Non-positional codes

Error-detecting code

Error-correcting code

Gray code

0  0000  only one bit change
1  0001
2  0011  only one bit change
3  0010  only one bit change

Gray codes can be built by "mirroring"

Q5

(1) 8_{10} = ( )_2

(2) Find the 2's complement of this; i.e., represent
-8_{10} \text{ in the 2's complement number system. This example is on p. 20 of the textbook.}

(1) \quad 00001000 = a \quad \text{(do by repeated division)}
(2) \quad a^* = 11111000 \quad \text{(flip bits to the left of the rightmost 1)}
(3) \quad b = 00010011 \quad \text{(do by repeated division)}
(4) \quad 00010011 +
\quad \begin{array}{c}
\hline
00010011 \\
11111000
\end{array}
\quad \overline{10001011} \quad \Rightarrow b - a = 00001011 \quad \text{(discard carry bit)}
Boolean algebra (Chs 2 & 3) (Combinational circuits)

Who was George Boole?
What did he write? — Please find out the exact title of his 1868 book.

We study Boolean algebra for its use in digital logic design, and especially for the design of combinational circuits.

Boolean variables may take two values which we represent as 0 and 1 (other representations are: T and F, t and f, true and false, yes and no, y and n, Y and N).
Boolean variables are usually represented by upper case Roman letters (A, B, ..., x, y, ...).

You indicate that a Boolean variable has value 1 by writing it lower case.

You indicate that a Boolean variable has value 0 by writing it with a prime.

Ex.: \( A = 1 \) is written \( a \)

\( A = 0 \) is written \( a' \) (\( \bar{a} \), \( \neg a \), \( \neg a \)) is common.

A pair of a Boolean variable and its value is called a literal. So, \( a \) and \( a' \) are literals.
The basic operations of Boolean algebra are AND, OR, and complement.

Complement (complementation) is indicated by '.

So, 0' = 1 1' = 0

\[ X' = 1 \text{ if } X = 0 \]
\[ X' = 0 \text{ if } X = 1 \]

This defines a new Boolean expression, from another one, namely the complement of a Boolean expression.

\[ X \rightarrow \overline{X} \rightarrow X' \]

\[ \text{negation circuit element (complementation)} \]

You "feed" \( X \) to the negation circuit element ("negator", or "inverter") and you get its complement \( X' \).
AND operation

\[ A \cdot B = A \text{ and } B = A \cap B = A + B \in AB \]

Most common and used in our text.

OR operation

\[ A + B = A \text{ or } B = A \cup B \]

Used by our text.