

סיבית

סִבִּית (קיצור של **סִפְּכָה בִּינָרִית**^[1] באנגלית bit או ביט, מתוך השם "binary digit") היא ספרה בינארית – יחידת הנתונים הקטנה ביותר שבה משתמש המחשב.

המושג bit נטבע לראשונה על ידי ג'ון טוקי והובא ב-1948 במאמרו המפורסם של קלוד שאנון^[2], אבי תורת האינפורמציה, כדי לתאר את יחידת המידע הקטנה ביותר^[3].

תוכן עניינים

יצוג מידע

יחידות מידה מבוססות סיביות

העברת מידע

ייצוג בינארי של אותיות

ראו גם

קישורים חיצוניים

הערות שוליים

יצוג מידע

סיבית יכולה להכיל ערך 0 או 1 בלבד. הסיבה לשימוש בשיטה הבינארית היא פשטות המימוש האלקטרוני והלוגי של שיטה זו - נדרש טיפול בשני מצבים בלבד (למשל: יש זרם = 1, אין זרם = 0).

ניתן ליצג מידע באמצעות דפוסים של סיביות על ידי תהליך של דיגיטציה^[3].

יחידות מידה מבוססות סיביות

בַּיְת (Byte) מורכב מ-8 סיביות. בית מיוצג באנגלית באות B. אין להתבלבל בינה לבין סיבית, שמוצגת באות b.

בעבר גודל הבית לא היה מוגדר והיה תלוי מימוש. יחידה שהתייחסה במפורש ל-8 סיביות כונתה Octet אך מאחר שעם השנים בית התייחס באופן בלעדי ל-8 סיביות, כינוי זה הפך לפחות שכיח. גודל של ארבע סיביות מיוצג על ידי מונחים שונים כגון Nibble, חצי בית (halfbyte), סמי-אוקטט (semi-octet) וקואדביט (quadbit). מאחר שגודל זה מייצג ספרה בבסיס הקסדצימלי, לעיתים מתייחסים לכך כספרת הקסה (Hex digit). כך למשל, לעיתים נוח יותר להציג בית כשתי ספרות הקסה.

בסיס 10	בסיס 16	ערכים וחזקות	סימול גודל	שם
$10^3 <$	$16^{2.5} =$	$1,024 = 2^{10}$	k	kilo
$10^6 <$	$16^5 =$	$1,048,576 = 2^{20}$	M	mega
$10^9 <$	$16^{7.5} =$	$1,073,741,824 = 2^{30}$	G	giga
$10^{12} <$	$16^{10} =$	$1,099,511,627,776 = 2^{40}$	T	tera
$10^{15} <$	$16^{12.5} =$	$1,125,899,906,842,624 = 2^{50}$	P	peta
$10^{18} <$	$16^{15} =$	$1,152,921,504,606,846,976 = 2^{60}$	E	exa
$10^{21} <$	$16^{17.5} =$	$1,180,591,620,717,411,303,424 = 2^{70}$	Z	zetta
$10^{24} <$	$16^{20} =$	$1,208,925,819,614,629,174,706,176 = 2^{80}$	Y	yotta

העברת מידע

מערכת תקשורת יכולה להעביר מידע דיגיטלי.

מהירות ההעברה של נתונים בקווי תקשורת נמדדת בסיביות לשנייה, או בכפולות של יחידה זו, כגון קילוביט לשנייה, מגה-ביט לשנייה וג'יגה-ביט לשנייה.

ייצוג בינארי של אותיות

באמצעות מחרוזת של סיביות, מושג במחשב ייצוג בינארי של אותיות (וסימנים אחרים). כאשר למטרה זו משמשת סיבית אחת, ניתן לייצג שתי אותיות שונות. ככל שמחרוזת הסיביות ארוכה יותר נוכל להגדיר יותר סימנים שונים, אך נשלם בצריכת מקום אחסון (לסוגיו השונים). רצף של סיביות המשמש להצגה של סימן אחד קרוי ת.

ייצוג בינארי של אותיות נכנס לשימוש שנים רבות לפני המצאת המחשב, במקרים שבהם נדרשה דרך פשוטה לכתיבה וקריאה. כתב ברייל הוא קוד בינארי בן ששה ביטים, כך שיתכנו בו $2^6 = 64$ סימנים שונים, אך ציחופים אחדים אינם מנוצלים משום שהם עלולים להטעות את הקורא.

במכשירי טלפרינטר שימש להעברת המידע סרט ממוקב בקוד בודו, שבו כל תו יוצג באמצעות חמישה חורים, כלומר התאפשרו $2^5 = 32$ סימנים שונים. זה מספר קטן מדי של סימנים, שאינו מאפשר אפילו ייצוג של אותיות וספרות בלבד. לפתרון הבעיה נלקחו שני תווים להיות תויו בקרה, בעלי אופי דומה לזה של המקש Shift שבמקלדת, כלומר לעבור למערכת חלופית של סימנים. עוד שלושה תוויו בקרה משותפים לשתי מערכות הסימנים, כך שבסך הכול מאפשרת שיטה זו ייצוג של $2 \times 2^7 = 54$ סימנים שונים, וזה מספיק לאותיות (גדולות בלבד), ספרות וסימני פיסוק.

במחשבי CDC, שפעלו בתחילת שנות השבעים, גודל מילה היה 60 ביטים (גודל זה נקבע משום שיש לו יתרון בולט בחישובים, שזו הייתה מטרתם העיקרית של מחשבים אלה). גודל סימן נקבע ל-6 ביטים, כך שבמילה אחת ניתן לאחסן 10 סימנים. מספר הסימנים השונים היה $2^6 = 64$. זהו מספר קטן יחסית, אך, כאמור, עיקר פעילותם של מחשבים אלה הייתה חישובים ולא טיפול במידע תויו.

בקוד ASCII נקבעו תחילה 7 ביטים לייצוג תו, כך שהתאפשרו 128 סימנים שונים: אותיות (גדולות וקטנות), ספרות, סימני פיסוק ועוד. בקוד ASCII מורחב נוסף עוד ביט, כך שמתאפשרים 256 סימנים שונים, כולל אותיות עבריות, למשל, ועוד סימנים מיוחדים רבים. קוד זה משמש במרבית המחשבים הפועלים כיום. גם במחשבי IBM-Mainframe, שייצורם החל באמצע שנות השישים, נהוג קוד בן 8 ביטים, אך המשמעות שניתנת בו לכל צירוף של ביטים שונה. קוד זה קרוי EBCDIC.

קוד בן 8 ביטים די בו למערכות טקסטואליות מקובלות, שבהן משתמשים בשפה אחת, או לכל היותר בשתי שפות. במערכות רב לשוניות, שבהן נדרשות גם אותיות ערביות, קיריליות וכו', לא די בכך. פתרון אפשרי הוא זה שננקט במכשירי הטלגרף, כלומר שימוש בתווים בקרה הגרמים להחלפת מערכת האותיות. בפתרון זה יש אי-מחוח מסוימת, מה גם שאינו פותר את בעיית השפות של המזרח הרחוק, שבהן יש אלפי סימנים. פתרון יסודי לבעיה זו הוא המעבר למערכת יוניקוד, שבה משמשים אחד עד ארבעה בתים, כלומר עד 32 ביטים, לייצוג כל סימן. מערכת זו מאפשרת יותר ממיליון סימנים שונים, די והותר לכל סימני השפות המקובלות בעולמנו גם יחד.

בכל מערכות הסימנים שבהן עסקנו עד כה, ניתן אורך אחיד לכל הסימנים במערכת. זו שיטה פשוטה, אך היא אינה יעילה מבחינת נפח האחסון שהיא צורכת: יש סימנים נדירים שיכולנו לתת להם קוד ארוך יותר, ולעומתם להעדיף קוד קצר יותר לסימנים הנפוצים. יישום של גישה זו קיים כבר בקוד מורס לטלגרף אלחוטי (קוד זה משמש עד היום כשנדרשת תקשורת אמינה בתנאי קליטה גרועים, למשל לקשר עם אוניות בלב ים). לאות הנפוצה 'ו' משמש בקוד מורס הקוד '...-'. וזוהי לאות 'ע' משמש הקוד '---'.

יישום גישה זו במחשבים נהוג כאשר נדרש חיסכון בנפח האחסון, אך עקב מורכבותה נדרשת השקעה של זמן מעבד לפענוח הקוד. תכניות לדחיסת נתונים מיישמות גישה זו. נפח האחסון המינימלי מושג באמצעות קוד הופמן.

טבלת המרה בין בסיסי מספרים נפוצים:

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	עשרוני:
10	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	הקסדצימלי:
20	17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	אוקטלי:
10000	1111	1110	1101	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001	0000	בינארי:

ראו גם

- תקשורת נתונים
- מונחים בחומרה
- מיתוג

קישורים חיצוניים

- השוואה בין בינארי לדצימלי (<http://www.pcguide.com/intro/fun/bindec-c.html>) (באנגלית)
- מהו קילובייט - הפשרה במשפט (<http://arstechnica.com/news.ars/post/20060630-7174.html>) (באנגלית)
- סיבית (<http://mathworld.wolfram.com/Bit.html>), באתר MathWorld (באנגלית)

הערות שוליים

- מבוא למערכות מחשב ואסמבלי (<http://cse.proj.ac.il/assembly/st-book-pdf.pdf>) עמ' 51
- A Mathematical Theory of Communication (<http://math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf>)
- Denning, P. J., & Bell, T. (2012). The Information Paradox. American Scientist, 100(6), 470-477.

דף זה נערך לאחרונה ב־13:50, 22 ביולי 2019.

הטקסט מוגש בכפוף לרישיון Creative Commons ייחוס-שיתוף זהה 3.0; ייתכן שישנם תנאים נוספים. ר' את תנאי השימוש לפרטים.

