 38, 0, 193, 0, 1331, 0, 7275, 0, 40406, 0, 234969, 0, 1337714, 0, 7594819, 0) [36, 0, 211, 0, 1404, 0, 11633, 0, 77433, 0, 502690, 0, 3322763, 0, 21292910, 0, 134365911, 0]	4/13
8	247,371	10	(1, 6, 12, 26, 52, 132, 317, 730, 1823, 4446, 10739, 25358, 60773, 146396, 350399, 842174, 2021290, 4853474) [2, 22, 60, 148, 340, 1008, 2642, 6748, 18312, 48478, 126364, 320062, 821350, 2102864, 5335734, 13549068, 34254388, 86441848]	5/18
9	561,753	12	(11, 0, 50, 0, 286, 0, 1630, 0, 9639, 0, 55152, 0, 320782, 0, 1859184, 0, 10777264, 0) [33, 0, 281, 0, 2179, 0, 15035, 0, 105166, 0, 692330, 0, 4580007, 0, 29692894, 0, 190453145, 0]	8/27
10	1167,1545	12	(2, 8, 15, 35, 68, 170, 458, 1084, 2574, 6177, 14939, 36200, 86856, 208847, 504561, 121706, 2933502, 7066863) [14, 26, 74, 257, 496, 1378, 4122, 10832, 27988, 72209, 186920, 483102, 1234736, 3149395, 8033048, 20419644, 51688436, 130527021]	1/4
11	2335,3661	14	(21, 0, 74, 0, 454, 0, 2687, 0, 15629, 0, 90518, 0, 526556, 0, 3067758, 0, 17845415, 0) [94, 0, 463, 0, 3783, 0, 26711, 0, 181571, 0, 1207474, 0, 7919894, 0, 51390913, 0, 329342619, 0]	2/15
12	4335,5723	15	(16, 31, 44, 129, 309, 697, 1713, 4175, 10158, 24508, 58600, 141960, 343347, 826478, 1996843, 4820534, 11619637, 28039590) [76, 180, 374, 1142, 2783, 6836, 18709, 49242, 128178, 329408, 836478, 2151230, 5497355, 13931276, 35357451, 89485786, 225656685, 568414202]	14/53
13	10533,47661	16	(33, 0, 11, 0, 779, 0, 4128, 0, 24173, 0, 142500, 0, 828402, 0, 4829478, 0, 28122349, 0) [152, 0, 971, 0, 6933, 0, 45436, 0, 303435, 0, 2036131, 0, 13256560, 0, 85514159, 0, 546034284, 0]	8/33
14	21675,27123	16	(4, 17, 35, 76, 193, 454, 1047, 2624, 6138, 14944, 36179, 86640, 210568, 508233, 1225765, 2960696, 7146740, 17245991) [22, 99, 218, 608, 1724, 4404, 11108, 30438, 75942, 196714, 507723, 1289364, 3311290, 8425785, 21377872, 54168142, 136847122, 344912207]	27/103

טבלה 5.2 תחום ערכים של $a(d), b(d)$

לחישוב הסתברות שגיאה לאירוע יש לקחת בחשבון כל אפשרות לטעות לכוון מסלול אחר. יש $a_{(df)}$ מסלולים במרחק d_f מהמסלול הנכון. כנ"ל $a_{(df+1)}$ מסלולים במרחק d_{f+1} מהמסלול הנכון, וכן הלאה. השגיאה ניתנת על ידי חסם האיחוד.

לקוד עם תיקון שגיאות הסתברות שגיאה לאירוע שווה ל:

$$P_e < \sum_{d=df} a(d)P(d)$$

נוסחא זו נכונה עבור שיטת אפנון QPSK, BPSK כאשר:

d_f - מרחק הקוד

$a(d)$ - מספר האירועים במרחק d , שיוצאים ממצב מסוים

$P(d)$ - הסתברות השגיאה של אירוע במרחק d .

עבור החלטה רכה:

$$P(d) = Q\left(\sqrt{\frac{2E_s d_f}{N_0}}\right)$$

$$E_s = E_{eb} = R_c \cdot E_b$$

E_s יותר קטן מ- E_b

עבור החלטה קשה:

$$P(d) = \sum_{i=\frac{d}{2}+1}^d \binom{d}{i} \cdot p^i (1-p)^{d-i} + \frac{1}{2} \binom{d}{d/2} p^{\frac{d}{2}} (1-p)^{\frac{d}{2}}$$

כאשר d זוגי:

$$P(d) = \sum_{i=\frac{d+1}{2}}^d \binom{d}{i} \cdot p^i (1-p)^{d-i}$$

כאשר d אי זוגי:

הסתברות שגיאה לביט $\overline{P_b}$ עם קידוד

לחישוב הסתברות שגיאה לביט יש לקחת בחשבון כמה ביטים שגויים ישנם בכל אירוע. ניתן לחלק את $a(d)$, מספר המסלולים במרחק d לכאלה שגורמים שגיאה אחת $i = 1$, שתי שגיאות וכן הלאה.

$$P_b < \sum_{i=1}^{\infty} i \sum_{d=df}^{\infty} a(d,i)P(d) = \sum_{df}^{\infty} \sum_{i=1}^{\infty} ia(d,i)P(d) =$$

$$= \sum_{df}^{\infty} b(d)P(d)$$

$$b(d) \triangleq \sum_{i=1}^{\infty} ia(d,i)$$

כאשר:

$$a(d) = a(d,1) + a(d,2) + \dots = \sum_{i=1}^{\infty} a(d,i)$$

כדי לחשב את הסתברות השגיאה לביט

$$p_b = \sum_{i=1}^{\infty} i \cdot p$$

$a(d,i) \triangleq$ מספר אירועים במרחק d עם i שגיאות מידע.

i מציין את מספר השגיאות לאירוע במרחק d .

ניתן להתייחס ל- $b(d)$ כאל מספר השגיאות הכולל בכל המסלולים במרחק d .

דוגמא 5.6 :

עבור השגיאות באיור 5.14 נתון $K = 3, R_c = \frac{1}{2}$

$$P_e \leq a(5) \cdot Q \sqrt{\frac{2 \cdot 5 \cdot E_s}{N_0}}$$

$$E_s = E_{eb} = R_c \cdot E_b = \frac{1}{2} \cdot E_b$$

$$P_e \leq a(5) \cdot Q \sqrt{\frac{10 \cdot \frac{E_b}{2}}{N_0}} + a(6) \cdot Q \sqrt{\frac{2 \cdot 6 \cdot \frac{E_b}{2}}{N_0}} + a(5) \cdot Q \sqrt{\frac{2 \cdot 7 \cdot \frac{E_b}{2}}{N_0}}$$

$$a(5) = 1$$

$a(5)$ - מופיע פעם אחת

$$a(6) = 2$$

$a(6)$ - מופיע פעמיים

$$a(7) = 4$$

$a(7)$ - מופיע 4 פעמים

נציב במשוואה:

$$P_e \approx Q \sqrt{\frac{5 \cdot E_b}{N_0}} + 2Q \sqrt{\frac{6 \cdot E_b}{N_0}} + 4Q \sqrt{\frac{7 \cdot E_b}{N_0}} + \dots$$

חישוב הסתברות לביט

$$P_b \cong Q \sqrt{\frac{5 \cdot E_b}{N_0}} \quad \text{נעשה קרוב ראשוני ל- } P_b \text{ על ידי האיבר הראשון שהוא}$$

$$\text{כעת נשווה בין הסתברות שגיאה לביט עם קידוד } P_b \cong Q \sqrt{\frac{5 \cdot E_b}{N_0}} \text{ להסתברות שגיאה לביט}$$

$$P_b = Q \sqrt{\frac{2 \cdot E_b}{N_0}} \quad \text{ללא קידוד}$$

$$Q \sqrt{2 \left(\frac{E_b}{N_0} \right)_{nc}} = Q \sqrt{5 \left(\frac{E_b}{N_0} \right)_c}$$

ונקבל

$$\left(\frac{E_b}{N_0}\right)_{nc} = 2.5 \left(\frac{E_b}{N_0}\right)_c = 4dB + \frac{E_b}{N_0}_c$$

זאת אומרת שעבור SD שבח הקידוד הוא בערך 4dB.

$$p_b = 36Q \sqrt{\frac{10E_b}{N_0}} \quad \text{עבור } K = 7 \text{ נקבל בקרוב של האיבר הראשון}$$

עבור $\frac{E_b}{N_0}$ גדול ניתן להחליף את המקדם 36 ב-1 עם שגיאה קטנה ביותר בעקום.
עם נחזור על ההשוואה:

$$Q \sqrt{2 \left(\frac{E_b}{N_0}\right)_{nc}} = Q \sqrt{10 \left(\frac{E_b}{N_0}\right)_c}$$

$$\left(\frac{E_b}{N_0}\right)_{nc} = 5 \left(\frac{E_b}{N_0}\right)_c = 7dB + \frac{E_b}{N_0}_c$$

זאת אומרת שבח הקידוד הוא בקרוב 7dB.

5.10 שאלות סיכום פרק

1. כמה ביטים בקוד קונבולוציה עם $k = 7$ constraint length ועם יחס קידוד $\frac{1}{2}$ צריכים להישלח כדי לשדר חבילת ביטים באורך k .

2. נתון מקודד קוד קונבולוציה ביחס קידוד $1/3$ שמקודד לפי חוקיות הבאה:

$$[n_1, n_2, n_3] = [i, i \oplus i + 2, i \oplus i + 1 \oplus i + 2]$$

2.1 שרטט סכמה של המשדר.

2.2 הראה את הטרליס האינסופי ומצא את המרחק המינימלי.
כמה סדרות יש במרחק הזה?

2.3 אם מכניסים למשדר סדרת סיביות מידע $[1, 1, 0, 0, 1]$ (מימין לשמאל) ולאחריה אפסים לאיפוס הרגיסטר של המקודד, מה תהיה סדרת הביטים במוצא המשדר. הראה את הטרליס הסופי.

3. ביציאת מקלט שמתואם לעבוד מול המשדר של שאלה 2 מתקבלת סדרת הסיביות הבאה

$$[110 \ 010 \ 111 \ 000 \ 110 \ 110 \ 111]: \text{ (מימין לשמאל)}$$

פענח את האות הנקלט.

4. ביציאת מקלט שמתואם לעבוד מול המשדר של שאלה 2 מתקבלת סדרת רמות

המתח הבאה (מימין לשמאל):

$$[-.5, -.1, 1 \quad .9, -.1, .3 \quad -.3, -.7, -.9 \quad .9, .2, .7 \quad -.8, -.3, .9 \quad -.9, -.9, .9 \quad -.8, -.3, -.8]$$

פענח את האות הנקלט.

5. הנח קוד עם יחס קידוד $1/2$ נקב אותו ל: $3/4$, $4/5$, $5/6$, $7/8$

6. נתון מקודד קוד קונבולוציה ביחס קידוד $\frac{1}{2}$ שמקודד מידע d_i לשני ביטי יציאה עבור שני ביטי כניסה לפי חוקיות הבאה:

$$p_{i,1} = d_i \oplus d_{i-1} \oplus d_{i-2}$$

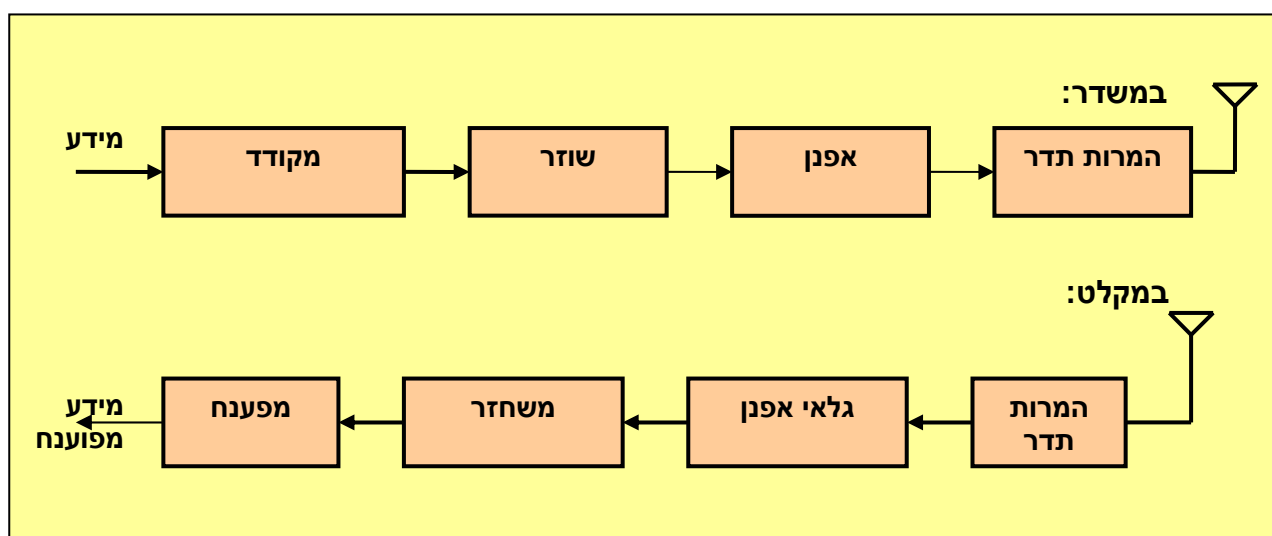
$$p_{i,2} = d_i \oplus d_{i-1}$$

- 6.1 הראה את הטרליס הסופי עבור 4 ביטי מידע. כמה ביטים מקודדים צריך לשדר?
- 6.2 אם מכניסים למקודד סדרת סיביות מידע $[1,0,0,1]$ ולאחריה אפסים לאיפוס הרגיסטר של המקודד, מה תהיה סדרת הביטים במוצא המשדר.
- 6.3 ביציאת מקלט שמתואם לעבוד מול המשדר הנ"ל מתקבלת סדרת רמות המתח הבאה: $[-.3, -.7; -.9, .9; .2, .7; -.8, -.3; .9, -.9; -.9, .9]$
- הזמן מתקדם משמאל לימין. רמות המתח המתאימות ל 0 ו 1 הן 1 ו 1 – בהתאמה. פענח את האות הנקלט.
- 6.4 הנח ניקוב אותו קוד ליחס קידוד $3/4$. כמה ביטים מקודדים צריך לשדר?
- 6.5 אם מכניסים למקודד סדרת סיביות מידע $[1,0,0,1]$ ולאחריה אפסים לאיפוס הרגיסטר של המקודד, מה תהיה סדרת הביטים במוצא המשדר.
- 6.6 ביציאת מקלט שמתואם לעבוד מול המשדר הנ"ל מתקבלת סדרת רמות המתח הבאה: $[-.3, -.9, .9, .7; -.8, .4, -.7, .9]$
- הזמן מתקדם משמאל לימין. רמות המתח המתאימות ל 0 ו 1 הן 1 ו 1 – בהתאמה. פענח את האות הנקלט.

6 שוזר – Interleaver

מערכת המשנה את סדר שידור המידע במשדר ובמקלט מחזירה את הסדר המקורי. במשדר הוא נקרא שוזר - Interleaver ואילו במקלט הוא נקרא משחזר - Deinterleaver. זהו תהליך שמבוצע בהרבה מערכות תקשורת. השם שוזר מאוד מקובל בעברית. לפעולה ההפוכה אין שם מוסכם ולכן המונח משחזר הוא לצורך נוחות. למיקום השוזר במערכת ישנם שתי צורות.

תצורה א' – לערוצים עם יחס אות לרעש לא קבוע



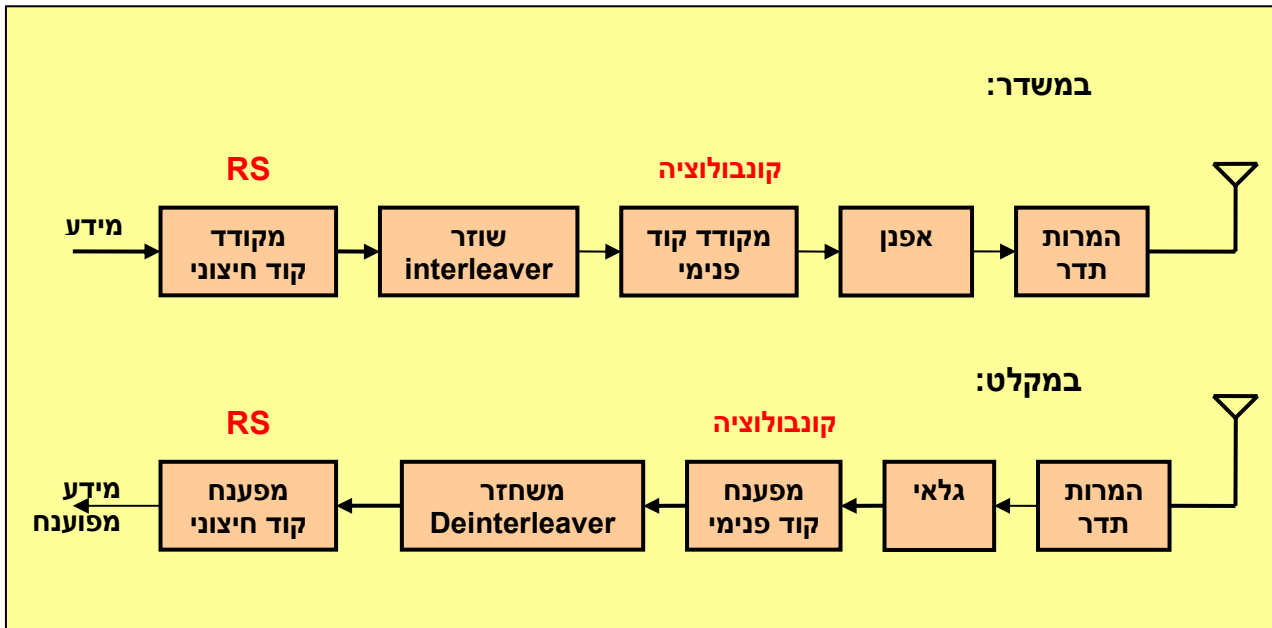
איור 6.1 מיקום השוזר במערכת תקשורת ספרתית (תצורה א')

השוזר נועד לטפל בערוצים שאינם ערוצים רגילים לדוגמא ערוץ רעש לבן "רגיל" – AGWN=Additive Gaussian White Noise. בערוץ רעש לבן "רגיל" השוזר אינו מועיל ואינו גורע.

השוזר מועיל במקרה של ערוצים עם חוסמים וערוצים דועכים. מאפיין ערוצים אלו הוא שיחס אות לרעש SNR אינו קבוע עם הזמן וכתוצאה מכך נוצרים פרצים של הפרעות. בערוצים עם חוסמים האות קבוע אבל הרעש גדל וגורם להפרעה. בערוצים הדועכים הרעש קבוע אך האות דועך בפתאומיות. בשני המקרים יחס אות לרעש עלול להשתנות לרעה.

בערוצים מסוג זה נוצרים פרצים של הפרעות. רוב הקודים המעשיים מתקשים לטפל בפרצים של שגיאות רבות המגיעות יחד והשוזר בא לטפל בבעיה זו.

תפקיד השוזר הוא לפזר את פרץ השגיאות ואז הפענוח נעשה על מידע עם שגיאות מפוזרות. המשחזר מסדר את המידע בסדר נכון ובו בזמן מפזר את השגיאות. במשדר קיים השוזר כדי שסדר המידע לא ישתנה.



איור 6.2 מיקום השוזר במערכת תקשורת ספרתית (תצורה ב')

תצורה זו הומצאה ע"י Forney והוא קרא לשיטה קודים משורשרים- Concatenated codes, כיום השיטה נקראת שרשור טורי- Serial concatenated. זה היה עד להמצאת קודי הטורבו. הקוד האולטימטיבי משרשר קוד פנימי וחיצוני. במקודד הפנימי נשתמש בקודי קונבולוציה ואילו במקודד החיצוני נשתמש בקודי RS וזאת משום שקודי RS עובדים על אותיות טובים לתיקון שגיאות המגיעות בפרצים. המשחזר עוזר לקוד RS לפזר את השגיאות. מי שיוצר את פרצי השגיאות זה המפענח קוד פנימי שהוא קוד קונבולוציה. ואילו קוד RS תפקידו לתקן אותן.

ישנם שני סוגים של שוזרים נפוצים : 1. שוזר בלוקים
2. שוזר קונבולוציה.

6.1 שוזר בלוקים:

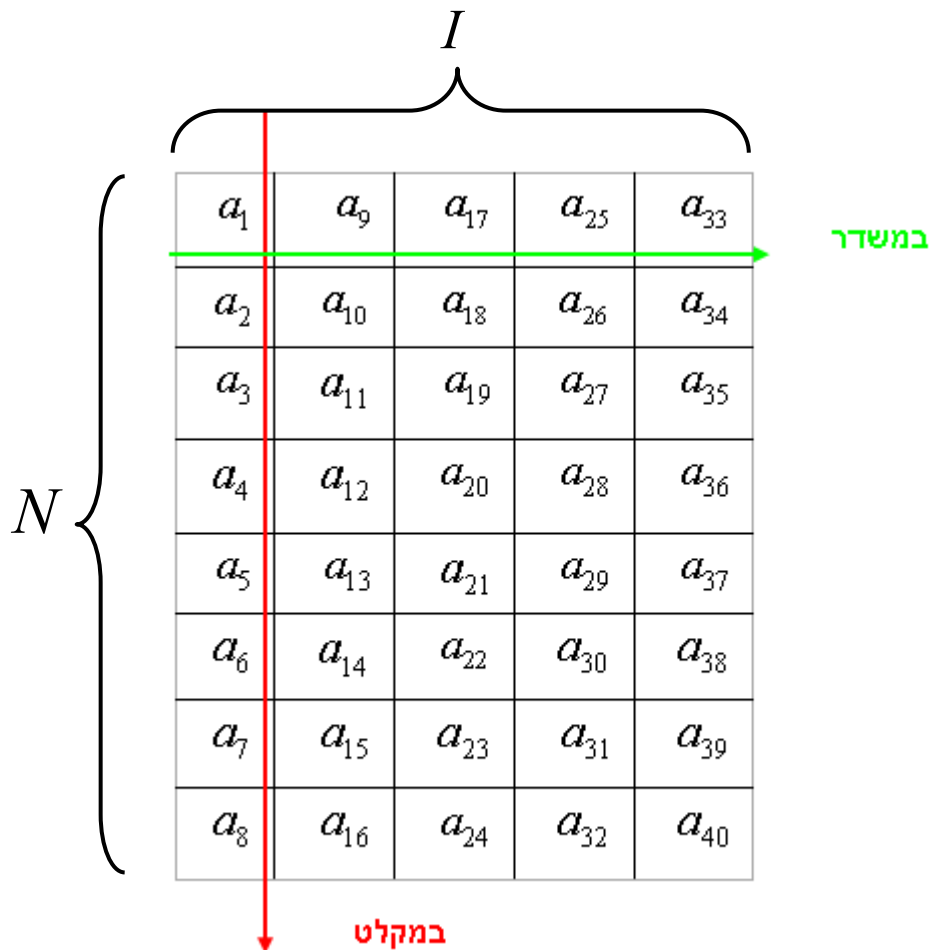
לעמודות קוראים עומק $I = 5$ depth
לשורות קוראים פריסה $N = 8$ span

$a_1, \dots, a_{40}$ - מידע מקודד.

במשדר מכניסים מידע לפי עמודות ומשדרים לפי שורות. כלומר בשורה הראשונה

נוציא $a_1, a_9, a_{17}, a_{25}, a_{33}$.

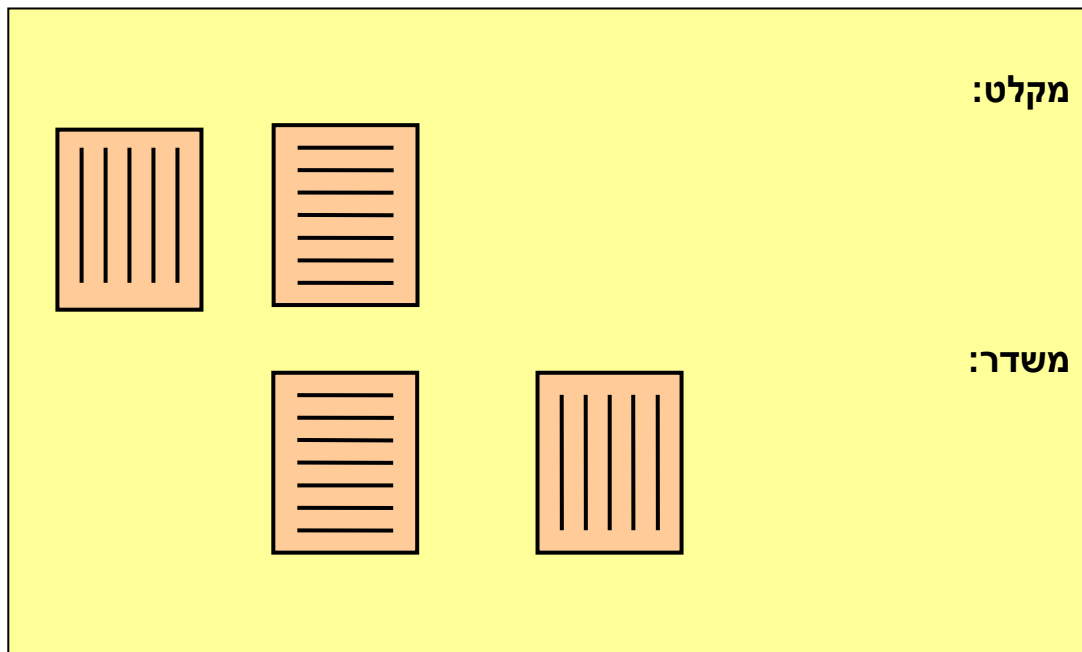
במקלט מבצעים את התהליך בצורה הפוכה כלומר ממלאים את הטבלה לפי שורות ומוציאים למפענח לפי עמודות.



טבלה 6.1 שוזר בלוקים

כמה תאי זיכרון נדרשים במשדר?
 התשובה לכך היא $2 \cdot N \cdot I$. במקרה שלנו $2 \cdot N \cdot I = 2 \cdot 8 \cdot 5 = 80$ תאי זיכרון כאשר גם במקלט נדרשים $2 \cdot N \cdot I$ תאי זיכרון.

6.2 השהיה הנגרמת על ידי שוזר/משחזר



איור 6.3 השהיית שוזר ומשחזר

דוגמא 6.1 :

נשאלת השאלה מה ההשהיה הנגרמת ע"י שוזר / משחזר?

ההשהיה הנגרמת במערכת היא $2 \cdot N \cdot I$. כאשר במשדר מגיע הביט הראשון הוא ממתין $N \cdot I$ ביטים עד למילוי השוזר ולאחר מכן שולח אותו. במקלט האות נקלט במקביל לשליחת המידע מהמשדר כך שאין השהיה. כעת על המקלט למלא את הטבלה במשחזר בגודל של $N \cdot I$ ביטים לפני הוצאת המידע ממנו. הביט הראשון שיצא מהמשחזר יהיה בעל השהיה $2 \cdot N \cdot I = 80$.

6.3 טיפול בפרץ שגיאות

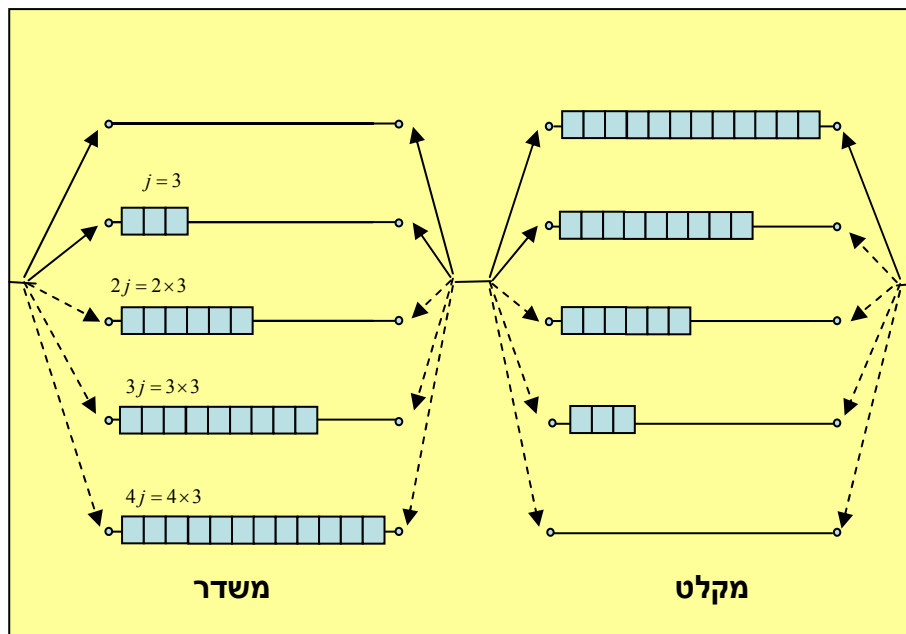
אם יש פרץ שגיאות בערוץ באורך I , אזי המרחק שנוצר ביניהם בכניסה למפענח הוא $N-1$ לפחות.
אם משתמשים בשזור בשרשור טורי המצוין בתצורה ב' (איור 6.2) אזי עליו להיות שזור בתים עקב השימוש בקוד RS של בתים וזאת על מנת למנוע הכפלה של מספר השגיאות אשר היו מתקבלות אילו היינו משתמשים בביטים. כלומר בתצורה ב' עם שרשור קודים כאשר הקוד החיצוני הוא עם אותיות בגודל m ($m=8$), אזי השזור צריך להעביר אותיות בגודל m (עבור $m=8$ שיזור בתים).

שזור במערכת דילוגי תדר – Frequency Hopping

מערכת בה משנים את התדר מידי פעם. בתקשורת משתמשים במערכת זו כדי להקשות על החוסם ואז נאלצים להשתמש ברוחב סרט הרבה יותר גדול מהנדרש. (spread spectrum).
בשזור בלוקים המשולב עם מערכת דילוגי תדר משדרים כל שורה בתדר אחר.

6.4 שזור קונבולוציה

השזור והמשחזר מסוג קונבולוציה מתואר באיור 6.4.
המידע הן במשדר והן במקלט עובדת לסירוגין דרך I מסלולים. בכל מסלול קו השהייה. מידע שעובר במשדר רק השהיה קצרה עובר במקלט השהיה ארוכה כך שלאחר השהייה של שזור ומשחזר ההשהיה של כל הביטים זהה.



איור 6.4 שזור קונבולוציה $j=3, I=5$

I הינו מספר הערוצים j הינו גודל ההשהיה הבסיסית. במקרה איור 6.4 $j = 3, I = 5$. הביטים נכנסים מלמעלה למטה ואז I כניסות יוצאות בסדר שונה. בשלושה סבבים ראשונים יצא מידע הקופץ ב I , אחרי $3 \cdot I$ ביטים יצטרף הביט השני, אחרי $6 \cdot I$ יצטרף הביט השלישי, אחרי $9 \cdot I$ הביט הרביעי, אחרי $12 \cdot I$ הביט החמישי ובביט השישי מתחילים מחדש וכך ממשיכים הלאה. שימו לב לכל ביט יש השהייה שונה.

טבלת שוזר קונבולוציה:

בב

1	a_1	—	—	—	—
2	a_6	—	—	—	—
3	a_{11}	—	—	—	—
4	a_{16}	a_2	—	—	—
5	a_{21}	a_7	—	—	—
6	a_{26}	a_{12}	—	—	—
7	a_{31}	a_{17}	a_3	—	—
8	a_{36}	a_{22}	a_8	—	—
9	a_{41}	a_{27}	a_{13}	—	—
10	a_{46}	a_{32}	a_{18}	a_4	—
11	a_{51}	a_{37}	a_{23}	a_9	—
12	a_{56}	a_{42}	a_{28}	a_{14}	—
13	a_{61}	a_{47}	a_{33}	a_{19}	a_5
14	a_{66}	a_{52}	a_{38}	a_{24}	a_{10}
15	a_{71}	a_{57}	a_{43}	a_{29}	a_{15}

טבלה 6.2 טבלת שוזר קונבולוציה

בטבלה מתוארים הביטים שיוצאים מתוך השוזר לפי הסדר. את הטבלה יש לקרוא משמאל לימין ומלמעלה למטה, זאת אומרת שורה אחרי שורה כמו באנגלית. הביטים שנכנסים לשוזר הם a_i . יוצא ללא השהיה במסלול הראשון, a_2 עד a_5 נכנסים לקו ההשהיה, ועל כן ביציאה יש ---- זאת אומרת התוכן הראשוני של תאי הזכרון. a_6 שוב יוצא במסלול הראשון ללא השהיה ו- a_7 עד a_{10} נכנסים לזיכרונות של המסלולים האחרים. לאחר סבב נוסף a_{16} שוב יוצא במסלול הראשון ללא השהיה אך הפעם כאשר a_{17} נכנס למסלול השני a_2 יוצא. במסלולים האחרים עדיין יש מידע ראשוני רק הסבב השביעי לאחר ש a_{31} יוצא ללא השהיה ו- a_{17} יוצא מהמסלול השני a_3 יוצא מהמסלול השלישי. החל מסבב 13 בכל המסלולים יוצא מידע.

סדר יציאת הביטים מהמשדר יהיה:

$a_1, a_6, a_{11}, a_{16}, a_2, a_{21}, a_7, a_{26}, a_{12}, a_{31}, a_{17}, a_3, a_{36}, a_{22}, a_8, \dots, a_{15}$

גודל הזיכרון במשדר או במקלט:

$$\frac{(I-1) \cdot j \cdot I}{2} = \frac{(I-1) \cdot N}{2}$$

כאשר: $N \triangleq jI$

עבור פרץ שגיאות בערוץ בגודל I , אזי בכניסה למפענח (לאחר השוזר) המרחק בין שתי שגיאות יהיה לפחות $N-1$.

השהיה ניתנת על ההשהיה במסלול האחרון: $(I-1)jI = (I-1)N$

כאשר משתמשים בשוזר קונבולוציה נדרשים רבע זיכרון וחצי השהיה לעומת שוזר בלוקים.

החיסרון של שוזר זה הוא שבתחילה צריך לשדר הרבה אפסים (במקרה שלנו 30 אפסים) עד שאוגרי ההזזה מתמלאים. כמות האפסים היא כגודל הזיכרון מכון וצריך לרוקן את כל הזיכרון מאפסים. כלומר מספר האפסים הוא:

$$\frac{(I-1)N}{2} = \frac{jI}{2}(I-1)$$

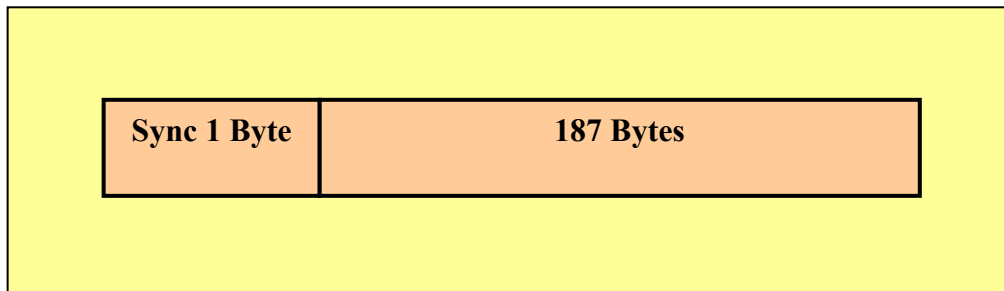
חסרון זה אינו משמעותי עבור רצף ארוך של מידע כמו במקרה של תקנים לשידור טלויזיה ספרתית.

6.5 שאלות סיכום פרק

1. בהינתן קוד גולאי לתיקון שגיאות $(24,12,3)$ עם קצב סימבולים R_s . בערוץ קיימים פרצי שגיאת באורך b סימבולים.
 - 1.1. תכנן שוזר בלוקים אופטימלי. מהם ממדי השוזר, גודל הזיכרון, והשהית השוזר?
 - 1.2. מהו מספר הפרצים שהקוד והשוזר יכלו להתמודד ומהו קצב הפרצים המחזוריים המקסימלי שיאפשר קבלת מידע ללא שגיאות.
 - 1.3. שנה את השוזר לקבלת השהיה מינימלית (לקצב פרצים מחזורי נמוך), מהו כעת קצב הפרצים המחזוריים?
2. בהינתן מערכת עם קוד קונבולוציה לתיקון שגיאות בעל יחס קידוד חצי וקצב סיביות R_s
 - 2.1. תכנן שוזר קונבולוציה כך שכול פרץ שגיאות בכניסה למפענח לא יפגע ביותר מסיבית אחת של מידע מתוך 32 סיביות מידע. מה הם ממדי השוזר, זיכרון של השוזר ומהי ההשהיה בשוזר?
 - 2.2. מהו קצב הפרצים המחזוריים המקסימלי כך שלא תהיה שגיאה במידע המתקבל?
3. בהינתן מערכת עם דילוגי תדר ברוחב פס W עם קצב דילוגים R_H , מאופנן בשיטת BPSK עם קוד לתיקון שגיאות BCH (n,k,t) , קצב סימבולים R_s , הספק אות S וחם בעל הספק J וברוחב פס ρ
 - 3.1. תכנן שוזר בלוקים אופטימלי, מהם: ממדי השוזר, השהית השוזר וגודל הזיכרון של השוזר?
 - 3.2. מהי הסתברות לשגיאה במוצא המפענח ומהי הסתברות לשגיאה במקרה הגרוע ביותר שיכול להיות?
- תכנן מחדש את השוזר כדי שהשהיה תקטן פי שתיים, מהם הממדים החדשים של השוזר? כיצד זה משפיע על ביצועי השוזר?

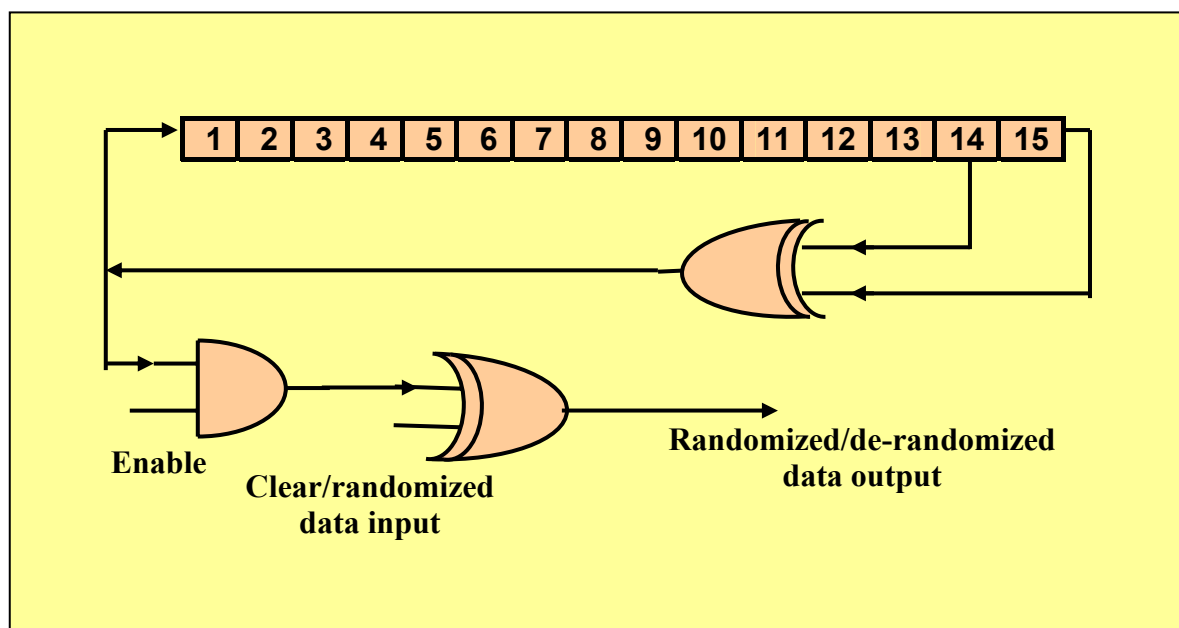
DVB - S 7

המידע הנכנס למערכת צריך להיות מאורגן כחבילות בעלות אורך קבוע. (ראה איור 7.1) והם באות לאחר ריבוב העברה של MPEG-2. האורך הכולל של חבילת מרבב ה MPEG-2 הוא 188 Byte. וזה כולל בתוכו מילת סנכרון (לדוגמא 47HEX) באורך Byte אחד. הצד המשדר מתחיל תמיד בתהליך השידור מהביט המשפיע ביותר של מילת הסנכרון.



איור 7.1 מרבב 2 - MPEG

המידע מסוכם בפעולת XOR עם המוצא של מחולל אותות דמוי אקראיים (Pseudo Random Binary Sequence) בעל פולינום יוצר : $1 + x^{14} + x^{15}$

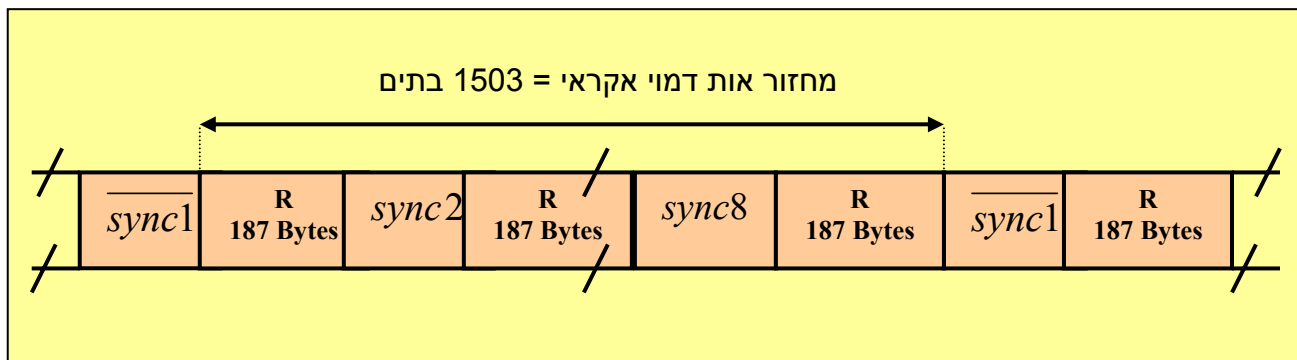


איור 7.2 מחולל אותות דימוי אקראי

מחולל האותות הדמוי אקראי מבצע ערבול של המידע על ידי חיבורו (XOR) בסדרה הדמוי אקראית. לאוגר ההזזה יש משובים כאשר מספר האפשרויות של המצבים הוא $2^{15} - 1$. מכון שאסור שכל המצבים יהיו אפסים הסדרה הדמוי אקראית הינה סדרה מחזורית ואורך מחזור אחד הוא $2^{15} - 1$. מחולל הסדרה הדמוי אקראית מונע קיום רצפים של אפסים ואחדים במידע. במקלט ביצוע חוזר של אותה פעולה מבטלת אותה.

7.1 קידוד חיצוני

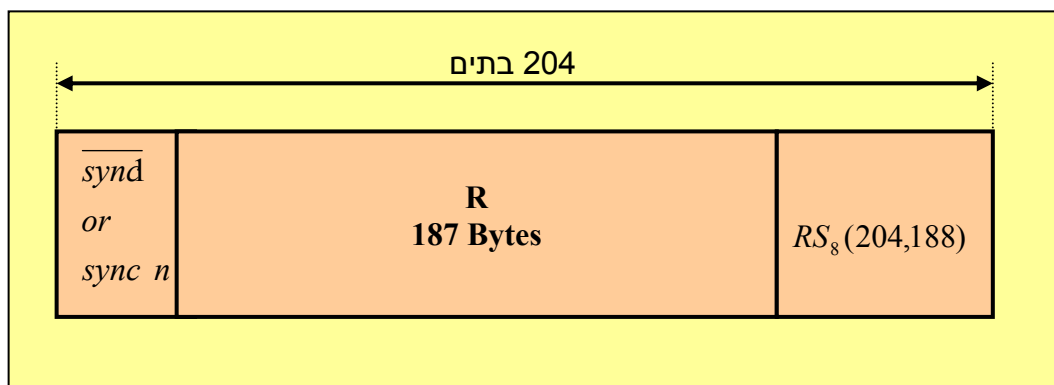
נשתמש בשיטה זו בקוד RS. ניקח קוד מקוצר $RS_8(204,188)$ של הקוד $RS_8(255,239)$. נצרף את הקוד לכל משלוח חבילה (188 בתים) אקראית מאיור 7.3 וזאת על מנת ליצור חבילה המוגנת משגיאות (מתואר באיור 7.4) קוד RS יצורף גם כן לאות הסנכרון בין אם הוא B8HEX או 47HEX.



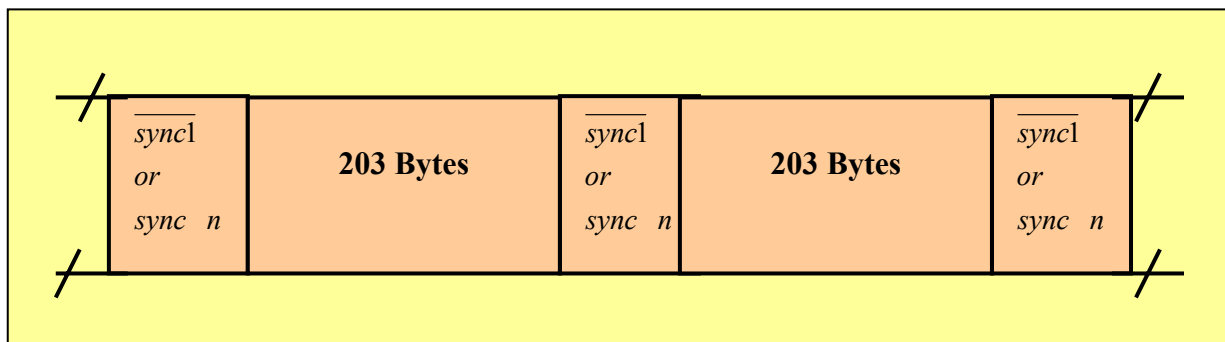
איור 7.3 מחזור אות דמוי אקראי

קוד RS המקוצר יכול להיות מיושם על ידי הוספת 51 בתים אשר כולם יהיו אפסים לפני בתי המידע בכניסה למקודד (255,239). לאחר תהליך קידוד RS הבתים המאופסים נזרקים. שוזר קונבולוציה יצורף לחבילה המוגנת משגיאות (איור 7.4) וכתוצאה מכך נקבל מסגרת משוחזרת (איור 7.5) למטרות סנכרון. אות הסנכרון ינותב תמיד בענף "0" של השוזר. ההתייחסות הינה להשהיית "0".

הערה: המשחזר בעקרון תואם לשוזר אבל המידע בענפים בו הפוך – הסנכרון של המשחזר יתבצע על ידי ניתוב האות המסנכרן המזוהה הראשון בענף ה "0".



איור 7.4 יצירת חבילה המוגנת משגיאות



איור 7.5 מסגרת משוזרת עבור שזור בעל עומק $I = 12$

7.2 קידוד פנימי

הקוד הפנימי הוא בקוד קונבולוציה. המערכת תאפשר תחום של קודי קונבולוציה מנוקבים המבוססים על יחס קידוד חצי אם אורך $k=7$. עובדה זו תאפשר את הבחירה המתאימה ביותר של יחס הקידוד עבור קצב מידע נתון. המערכת תקבל קודי קונבולוציה בעלי יחסי קידוד חצי, שני שלישי, שלוש רבעים, חמש שישיות, שבע שמניות. שימוש בקודי קונבולוציה המנוקבים יעשה לפי טבלה 7.1.

Original Code			Code Rates					
			$\frac{1}{2}$		$\frac{2}{3}$		$\frac{3}{4}$	
k	G_1 (X)	G_2 (Y)	P	d_f	P	d_f	P	d_f
7	171_{OCT}	133_{OCT}	$X : 1$ $Y : 1$ $I = X_1$ $Q = Y_1$	10	$X : 10$ $Y : 11$ $I = X_1 Y_2 Y_3$ $Q = Y_1 X_3 Y_4$	6	$X : 101$ $Y : 110$ $I = X_1 Y_2$ $Q = Y_1 X_3$	5

Original Code			Code Rates			
			$\frac{5}{6}$		$\frac{7}{8}$	
k	G_1 (X)	G_2 (Y)	P	d_f	P	d_f
7	171_{OCT}	133_{OCT}	$X : 10101$ $Y : 11010$ $I = X_1 Y_2 Y_4$ $Q = Y_1 X_3 X_5$	4	$X : 1000101$ $Y : 1111010$ $I = X_1 Y_2 Y_4 Y_6$ $Q = Y_1 Y_3 X_5 X_7$	3

טבלה 7.1 הגדרת קוד מנוקב

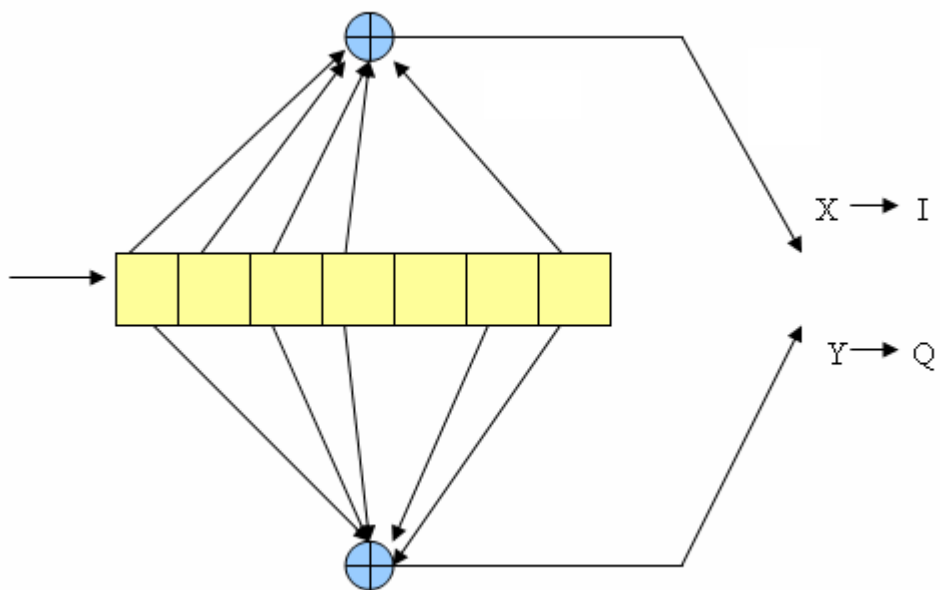
דוגמא 7.1 :

נתון קוד מקורי: 171 oct אוקטאלי
133 oct אוקטאלי

1	3	3	1	7	1
1	011	011	1	111	001

נמיר לבינארי ונקבל :

ונבנה את המקודד הבא:



מספר הקוביות שווה ל - k כלומר 7 , במקרה שלנו לא ייתכן שיהיה מספר אוקטלי גדול מ- 177 כי מעבר לגודל זה נצטרך k יותר ארוך.
בכדי להגיע ליחס קידוד $\frac{2}{3}$ יש לנקב את הביט השני בכל רביעייה של ביטים מקודדים.

8 קודי טורבו

8.1 מבוא לקודי טורבו – Iterative decoding -Turbo

השיטה התפרסמה ב - 1993 כאשר **Berrou** הציג אותה.
זו הייתה ההתפתחות החשובה ביותר של העשור בתיאורית הקודים.

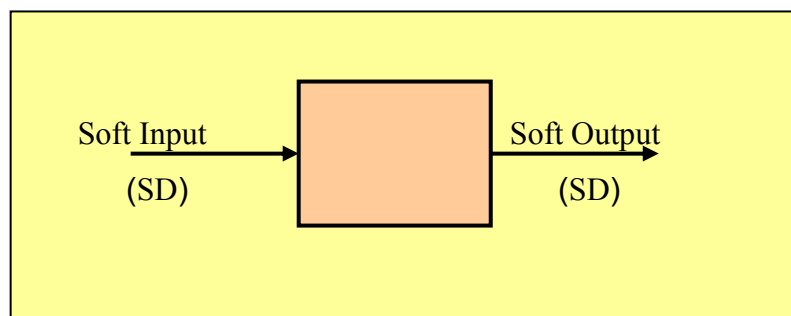
יתרונות:

- שבח הקידוד - בשבח הקידוד הטוב ביותר. קרוב לחסמים של תורת האינפורמציה.
- עוצמת שידור נמוכה
- אנטנה קטנה יותר
- תווך עבודה רחב יותר
- קצבי מידע גבוהים יותר

מאפייני הקוד:

- א. קוד טורבו זהו קוד שבנוי משניים או יותר קודים בסיסיים (בדרך כלל שניים). שני הקודים יכולים להיות כל סוג (קונבולוציה, RS, חזרות) כמו כן הם יכולים להיות משורשים בכל צורה.
- ב. ישנו שוזר Interleaver (המידע לא נכנס באותו סדר למקודדים).

ג. הפענוח מתבצע בעזרת מנגנון SISO



הערה: באלגוריתם הרגיל של ויטרבי קיים SIHO (SD בכניסה HD ביציאה).

- ד. פענוח איטראטיבי – בפענוח המידע המפוענח עובר ממפענח קוד אחד למפענח קוד אחר.

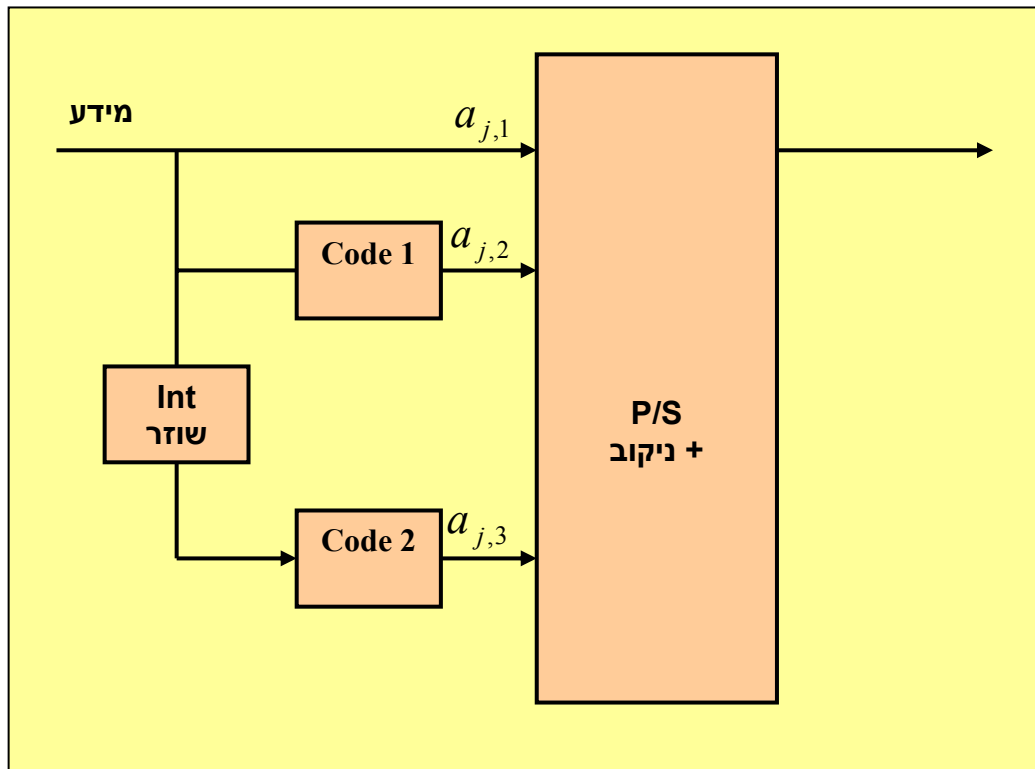
8.2 מבנה של מקודד

ישנן ארבע ארכיטקטורות נפוצות:

- שרשור מקבילי - Parallel concatenated code
- שרשור טורי - Serial concatenated code
- קוד מכפלה - Product code
- LDPC - Low density parity code

8.2.1 שרשור מקבילי Parallel concatenated

לוקחים את המידע מכניסים אותו לשני מקודדים.



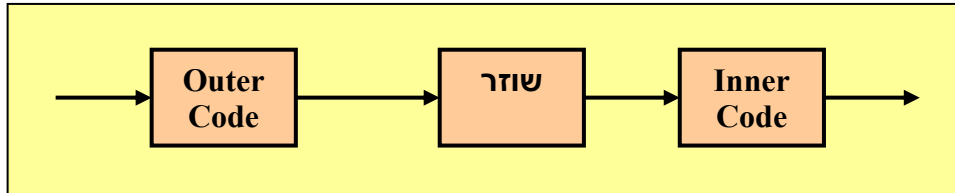
איור 8.1 שרשור מקבילי

יחס הקידוד פה הוא $\frac{1}{3}$ ואם רוצים להגדיל את יחס הקידוד (לדוגמא ל- $\frac{1}{2}$) אז יש לנקב.

ברו הציע קודי קונבולוציה ושרשור מקבילי.

8.2.2 שרשור טורי – Serial Concatenated

שני הקודים משורשרים בטור כמו באיור 8.2



איור 8.2 שרשור טורי

לקוד שקרוב לערוץ קוראים קוד פנימי ולרחוק ממנו קוראים חיצוני.

8.2.3 קוד מכפלה Product Code

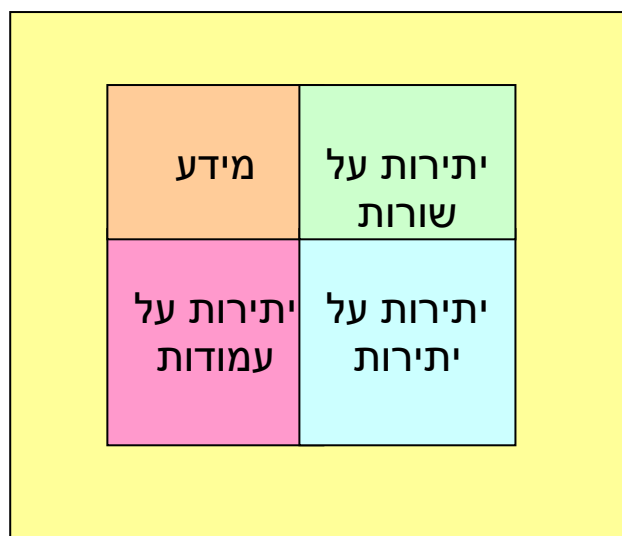
את המידע נכניס לקובייה של $k \times k$, גודל המטריצה יהיה:

a_{ij} שהינו מידע בגודל של $k \times k$, ולכן $a_{11} \dots a_{kk}$

כל שורה ושורה ומקודד בקוד בלוקים. את המידע שקיבלנו רושמים מצד ימין (הריבוע הירוק).
נקרא לזה יתירות (Redundancy) (מקבלים את המידע הרשום בריבועים החום והירוק).
לאחר מכן מקודדים כל עמודה ועמודה. את המידע שקיבלנו רושמים מתחת (מקבלים את המידע הרשום בריבועים החום והורוד).

לאחר מכן מבצעים יתירות על יתירות וזאת על ידי הכנסת מידע שני הריבועים הורוד והירוק יחד כך שיווצר מידע חדש שהוא הריבוע התכלת.

המבנה מתואר באיור 8.3:



איור 8.3 מבנה כללי לקוד מכפלה

סה"כ יתירות שקיבלנו: $n^2 - k^2$ או שהקוד יהיה (n^2, k^2) .

יתירות על יתירות ניתן לעשות על שורות או על עמודות, בכל מקרה נקבל אותה תוצאה.

רוב התקנים בתקשורת סלולארית משתמשים בקידוד מקבילי עם קודי קונבולוציה.

a_{11}	a_{12}	..	..	a_{1k}	$a_{1,k+1}$	..	..	..	$a_{1,n}$
a_{21}	a_{22}	..	..	a_{2k}	..	..	..	..	..
.	..	..	..	..	.	..	..	..	..
	..	..	..			..	..	..	
a_{k1}	a_{k2}	..	..	a_{kk}	$a_{k,k+1}$	..	..	..	$a_{k,k+1}$
$a_{k+1,1}$	..	..	..	$a_{k+1,k}$	$a_{k+1,k+1}$	..	..	..	$a_{k+1,n}$
..	..	..	..	..	..	..	..	..	..
.	..	..	..	..	.	..	..	..	..
	..	..	..			..	..	..	
$a_{n,1}$	..	..	..	$a_{n,k}$		..	..	..	$a_{n,n}$

איור 8.4 מבנה מפורט לקוד מכפלה

8.3 הסבר פענוח איטרטיבי של קוד מקבילי:

לאחר ששידרנו $a_{j,1}$ $a_{j,2}$ $a_{j,3}$ נקלט r_{i1}, r_{i2}, r_{i3}

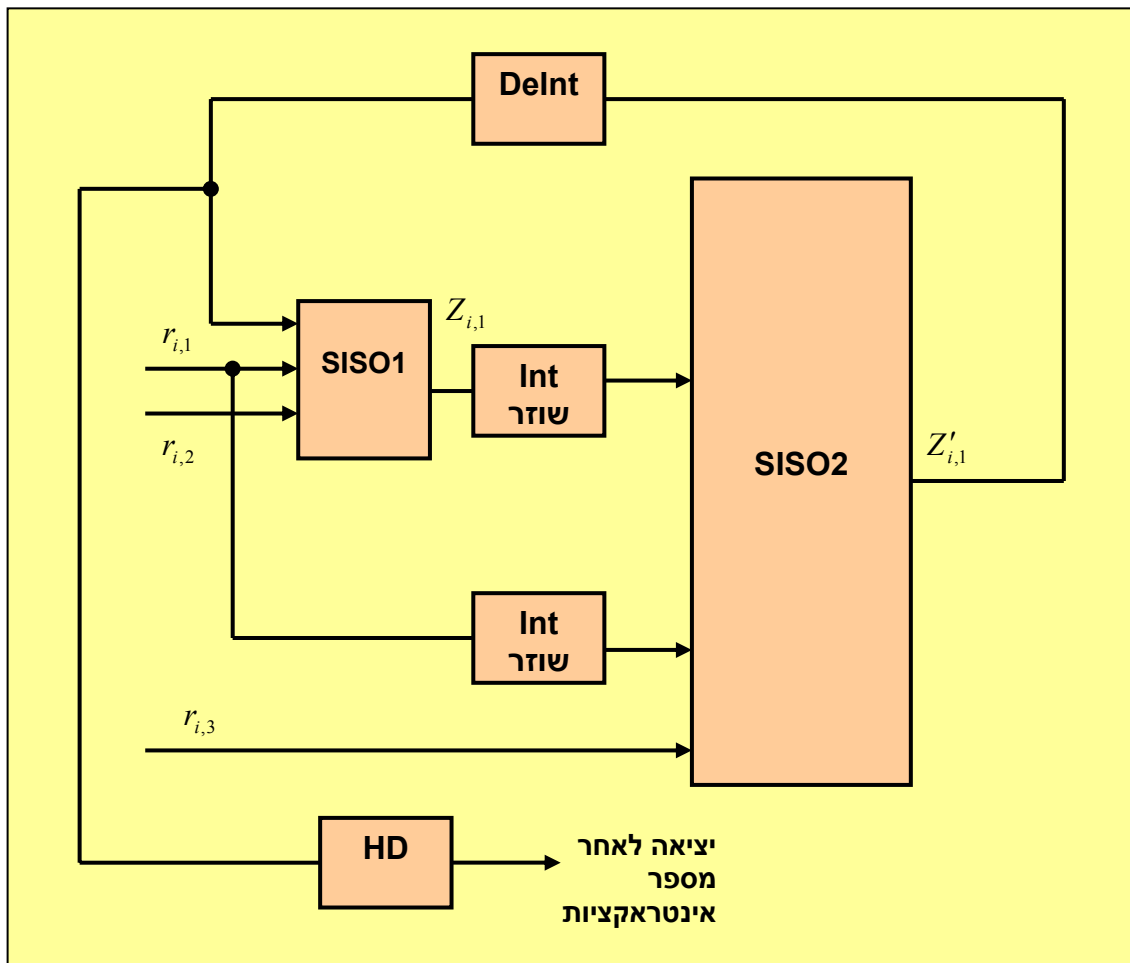
המשדר יהיה כמו באיור 8.1 .

לעומת זאת המקלט יהיה כמתואר באיור 8.5:

נגדיר ש- SISO1 מפענח CODE1 , החלטות רכות לגבי המידע. $Z_{i,1}$

$Z'_{i,1}$ החלטות של השני שנכנסות חזרה לראשון.

המפענח הראשון מקבל את המידע המקודד.
המפענח השני מקבל את המידע ואת ההחלטות של הראשון,
את ההחלטות של השני הוא מחזיר לראשון וכך הלאה.
כל סבב של שני המפענחים מכונה איטרציות. המידע יוצא להחלטה קשה לאחר מספר איטרציות.



איור 8.5 מקלט פענוח איטרטיבי בשרשור מקבילי

מקורות

- Digital Communications - נכתב על ידי SKLAR בהוצאת Prentice Hall מהדורה שנייה 2001. פרקים 6,7,8.
- תקן DVB-S.