به نام خدا

مقدمه مؤلف

کتاب حاضر حاصل چندین سال تدریس در دانشگاه‌ها و مراکز آموزش عالی می‌باشد.

ویرایش جدید این کتاب مرجع بسیار مناسبی برای داوطلبان کنکور کارشناسی ارشد می‌باشد. به طوریکه با مطالعه این کتاب دیگر نیاز به هیچ منبعی نیست. کتاب در ۸ فصل تنظیم شده است و انتهای هر فصل تست‌های رسمی و تألیفی آمده است و پاسخ جامع و کاملی به تست‌ها داده شده است. قبل از بررسی تست‌های هر فصل ابتدا مطالب آن فصل را با دقت مطالعه کنید و سپس تست‌ها را کامل بررسی کنید، زیرا در این صورت تست‌ها را به راحتی پاسخ خواهید داد.

فصل ۸ در سال‌های اخیر در کنکورهای رسمی مطرح نشده است. ولی بهتر است فصل ۸ مطالعه شود. برای مطالعه عمیق‌تر اعداد و جمع‌کننده‌ها و ضرب‌کننده‌ها، فصل اول کتاب معماری کامپیوتر را حتماً مطالعه بفرمایید.

دوستان عزیزی در آماده‌سازی این کتاب مرا یاری کردند که از همه آنها سپاسگزارم. اگر پیشنهاد یا انتقادی در مورد این کتاب یا سایر کتاب‌های تألیف اینجانب داشتید لطفاً مطرح کنید.

هادی یوسفی – زمستان ۱۳۹۹
hadi_yusefi@ yahoo.com
۰۹۱۲۱۷۸۸۵۳۴

فهرست مطالب

فصل ۱ سیستم‌های اعداد و تبدیل مبنا

ما برای نمایش اعداد از مبنای ۱۰ (decimal) استفاده می‌کنیم. در سیستم‌های دیجیتال از مبنای ۲ (binary) استفاده می‌شود. مبناهای ۸ (octal) و ۱۶ (hexadecimal) نیز برخی مواقع برای نمایش اعداد مفید هستند. به طور کلی عدد a در مبنای r را می‌توان به شکل $a=(a_{n-1}a_{n-2}...a_1a_0/a_{-1}a_{-2}..a_{-m})_r$ نمایش داد. این نمایش دارای n رقم صحیح $a_{n-1},..,a_1,a_0$ است و دارای m رقم اعشار $a_{-1},a_{-2},..,a_{-m}$ می‌باشد. در مبنای r هر رقم در بازه $[0,r-1]$ است یعنی به ازای هر $-m \le i \le n-1$ رابطه $0 \le a_i \le r-1$ برقرار می‌باشد. مثلاً در مبنای ۱۰ ارقام ۰ تا ۹ مجاز هستند، در مبنای ۲ ارقام ۰ و ۱، در مبنای ۸ ارقام ۰ تا ۷ و در مبنای ۱۶ ارقام ۰ تا ۱۵ مجاز می‌باشند که ۱۰ را با A ، ۱۱ را با B ، ۱۲ را با C ، ۱۳ را با D ، ۱۴ را با E و ۱۵ را با F نمایش می‌دهیم. هر رقم مثل a_i در مبنای r دارای ارزش r^i می‌باشد. برای تبدیل عدد a از مبنای r به مبنای ۱۰، هر رقم را در ارزش آن رقم ضرب می‌کنیم و مقادیر بدست آمده را جمع می‌کنیم:

$$a=(a_{n-1}a_{n-2}...a_1a_0/a_{-1}a_{-2}...a_{-m})_r$$

$$=a_{n-1}\times r^{n-1}+a_{n-2}\times r^{n-2}+...+a_0\times r^0+a_{-1}\times r^{-1}+..+a_{-m}\times r^{-m}=\sum_{i=-m}^{n-1} a_i\times r^i$$

مثال ۱: اعداد $(147/3)_8$ و $(257/2)_8$ و $(1FA/C)_{16}$ و $(1011001/101)_2$ را به مبنای ۱۰ تبدیل کنید.

جواب مثال ۱:

$$(147/3)_8 = 1\times 8^2 + 4\times 8^1 + 7\times 8^0 + 3\times 8^{-1} = 64 + 32 + 7 + \frac{3}{8} = (103/375)_{10}$$

$$(257/2)_8 = 2\times 8^2 + 5\times 8^1 + 7\times 8^0 + 2\times 8^{-1} = 128 + 40 + 7 + \frac{2}{8} = (175/25)_{10}$$

$$(1FA/C)_{16} = 1\times 16^2 + 15\times 16^1 + 10\times 16^0 + 12\times 16^{-1} = 256 + 240 + 10 + \frac{12}{16} = (506/75)_{10}$$

$$(1011001/101)_2 = 1\times 2^6 + 0\times 2^5 + 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0 +$$
$$1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3} = 64 + 16 + 8 + 1 + \frac{1}{2} + \frac{1}{8} = (89/625)_{10}$$

مثال ۲: بزرگترین عدد n رقمی صحیح در مبنای r را بیابید.

جواب مثال ۲: برای درک بهتر، بزرگترین عدد ۳ رقمی در مبنای ۱۰ برابر ۹۹۹ می‌باشد که می‌توان $10^3 - 1$ نوشت. بزرگترین عدد ۳ رقمی مبنای ۲ برابر $(111)_2$ است که اگر به مبنای ۱۰ تبدیل شود برابر $2^3 - 1 = (7)_{10} = 4 + 2 + 1$ می‌باشد. پس بزرگترین عدد n رقمی در مبنای r برابر است با:

$$\Big(\underbrace{(r-1)(r-1)\ldots(r-1)}_{n \text{ بار } (r-1)}\Big)_r = (r-1)\times r^{n-1} + (r-1)\times r^{n-2} + \ldots + (r-1)\times r^0$$
$$= (r-1)\times(r^{n-1} + r^{n-2} + \ldots + r^0) = (r-1)\times\frac{r^n - 1}{r-1} = r^n - 1$$

توجه: تصاعد هندسی به شکل $\sum_{i=0}^{n-1} ax^i = a + ax + ax^2 + \ldots + ax^{n-1}$ است که a جمله اول است و x قدر نسبت (یعنی عدد ثابتی که در هر جمله ضرب می‌شود و جمله بعدی حاصل می‌شود) است و تعداد جملات برابر n است. حاصل این تصاعد برابر $\frac{a(1-x^n)}{1-x} = \frac{a(x^n - 1)}{x-1}$ می‌باشد و اگر $n \to \infty$ و $|x|<1$ آنگاه $a + ax + ax^2 + \ldots = \frac{a}{1-x}$

مثال ۳: بزرگترین اعشار m رقمی مبنای r را بیابید.

جواب مثال ۳: به عنوان نمونه، بزرگترین اعشار ۳ رقمی در مبنای ۱۰ برابر $۱-۱۰^{-۳}=۰/۹۹۹$ می‌باشد. بزرگترین اعشار m رقمی در مبنای r برابر است با:

$$\left(۰/\underbrace{(r-۱)(r-۱)...(r-۱)}_{(r-۱)\ \text{بار}\ m}\right)_r=(r-۱)\times r^{-۱}+(r-۱)\times r^{-۲}+...+(r-۱)\times r^{-m}$$

$$=(r-۱)\times(r^{-۱}+r^{-۲}+...+r^{-m})=(r-۱)\times\frac{r^{-۱}(۱-r^{-m})}{۱-r^{-۱}}$$

$$=(r-۱)\times\frac{(۱-r^{-m})}{r-۱}=۱-r^{-m}$$

تبدیل از مبنای ۱۰ به سایر مبناها: برای تبدیل از مبنای ۱۰ به مبنای $r\neq۰$، قسمت صحیح عدد را به r تقسیم می‌کنیم، باقیماندهٔ تقسیم، کم ارزش‌ترین رقم صحیح حاصل است. خارج قسمت را مجدداً به r تقسیم می‌کنیم و باقیماندهٔ تقسیم دومین رقم صحیح حاصل است (رقم با ارزش $r^۱$) و به همین ترتیب عمل تقسیم را ادامه می‌دهیم و باقیماندهٔ هر تقسیم را انتخاب می‌کنیم.

برای درک درستی این روش، فرض کنید حاصل تبدیل برابر عدد صحیح $b=(a_{n-۱}a_{n-۲}...a_۱a_۰)_r$ می‌باشد.

این عدد در مبنای ۱۰ برابر است با:

$$b=a_{n-۱}\times r^{n-۱}+...+a_۱\times r^۱+a_۰$$

که باقیماندهٔ عدد b به r برابر $a_۰$ (یعنی کم‌ارزش‌ترین رقم مبنای r) می‌باشد. خارج قسمت تقسیم b به r برابر است با

$$q_۱=\frac{b}{r}=a_{n-۱}\times r^{n-۲}+a_{n-۲}\times r^{n-۳}+..+a_۲\times r^۱+a_۱$$

باقیماندهٔ تقسیم $q_۱$ به r برابر $a_۱$ (یعنی دومین رقم کم ارزش مبنای r) می‌باشد. و به همین ترتیب با هر تقسیم به r، باقیماندهٔ تقسیم، ارقام عدد در مبنای r هستند.

مثال ۴: عدد $(۳۰۱)_{۱۰}$ را به مبنای ۸ تبدیل کنید.

جواب مثال ۴: این عدد را به ۸ تقسیم می‌کنیم و باقیمانده تقسیم $a_۰$ است. خارج قسمت تقسیم را به ۸ تقسیم می‌کنیم و باقیمانده برابر $a_۱$ است و به همین ترتیب:

$$۳۰۱=۸\times۳۷+۵\rightarrow a_۰=۵$$

$۳۷ = ۸ \times ۴ + ۵ \rightarrow a_۱ = ۵$

$۴ = ۸ \times ۰ + ۴ \rightarrow a_۲ = ۴$

پس $(۳۰۱)_{۱۰} = (۴۵۵)_۸$.

توجه: در تقسیم عدد a به r، اگر خارج قسمت برابر q و باقیمانده برابر $a_۰$ باشد $(۰ \le a_۰ < r)$ می‌توان نوشت: $a = r \times q + a_۰$

مثال ۵: عدد $(۵۳)_{۱۰}$ را به مبنای ۲ تبدیل کنید.

جواب مثال ۵: ۵۳ را به ۲ تقسیم می‌کنیم، باقیمانده $a_۰$ است. خارج قسمت را به ۲ تقسیم می‌کنیم، باقیمانده $a_۱$ است و به همین ترتیب:

$۵۳ = ۲ \times ۲۶ + ۱ \rightarrow a_۰ = ۱$

$۲۶ = ۲ \times ۱۳ + ۰ \rightarrow a_۱ = ۰$

$۱۳ = ۲ \times ۶ + ۱ \rightarrow a_۲ = ۱$

$۶ = ۲ \times ۳ + ۰ \rightarrow a_۳ = ۰$

$۳ = ۲ \times ۱ + ۱ \rightarrow a_۴ = ۱$

$۱ = ۲ \times ۰ + ۱ \rightarrow a_۵ = ۱$

پس $(۵۳)_{۱۰} = (۱۱۰۱۰۱)_۲$ ■

برای تبدیل قسمت اعشار عدد از مبنای ۱۰ به مبنای r، قسمت اعشار را در r ضرب می‌کنیم و رقم صحیح حاصلضرب برابر $a_{-۱}$ (یعنی اولین رقم بعد از ممیز در جواب) می‌باشد. سپس قسمت اعشار حاصل را مجدداً در r ضرب می‌کنیم و رقم صحیح حاصلضرب برابر $a_{-۲}$ (یعنی دومین رقم بعد از ممیز در جواب) می‌باشد. برای درکِ درستیِ این روش فرض کنید حاصلِ تبدیلِ قسمت اعشارِ عدد برابرِ $c = (۰/a_{-۱}a_{-۲}...a_{-m})_r$ است. این عدد در مبنای ۱۰ برابر است با: $c = a_{-۱} \times r^{-۱} + a_{-۲} \times r^{-۲} + .. + a_{-m} \times r^{-m}$. که اگر c را در r ضرب کنیم مقدار $c \times r = a_{-۱} + \underbrace{a_{-۲} \times r^{-۱} + ... + a_{-m} \times r^{-m+۱}}_{c_۱} = a_{-۱} + c_۱$ بدست می‌آید که $a_{-۱}$ رقم صحیح حاصلضرب است که اولین رقم بعد از ممیز در عدد c است. حال اگر $c_۱$ را مجدداً در r ضرب کنیم مقدار $c_۱ \times r = a_{-۲} + a_{-۳} \times r^{-۱} + ... + a_{-m} \times r^{-m+۲}$ حاصل می‌شود که رقم صحیح حاصلضرب یعنی $a_{-۲}$، دومین رقم بعد از ممیز در عدد c است.

مثال ۶: عدد $(۰/۶۲۵)_{۱۰}$ را به باینری (مبنای ۲) تبدیل کنید.

جواب مثال ۶: ۰/۶۲۵ را در ۲ ضرب می‌کنیم، قسمت صحیح حاصل برابر $a_{-۱}$ است. قسمت اعشار را مجدداً در ۲ ضرب می‌کنیم، قسمت صحیح حاصل برابر $a_{-۲}$ است. عمل ضرب را تا زمانی انجام می‌دهیم که یا اعشار به صفر برسد و یا اعشار، تکراری شود:

$$\begin{array}{l} ۰/۶۲۵\times ۲ = ۱/۲۵ \rightarrow a_{-۱} = ۱ \\ ۰/۲۵\times ۲ = ۰/۵ \rightarrow a_{-۲} = ۰ \qquad \Rightarrow \quad (۰/۶۲۵)_{۱۰} = (۰/۱۰۱)_{۲} \\ ۰/۵\times ۲ = ۱/۰ \rightarrow a_{-۳} = ۱ \end{array}$$

مثال ۷: عدد $(۰/۷)_{۱۰}$ را به مبنای ۲ تبدیل کنید.

جواب مثال ۷: با توجه به روالِ تبدیل، این عدد در مبنای ۲ متناوب می‌باشد و قسمت $a_{-۲}a_{-۳}a_{-۴}a_{-۵}$ تا بی‌نهایت تکرار می‌شود:

$$\boxed{\begin{array}{l} ۰/۷\times ۲ = ۱/۴ \rightarrow a_{-۱} = ۱ \\ ۰/۴\times ۲ = ۰/۸ \rightarrow a_{-۲} = ۰ \\ ۰/۸\times ۲ = ۱/۶ \rightarrow a_{-۳} = ۱ \\ ۰/۶\times ۲ = ۱/۲ \rightarrow a_{-۴} = ۱ \\ ۰/۲\times ۲ = ۰/۴ \rightarrow a_{-۵} = ۰ \\ ۰/۴\times ۲ \rightarrow \text{repeat from } a_{-۲} \end{array}} \Rightarrow \quad (۰/۷)_{۱۰} = (۰/۱\underbrace{۰۱۱۰}\underbrace{۰۱۱۰}\underbrace{۰۱۱۰}\ldots)_{۲}$$

مثال ۸: عدد $(۳۰۱/۲)_{۱۰}$ را به مبنای ۸ تبدیل کنید.

جواب مثال ۸: با توجه به مثال ۴، $(۴۵۵)_{۸} = (۳۰۱)_{۱۰}$. برای تبدیل $(۰/۲)_{۱۰}$ به مبنای ۸ از روش ضرب متوالی استفاده می‌کنیم:

$$\boxed{\begin{array}{l} ۰/۲\times ۸ = ۱/۶ \rightarrow a_{-۱} = ۱ \\ ۰/۶\times ۸ = ۴/۸ \rightarrow a_{-۲} = ۴ \\ ۰/۸\times ۸ = ۶/۴ \rightarrow a_{-۳} = ۶ \\ ۰/۴\times ۸ = ۳/۲ \rightarrow a_{-۴} = ۳ \\ ۰/۲\times ۸ \rightarrow \text{repeat from } a_{-۱} \end{array}} \begin{array}{l} \Rightarrow (۰/۲)_{۱۰} = (۰/\underbrace{۱۴۶۳}\underbrace{۱۴۶۳}\underbrace{۱۴۶۳}\ldots)_{۸} \\ \Rightarrow (۳۰۱/۲)_{۱۰} = (۴۵۵/\underbrace{۱۴۶۳}\underbrace{۱۴۶۳}\ldots)_{۸} \end{array}$$

مثال ۹: کدام یک از اعداد $(۰/۱)_{۱۰}$ و $(۰/۱۵)_{۱۰}$ و $(۰/۷۷۵)_{۱۰}$ و $(۰/۸۷۵)_{۱۰}$، در مبنای ۴ خاتمه‌پذیر هستند؟

جواب مثال ۹: شرط لازم برای آنکه، عدد در مبنای ۲ یا ۴ یا ۸ یا ۱۶ خاتمه‌پذیر باشد آن است که سمت راست عدد، رقم ۵ باشد و هر بار که ضرب می‌شود نیز سمت راست حاصل ۵ باشد پس $(۰/۱)_{۱۰}$

در مبنای ۴ خاتمه‌پذیر نیست.

$$۰/۱\times ۴=۰/۴ \quad و \quad ۰/۴\times ۴=۱/۶ \quad و \quad ۰/۶\times ۴=۲/۴ \quad و \quad ۰/۴\times ۴\rightarrow \text{repeat}$$

رقم سمت راستِ ۰/۱۵ برابر ۵ است ولی با توجه به $۰/۱۵\times ۴=۰/۶$ چون رقم سمت راست ۰/۶ برابر ۵ نیست پس خاتمه‌پذیر نیست. ۰/۷۷۵ نیز در مبنای ۴ متناوب است زیرا $۰/۷۷۵\times ۴=۳/۱$ که سمت راست ۳/۱ برابرِ ۵ نیست. عدد ۰/۸۷۵ در مبنای ۴ مختوم است:

$$۰/۸۷۵\times ۴=۳/۵ \quad و \quad ۰/۵\times ۴=۲/۰$$

نکته

برای تبدیل عدد صحیح a از مبنای ۱۰ به مبنای ۲ روش تقسیم متوالی کمی وقت‌گیر است. روش سریعتر این است که عدد a را به شکل مجموع اعداد متمایز به شکل ۲^i بنویسیم و سپس اگر $۲^۰$ وجود داشت یعنی بیت سمت راست حاصل، ۱ است و در غیر این صورت ۰ است. اگر $۲^۱$ وجود داشت یعنی بیت دوم از سمت راست، ۱ است و در غیر این صورت ۰ است و به همین ترتیب.

مثال ۱۰: اعداد $(۳۴)_{۱۰}$ و $(۷۹)_{۱۰}$ و $(۳۰۱)_{۱۰}$ و $(۵۳)_{۱۰}$ را به مبنای ۲ تبدیل کنید.

جواب مثال ۱۰:

$$(۳۴)_{۱۰}=۳۲+۲=۲^۵+۲^۱=۱\times ۲^۵+۰\times ۲^۴+۰\times ۲^۳+۰\times ۲^۲+۱\times ۲^۱+۰\times ۲^۰=(۱۰۰۰۱۰)_۲$$

$$(۷۹)_{۱۰}=۶۴+۸+۴+۲+۱=۲^۶+۲^۳+۲^۲+۲^۱+۲^۰=(۱۰۰۱۱۱۱)_۲$$

$$(۳۰۱)_{۱۰}=۲۵۶+۳۲+۸+۴+۱=۲^۸+۲^۵+۲^۳+۲^۲+۲^۰=(۱۰۰۱۰۱۱۰۱)_۲$$

$$(۵۳)_{۱۰}=۳۲+۱۶+۴+۱=۲^۵+۲^۴+۲^۲+۲^۰=(۱۱۰۱۰۱)_۲$$

توجه: اعداد به شکل ۲^i را به خاطر بسپارید:

$$۲^۰=۱ \text{ و } ۲^۱=۲ \text{ و } ۲^۲=۴ \text{ و } ۲^۳=۸ \text{ و } ۲^۴=۱۶ \text{ و } ۲^۵=۳۲ \text{ و } ۲^۶=۶۴ \text{ و } ۲^۷=۱۲۸ \text{ و}$$

$$۲^۸=۲۵۶ \text{ و } ۲^۹=۵۱۲ \text{ و } ۲^{۱۰}=۱۰۲۴ \text{ و } ۲^{۱۱}=۲۰۴۸ \text{ و } ۲^{۱۲}=۴۰۹۶ \text{ و } ۲^{۱۳}=۸۱۹۲ \text{ و}$$

$$۲^{۱۴}=۱۶۳۸۴ \text{ و } ۲^{۱۵}=۳۲۷۶۸ \text{ و } ۲^{۱۶}=۶۵۵۳۶ \text{ و } ۲^{۱۷}=۱۳۱۰۷۲ \text{ و } \ldots \blacksquare$$

تاکنون تبدیل از مبنای ۱۰ به مبنای $r\neq ۱۰$ و برعکس را بررسی کردیم. حال می‌خواهیم تبدیل بین دو مبنا که هیچ یک مبنای ۱۰ نیستند را بررسی کنیم. به یک مثال توجه کنید.

مثال ۱۱: عدد $(۲۳۱/۳)_۴$ را به مبنای ۷ تبدیل کنید.

جواب مثال ۱۱: ابتدا عدد را به مبنای ۱۰ تبدیل می‌کنیم:

$$(۲۳۱/۳)_۴=۲\times ۴^۲+۳\times ۴^۱+۱\times ۴^۰+۳\times ۴^{-۱}=۴۵+\frac{۳}{۴}=(۴۵/۷۵)_{۱۰}$$

حال، حاصل را به مبنای ۷ تبدیل می‌کنیم:

$$(۴۵=۷\times۶+۳\rightarrow a_۰=۳\ ,\ ۶=۷\times۰+۶\rightarrow a_۱=۶)\Rightarrow(۴۵)_{۱۰}=(۶۳)_۷$$
$$(۰/۷۵\times۷=۵/۲۵\rightarrow a_{-۱}=۵\,,۰/۲۵\times۷=۱/۷۵\rightarrow a_{-۲}=۱,\text{repeat})\Rightarrow(۰/۷۵)_{۱۰}=(۰/\underbrace{۵}\underbrace{۱}\underbrace{۵}\underbrace{۱}...)_۷$$

پس جواب برابر $(۶۳/\underbrace{۵}\underbrace{۱}\underbrace{۵}\underbrace{۱}...)_۷$ می‌باشد. ■

همانطور که دیدیم برای تبدیل از مبنای ۴ به مبنای ۷، ابتدا عدد را به مبنای ۱۰ تبدیل کردیم و سپس نتیجه را به مبنای ۷ تبدیل کردیم. برخی مبناها را می‌توان مستقیم به هم تبدیل کرد. به طور کلی مبنای r را می‌توان مستقیم به مبنای r^n تبدیل کرد و برعکس. مثلاً مبنای ۲ به $۴=۲^۲$ و برعکس، مبنای ۲ به $۸=۲^۳$ و برعکس، مبنای ۲ به $۱۶=۲^۴$ و برعکس، مبنای ۴ بـه $۱۶=۴^۲$ و برعکس، مبنای ۳ به $۹=۳^۲$ و برعکس. برای تبدیل از مبنای r^n به مبنای r، به ازای هـر رقـم در مبنای r^n، باید n رقم در مبنای r قرار دهیم. برای تبدیل از مبنای r به مبنای r^n، بـه ازای هـر n رقم مبنای r، باید یک رقم در مبنای r^n قرار دهیم، اگر لازم بود به سمت چـپ عـدد، یـا سـمت راست بعد از ممیز، صفر اضافه می‌کنیم تا بتوان دسته‌های n رقمی ساخت.

مثال ۱۲: تبدیل مبناهای مقابل را انجام دهید:

$$(۴۵۷/۶۳۲)_۸=(\ )_۲ \quad و \quad (F۴E/۸C)_{۱۶}=(\ )_۲ \quad و \quad (B۲DA/۸۷)_{۱۶}=(\ )_۴ \quad و$$
$$(۷۳۸۵۲)_۹=(\ )_۳ \quad و \quad (۱۰۱۱/۱۰۱۱)_۲=(\ )_۸ \quad و \quad (۱۰۱۱۱/۱۰۱۱۱)_۲=(\ )_{۱۶} \quad و \quad (۲۱۰۱۱/۲۱۲)_۳=(\ )_۹$$

جواب مثال ۱۲: برای تبدیل از مبنای ۸ به مبنای ۲، به ازای هر رقم ۳ بیت قرار می‌دهیم:

$$(۴۵۷/۶۳۲)_۸=(\underbrace{۱۰۰}\underbrace{۱۰۱}\underbrace{۱۱۱}/\underbrace{۱۱۰}\underbrace{۰۱۱}\underbrace{۰۱۰})_۲$$

برای تبدیل از مبنای ۱۶ به مبنای ۲، به ازای هر رقم ۴ بیت قرار می‌دهیم:

$$(F۴E/۸C)_{۱۶}=(\underbrace{۱۱۱۱}\underbrace{۰۱۰۰}\underbrace{۱۱۱۰}/\underbrace{۱۰۰۰}\underbrace{۱۱۰۰})_۲$$

برای تبدیل از مبنای ۱۶ به مبنای ۴، به ازای هر رقم، ۲ رقم قرار می‌دهیم:

$$(B۲DA/۸۷)_{۱۶}=(\underbrace{۲۳}\underbrace{۰۲}\underbrace{۳۱}\underbrace{۲۲}/\underbrace{۲۰}\underbrace{۱۳})_۴$$

برای تبدیل از مبنای ۹ به مبنای ۳، به ازای هر رقم، ۲ رقم قرار می‌دهیم:

$$(۷۳۸۵۲)_۹=(۲۱۱۰۲۲۱۲۰۲)_۳$$

برای تبدیل از مبنای ۲ به مبنای ۸، به ازای هر ۳ بیت، یک رقم قرار می‌دهیم. قسمت صحیح را از سمت راست و قسمت اعشار را از سمت چپ به دسته‌های ۳ رقمی تقسیم می‌کنیم و اگر لازم بود سمت چپ عدد یا سمت راست بعد از ممیز صفر قرار می‌دهیم:

$$(۱۰۱۱/۱۰۱۱)_۲ = (\underbrace{۰۰۱}\underbrace{۰۱۱}/\underbrace{۱۰۱}\underbrace{۱۰۰})_۲ = (۱۳/۵۴)_۸$$

برای تبدیل از مبنای ۲ به ۱۶ به ازای هر ۴ بیت، یک رقم قرار می‌دهیم:

$$(۱۰۱۱۱/۱۰۱۱۱)_۲ = (\underbrace{۰۰۰۱}\underbrace{۰۱۱۱}/\underbrace{۱۰۱۱}\underbrace{۱۰۰۰})_۲ = (۱۷/B۸)_{۱۶}$$

برای تبدیل از مبنای ۳ به مبنای ۹ به ازای هر ۲ رقم، یک رقم قرار می‌دهیم:

$$(۲۱۰۱۱/۲۱۲)_۳ = (\underbrace{۰۲}\underbrace{۱۰}\underbrace{۱۱}/\underbrace{۲۱}\underbrace{۲۰})_۳ = (۲۳۴/۷۶)_۹$$

برای درک بهترِ تبدیلات فوق به جداول زیر توجه کنید:

مبنای ۳	مبنای ۹
۰۰	۰
۰۱	۱
۰۲	۲
۱۰	۳
۱۱	۴
۱۲	۵
۲۰	۶
۲۱	۷
۲۲	۸

مبنای ۴	مبنای ۲	مبنای ۱۶
۰۰	۰۰۰۰	۰
۰۱	۰۰۰۱	۱
۰۲	۰۰۱۰	۲
۰۳	۰۰۱۱	۳
۱۰	۰۱۰۰	۴
۱۱	۰۱۰۱	۵
۱۲	۰۱۱۰	۶
۱۳	۰۱۱۱	۷
۲۰	۱۰۰۰	۸
۲۱	۱۰۰۱	۹
۲۲	۱۰۱۰	A
۲۳	۱۰۱۱	B
۳۰	۱۱۰۰	C
۳۱	۱۱۰۱	D
۳۲	۱۱۱۰	E
۳۳	۱۱۱۱	F

مبنای ۲	مبنای ۸
۰۰۰	۰
۰۰۱	۱
۰۱۰	۲
۰۱۱	۳
۱۰۰	۴
۱۰۱	۵
۱۱۰	۶
۱۱۱	۷

لازم به ذکر است که عدد $(xy)_۳$ معادل $(۳x+y)_۹$ می‌باشد $(۰\leq x,y\leq ۲)$ و یا عدد $(xy)_۴$ معادل $(۴x+y)_{۱۶}$ می‌باشد $(۰\leq x,y\leq ۳)$.

محاسبات در مبناهای مختلف

در جمع دو عدد، به عدد اول Augend و به عدد دوم Addend می‌گوییم. برای جمع مبنای r دو عدد را زیر هم می‌نویسیم و ابتدا ارقام موجود در ستون صفر (ستون با ارزش $r^۰$) را با هم جمع می‌کنیم و اگر رقم نقلی (carry) تولید شد، آن را به ستون یک (ستون با ارزش $r^۱$) منتقل می‌کنیم و این عمل را برای ستون یک، تکرار می‌کنیم و به همین ترتیب برای سایر ستون‌ها انجام می‌دهیم. به چند مثال توجه کنید:

Carries	۱	۱ ۱ ۱ ۱ ۱ ۱	۱ ۱	۱ ۱
Augend	$(۲\ A\ ۵\ ۸)_{۱۶}$	$(۱\ ۰\ ۱\ ۱\ ۰\ ۱)_۲$	$(۲\ ۷\ ۶)_۸$	$(۲\ ۷\ ۶)_{۱۰}$
Addend	$+(۷\ ۱\ D\ ۰)_{۱۶}$	$+(۰\ ۱\ ۰\ ۱\ ۱\ ۱)_۲$	$+(۳\ ۵\ ۷)_۸$	$+(۳\ ۵\ ۷)_{۱۰}$
sum	$(۹\ C\ ۲\ ۸)_{۱۶}$	$(۱\ ۰\ ۰\ ۰\ ۱\ ۰\ ۰)_۲$	$(۶\ ۵\ ۵)_۸$	$(۶\ ۳\ ۳)_{۱۰}$

دقت کنید در مبنای ۸، جمع دو رقم ۶ و ۷ برابر $(۱۳)_{۱۰}$ یعنی $(۱۵)_۸$ می‌شود که ۵ برابر حاصل‌جمع (sum) است که در همان ستون نوشته می‌شود و ۱ رقم نقلی (carry) است که به ستون بعدی منتقل می‌شود. و یا در مبنای ۱۶، جمع دو رقم ۵ و $D=(۱۳)_{۱۰}$ برابر $(۱۲)_{۱۶}=(۱۸)_{۱۰}$ می‌شود که ۲ حاصلجمع است و ۱ رقم نقلی.

در عمل تفریق دو عدد $A-B$، به عدد اول minuend (مفروق منه) و به عدد دوم Subtrahend (مفروق) گویند. برای تفریق، دو عدد را زیر هم می نویسیم، و از ستون سمت راست شروع به تفریق می‌کنیم و هر جا رقمِ مفروق منه از رقمِ مفروق، کوچکتر باشد باید از ستونِ با ارزشِ بالاتر، قرض بگیریم. هر واحد قرض در مبنای r، معادل r واحد است به تفریق‌های زیر توجه کنید.

Borrows		۷	۱ ۱۰
Borrows	E ۱۱	$۳\ \cancel{۱۰}\ ۱۵$	$۰\ \cancel{۱۰}\ ۱۰\ ۰\ \cancel{۰}\ ۱۰$
Minuend	$(۹\ \cancel{F}\ \cancel{۱}\ B)_{۱۶}$	$(\cancel{۴}\ \cancel{۰}\ \cancel{۵})_۸$	$(\cancel{۱}\ \cancel{۰}\ \cancel{۰}\ \cancel{۱}\ \cancel{۱}\ \cancel{۰}\ ۱)_۲$
Subtrahend	$(۴\ A\ ۳\ ۶)_{۱۶}$	$(۲\ ۵\ ۶)_۸$	$(\ \ ۱\ ۰\ ۱\ ۱\ ۱)_۲$
Difference	$(۵\ ۴\ E\ ۵)_{۱۶}$	$(۱\ ۲\ ۷)_۸$	$(۰\ ۱\ ۱\ ۰\ ۱\ ۱\ ۰)_۲$

دقت کنید در مبنای ۲ هر قرض معادل $(۲)_{۱۰}=(۱۰)_۲$ می‌باشد و یا در مبنای ۸ هر قرض معادل $(۸)_{۱۰}=(۱۰)_۸$ می‌باشد و در مبنای ۱۶ هر قرض معادل $(۱۶)_{۱۰}=(۱۰)_{۱۶}$ می‌باشد. پس $(۱۷)_{۱۰}=(۱۱)_{۱۶}$ می‌باشد.

در عمل ضرب، عدد اول را مضروب (ضرب شونده multiplicand) و عـدد دوم را مضـروب فیـه (ضرب‌کننده multiplier) گویند. به چند عمل ضرب توجه کنید:

$$\begin{array}{rl}
(45)_{10} & \text{multiplicand} \\
\times\ (64)_{10} & \text{multiplier} \\
\hline
+\ 180 & \\
270 & \text{partial products} \\
\hline
(2880)_{10} & \text{product}
\end{array}
\qquad\qquad
\begin{array}{r}
(45)_8 \\
\times\ (64)_8 \\
\hline
1 \\
224 \\
+\ 336 \\
\hline
(3604)_8
\end{array}$$

$$\begin{array}{rrrrrrr}
 & & & (5 & C & 2 & A)_{16} \\
 & & \times & (7 & 1 & D & 0)_{16} \\
\hline
 & 1 & 1 & 0 & 0 & 0 & 0 \\
 & 4 & A & E & 2 & 2 & \\
 & 5 & C & 2 & A & & \\
2 & 8 & 5 & 2 & 6 & & \\
\hline
(2 & 8 & F & 9 & 6 & C & 2 & 0)_{16}
\end{array}$$

$$\begin{array}{rrrrrrrr}
 & & & & (1 & 1 & 0 & 1)_2 \\
 & & & \times & (1 & 0 & 1 & 1)_2 \\
\hline
 & & & 1 & 1 & 1 & 0 & 1 \\
 & & 1 & 1 & 1 & 0 & 1 & \\
 & 1 & 0 & 0 & 0 & 0 & & \\
1 & 1 & 1 & 0 & 1 & & & \\
\hline
1 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{array}
\qquad\qquad
\begin{array}{rrrrrrr}
 & & & (1 & 1 & 1)_2 \\
 & & \times & (1 & 1 & 1)_2 \\
\hline
 & & & 10 & 1 & & \\
 & & 10 & 1 & 1 & 1 & 1 \\
 & 10 & 1 & 1 & 1 & 1 & \\
1 & 1 & 1 & 1 & 1 & & \\
\hline
1 & 1 & 0 & 1 & 0 & 0 & 1
\end{array}$$

در عمل تقسیم دو عدد، عدد اول را مقسوم (Dividend) و عدد دوم را مقسوم‌علیـه (Divisor) گویند. به عمل تقسیم زیر توجه کنید:

$$\begin{array}{l}
\text{Dividend}\ (1010111)_2 \;\big|\; (1001)_2\ \text{Divisor} \\
\quad -\ 1001 \qquad\qquad\quad (10010)_2\ \text{Quotient} \\
\hline
\qquad\quad 1011 \\
\qquad\ -\ 1001 \\
\hline
\qquad\qquad (101)_2\ \text{Remainder}
\end{array}$$

مکمل‌ها

مکمل‌ها برای نمایش اعداد منفی و همچنین انجام عمل تفریق کاربرد دارند. ما ابتدا مکمل‌ها را معرفی می‌کنیم و سپس کاربرد آنها را بررسی می‌کنیم. در مبنای r دو نوع مکمل تعریف شده است:

مکمل $r-۱$ (Reduced (Diminished Radix) complement) و مکمل r (Radix complement).
مثلاً در مبنای ۲، مکمل ۱ و مکمل ۲ تعریف شده‌اند. در مبنای ۱۰، مکمل ۹ و مکمل ۱۰ تعریف می‌شوند. در مبنای ۸، مکمل ۷ و مکمل ۸ معرفی شده‌اند.

فرض کنید عدد $(a)_r$ دارای n رقم صحیح و m رقم اعشار است آنگاه مکمل $r-۱$ عدد a که به شکل $[a]_{r-۱}$ نشان می‌دهیم برابر است با:

$$\boxed{[a]_{r-۱} = r^n - r^{-m} - (a)_r}$$

و مکمل r عدد a که به شکل $[a]_r$ نشان می‌دهیم برابر است با:

$$\boxed{[a]_r = r^n - (a)_r}$$

مثال ۱۳: مکمل ۹ و مکمل ۱۰ عدد $a = (۴۷۶/۲۳)_{۱۰}$ را بیابید.

جواب مثال ۱۳: مکمل ۹ عدد a برابر است با:

$$[a]_۹ = ۱۰^۳ - ۱۰^{-۲} - (۴۷۶/۲۳)_{۱۰} = (۹۹۹/۹۹)_{۱۰} - (۴۷۶/۲۳)_{۱۰} = (۵۲۳/۷۶)_{۱۰}$$

پس برای یافتن مکمل ۹ عدد $(a)_{۱۰}$ کافی است هر رقم را از ۹ کم کنیم.

مکمل ۱۰ عدد a برابر است با:

$$[a]_{۱۰} = ۱۰^۳ - (۴۷۶/۲۳)_{۱۰} = (۱۰۰۰)_{۱۰} - (۴۷۶/۲۳)_{۱۰} = (۵۲۳/۷۷)_{۱۰}$$

یعنی برای یافتن مکمل ۱۰ عدد $(a)_{۱۰}$، ابتدا مکمل ۹ این عدد را می‌یابیم و سپس رقم سمت راست آن را با ۱ جمع می‌کنیم.

مثال ۱۴: مکمل ۷ و مکمل ۸ عدد $a = (۴۷۶/۲۳)_۸$ را بیابید.

جواب مثال ۱۴: مکمل ۷ عدد a برابر است با:

$$[a]_۷ = ۸^۳ - ۸^{-۲} - (۴۷۶/۲۳)_۸ = (۷۷۷/۷۷)_۸ - (۴۷۶/۲۳)_۸ = (۳۰۱/۵۴)_۸$$

پس برای یافتن مکمل ۷ عدد $(a)_۸$ کافی است هر رقم را از ۷ کم کنیم.

مکمل ۸ عدد a برابر است با:

$$[a]_۸ = ۸^۳ - (۴۷۶/۲۳)_۸ = (۱۰۰۰)_۸ - (۴۷۶/۲۳)_۸ = (۳۰۱/۵۵)_۸$$

یعنی برای یافتن مکمل ۸ عدد $(a)_۸$، ابتدا مکمل ۷ این عدد را می‌یابیم و سپس رقم سمت راست آن را با ۱ جمع می‌کنیم.

نتیجه: برای یافتن مکمل $r-۱$ عدد $(a)_r$ کافی است همهٔ ارقام عدد $(a)_r$ را از $r-۱$ کم کنیم.

نتیجه: برای یافتن مکمل r عدد $(a)_r$ ابتدا مکمل $r-۱$ عدد $(a)_r$ را می‌یابیم و سپس رقم سمت راست آن را یک واحد اضافه می‌کنیم.

مثال ۱۵: مکمل ۹ و مکمل ۱۰ عدد $(۲۰۴۰۰)_{۱۰}$ را بیابید.

جواب مثال ۱۵: برای یافتن مکمل ۹، همه ارقام را از ۹ کم می‌کنیم. پس مکمل ۹ عدد $(۲۰۴۰۰)_{۱۰}$ برابر $(۷۹۵۹۹)_{۱۰}$ می‌باشد. برای یافتن مکمل ۱۰، به رقم سمت راست مکمل ۹، یک واحد می‌افزاییم. پس مکمل ۱۰ عدد $(۲۰۴۰۰)_{۱۰}$ برابر $(۷۹۶۰۰)_{۱۰}$ می‌باشد.

نتیجه: برای یافتن مکمل r عدد $(a)_r$، از سمت راست حرکت کرده و صفرهای سمت راست عدد را در صورت وجود تغییر نمی‌دهیم، اولین رقم غیرصفر را از r و سایر ارقام را از $r-۱$ کم می‌کنیم.

مثال ۱۶: مکمل ۹ و ۱۰ عدد $(۲۰۳۰۰)_{۱۰}$، مکمل ۷ و ۸ عدد $(۲۰۳۰۰)_۸$، مکمل ۱ و ۲ عدد $(۱۰۱۰۰)_۲$ و مکمل ۹ و ۱۰ عدد $(۴۵۰/۲۷)_{۱۰}$ را بیابید.

جواب مثال ۱۶:

۹۹۹۹۹
$(۲۰۳۰۰)_{۱۰} \xrightarrow{\text{مکمل ۹}} (۷۹۶۹۹)_{۱۰}$

۹۹۱۰
$(۲۰۳۰۰)_{۱۰} \xrightarrow{\text{مکمل ۱۰}} (۷۹۷۰۰)_{۱۰}$

۷۷۷۷۷
$(۲۰۳\ ۰۰)_۸ \xrightarrow{\text{مکمل ۷}} (۵۷۴۷۷)_۸$

۷۷۸
$(۲۰۳۰۰)_۸ \xrightarrow{\text{مکمل ۸}} (۵۷۵۰۰)_۸$

۱۱۱۱۱
$(۱۰۱۰۰)_۲ \xrightarrow{\text{مکمل ۱}} (۰۱۰۱۱)_۲$

۱۱۲
$(۱۰۱۰۰)_۲ \xrightarrow{\text{مکمل ۲}} (۰۱۱۰۰)_۲$

۹۹۹۹۹
$(۴۵۰/۲۷)_{۱۰} \xrightarrow{\text{مکمل ۹}} (۵۴۹/۷۲)_{۱۰}$

۹۹۹۹۱۰
$(۴۵۰/۲۷)_{۱۰} \xrightarrow{\text{مکمل ۱۰}} (۵۴۹/۷۳)_{۱۰}$

نتیجه: مکمل ۱ عدد $(a)_۲$ به این معنی است که همهٔ صفرها به یک و همهٔ یک‌ها به صفر تبدیل شوند یعنی مکمل ۱ عدد $(a)_۲$ برابر $(\overline{a})_۲$ است.

نتیجه: برای یافتن مکمل ۲ عدد $(a)_۲$، صفرهای سمت راست و اولین ۱ از سمت راست عوض نمی‌شوند و سایر بیت‌ها عوض می‌شوند. مثلاً مکمل ۲ عدد $(۱۰۱۱۰۰)_۲$ برابر $(۰۱۰۱۰۰)_۲$ می‌باشد و یا مکمل ۲ عدد $(۱۰۰۰)_۲$ برابر $(۱۰۰۰)_۲$ می‌باشد. و یا مکمل ۲ عدد $(۱۱۱۱)_۲$ برابر $(۰۰۰۱)_۲$ می‌باشد.

نمایش اعداد علامت‌دار در مبنای ۲

برای نمایش اعداد مثبت و منفی در مبنای ۲، سه روش ارائه می‌دهیم:

۱- روش علامت مقدار (اندازه علامت sign magnitude)

۲- روش مکمل ۱ (1's complement)

۳- روش مکمل ۲ (2's complement)

این سه روش، اعداد مثبت را به یک شکل نمایش می‌دهند و تفاوت این روش‌ها در نمایش اعداد منفی است. در این سه روش، بیت سمت چپ اعداد مثبت برابر صفر و اعداد منفی برابر یک است.

مثلاً با ۴ بیت عدد $(۰۱۱۰)_۲$ در این سه روش برابر ۶+ می‌باشد ولی نمایش ۶- در این سه روش متفاوت است. در **روش علامت مقدار**، بیت سمت چپ فقط نشان دهندهٔ علامت است و ارزش ندارد و برای قرینه کردن یک عدد، کافی است بیت سمت چپ آن را تغییر دهیم. مثلاً با ۴ بیت، $+۶=(۰۱۱۰)_۲$ و $-۶=(۱۱۱۰)_۲$. عدد صفر در این روش دارای دو نمایش متفاوت است یعنی $+۰=(۰۰۰۰)_۲$ و $-۰=(۱۰۰۰)_۲$ و این یکی از معایب این روش است. ماکزیمم عدد قابل نمایش با ۴ بیت برابر $+۷=(۰۱۱۱)_۲$ و مینیمم عدد قابل نمایش $-۷=(۱۱۱۱)_۲$ می‌باشد.

در **روش مکمل ۱**، برای یافتن قرینهٔ یک عدد، آن را مکمل ۱ می‌کنیم یعنی همهٔ بیت‌های آن را عوض می‌کنیم مثلاً $+۶=(۰۱۱۰)_۲$ و $-۶=(۱۰۰۱)_۲$. عدد صفر در این روش دارای دو نمایش متفاوت است یعنی $+۰=(۰۰۰۰)_۲$ و $-۰=(۱۱۱۱)_۲$ و این یکی از معایب این روش است. ماکزیمم عدد قابل نمایش با ۴ بیت برابر $+۷=(۰۱۱۱)_۲$ و مینیمم عدد قابل نمایش $-۷=(۱۰۰۰)_۲$ می‌باشد.

در **روش مکمل ۲**، برای یافتن قرینهٔ یک عدد، آن را مکمل ۲ می‌کنیم. مثلاً $+۶=(۰۱۱۰)_۲$ و $-۶=(۱۰۱۰)_۲$. عدد صفر در این روش فقط یک نمایش دارد $۰=(۰۰۰۰)_۲$ پس در این روش می‌توان یک عدد بیشتر نمایش داد. ماکزیمم عدد قابل نمایش با ۴ بیت برابر $+۷=(۰۱۱۱)_۲$ و مینیمم عدد قابل نمایش $-۸=(۱۰۰۰)_۲$ می‌باشد. برای درک بهتر که چرا $-۸=(۱۰۰۰)_۲$، به عملیات زیر توجه کنید:

$$(-۸)+(+۷)=(۱۰۰۰)_۲+(۰۱۱۱)_۲=(۱۱۱۱)_۲=-(۰۰۰۱)_۲=-۱$$

جدول زیر، با ۴ بیت، همهٔ اعداد قابل نمایش در این ۳ روش را نشان داده است:

signed Decimal	Sign magnitude system	one's complement system	Two's complement system
+۷	۰۱۱۱	۰۱۱۱	۰۱۱۱
+۶	۰۱۱۰	۰۱۱۰	۰۱۱۰
+۵	۰۱۰۱	۰۱۰۱	۰۱۰۱
+۴	۰۱۰۰	۰۱۰۰	۰۱۰۰
+۳	۰۰۱۱	۰۰۱۱	۰۰۱۱
+۲	۰۰۱۰	۰۰۱۰	۰۰۱۰
+۱	۰۰۰۱	۰۰۰۱	۰۰۰۱
+۰	۰۰۰۰	۰۰۰۰	۰۰۰۰
−۰	۱۰۰۰	۱۱۱۱	−
−۱	۱۰۰۱	۱۱۱۰	۱۱۱۱
−۲	۱۰۱۰	۱۱۰۱	۱۱۱۰
−۳	۱۰۱۱	۱۱۰۰	۱۱۰۱
−۴	۱۱۰۰	۱۰۱۱	۱۱۰۰
−۵	۱۱۰۱	۱۰۱۰	۱۰۱۱
−۶	۱۱۱۰	۱۰۰۱	۱۰۱۰
−۷	۱۱۱۱	۱۰۰۰	۱۰۰۱
−۸	−	−	۱۰۰۰

مثال ۱۷: عدد ۸ بیتی $a=(۱۰۰۱۱۱۰۰)_۲$ را به مبنای ۱۰ تبدیل کنید با فرض اینکه

الف) این عدد در سیستم بی‌علامت باشد. ب) این عدد در سیستم علامت مقدار باشد.

ج) این عدد در سیستم مکمل ۱ باشد. د) این عدد در سیستم مکمل ۲ باشد.

جواب مثال ۱۷: الف)

$$a=(۱۰۰۱۱۱۰۰)_۲=۱\times۲^۷+۱\times۲^۴+۱\times۲^۳+۱\times۲^۲=۱۵۶$$

ب) بیت سمت چپِ عدد، برابر ۱ است پس عدد منفی است:

$$a=(۱۰۰۱۱۱۰۰)_۲=-(۰۰۱۱۱۰۰)_۲=-(۲^۴+۲^۳+۲^۲)=-۲۸$$

ج) بیت سمت چپ عدد، برابر ۱ است پس عدد منفی است، برای یافتن مقدار آن، عدد را مکمل ۱ می‌کنیم که مثبت شود و سپس مقدار عدد را می‌یابیم:

$$a=(۱۰۰۱۱۱۰۰)_۲=-(۰۱۱۰۰۰۱۱)_۲=-(۲^۶+۲^۵+۲^۱+۲^۰)=-۹۹$$

د) بیت سمت چپ عدد، برابر ۱ است، پس عـدد منفـی اسـت. بـرای یـافتن مقـدار آن، عـدد را مکمل ۲ می‌کنیم که مثبت شود و سپس مقدار عدد را می‌یابیم:

$$a=(10011100)_2=-(01100100)_2=-(2^6+2^5+2^2)=-100$$

نکته

بـرای یـافتن مقـدار عـدد در سیسـتم‌هـای مختلـف مـی‌تـوان فرمـول ارائـه داد. عـدد n بیتـی $a=(a_{n-1}a_{n-2}\dots a_1a_0)_2$ را در نظر بگیرید. در سیستم‌های مختلف مقدار این عدد برابر است با:

سیستم بی‌علامت: $a=a_{n-1}\times 2^{n-1}+a_{n-2}\times 2^{n-2}+\dots+a_1\times 2^1+a_0\times 2^0$

سیستم علامت مقدار: $a=(-1)^{a_{n-1}}\times(a_{n-2}\times 2^{n-2}+\dots+a_1\times 2^1+a_0\times 2^0)$

سیستم مکمل ۱: $a=a_{n-1}\times -(2^{n-1}-1)+a_{n-2}\times 2^{n-2}+\dots+a_1\times 2^1+a_0\times 2^0$

سیستم مکمل ۲: $a=a_{n-1}\times -2^{n-1}+a_{n-2}\times 2^{n-2}+\dots+a_1\times 2^1+a_0\times 2^0$

مثال ۱۸: مقدار عدد ۸ بیتی $a=(10011100)_2$ را در سیستم‌های مختلف با توجه به فرمول‌های ارائه شده بیابید:

جواب مثال ۱۸:

سیستم بی‌علامت: $a=\begin{array}{cccccccc} 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ (1 & 0 & 0 & 1 & 1 & 1 & 0 & 0)_2 \end{array}=2^7+2^4+2^3+2^2=156$

سیستم علامت مقدار: $a=\begin{array}{cccccccc} & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ (1 & 0 & 0 & 1 & 1 & 1 & 0 & 0)_2 \end{array}=-(2^4+2^3+2^2)=-28$

سیستم مکمل ۱: $a=\begin{array}{cccccccc} -(2^7-1) & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ (1 & 0 & 0 & 1 & 1 & 1 & 0 & 0)_2 \end{array}=-(2^7-1)+2^4+2^3+2^2=-99$

سیستم مکمل ۲: $a=\begin{array}{cccccccc} -2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ (1 & 0 & 0 & 1 & 1 & 1 & 0 & 0)_2 \end{array}=-2^7+2^4+2^3+2^2=-100$

مثال ۱۹: با توجـه بـه فرمـول‌هـای ارائـه شـده، مـاکزیمم و مینـیمم عـدد قابـل نمـایش بـا n بیـت در سیستم‌های مختلف را بیابید.

جواب مثال ۱۹: در سیستم بی‌علامت، مینیمم عدد قابل نمایش $(\underbrace{00\dots0}_{\text{n تا ۰}})_2=0$ و مـاکزیمم عـدد قابـل نمـایش $(\underbrace{11\dots1}_{\text{n تا یک}})_2=2^n-1$ اسـت. در سیسـتم علامـت مقـدار، مینـیمم عـدد

$-(۲^{n-۱}-۱)=(\underbrace{۱۱۱...۱۱}_{\text{n تا یک}})_۲$ و ماکزیمم $۲^{n-۱}-۱=(\underbrace{۰۱۱..۱}_{\text{n-۱ تا یک}})_۲$ است. در سیستم مکمل ۱، مینیمم عدد $-(۲^{n-۱}-۱)=(\underbrace{۱۰۰..۰}_{\text{n-۱ تا صفر}})_۲$ و ماکزیمم $۲^{n-۱}-۱=(\underbrace{۰۱۱...۱}_{\text{n-۱ تا یک}})_۲$ است. در سیستم مکمل ۲ مینیمم عدد $-۲^{n-۱}=(\underbrace{۱۰۰..۰}_{\text{n-۱ تا صفر}})_۲$ و ماکزیمم $۲^{n-۱}-۱=(\underbrace{۰۱۱..۱}_{\text{n-۱ تا یک}})$ می‌باشد.

پرچم‌ها (فلگ‌ها)

فلگ‌ها بیت‌هایی هستند که پس از انجام عملیاتی مثل جمع یا تفریق، مقداردهی می‌شوند. در این درس ۴ تا فلگ مهم هستند که بررسی می‌کنیم:

فلگ Z (Zero): این فلگ وقتی یک می‌شود که حاصل عملیات صفر باشد. و اگر حاصل عملیات مخالف صفر باشد، فلگ Z برابر صفر می‌شود.

فلگ N (Negative) یا S (Sign): این فلگ وقتی یک می‌شود که حاصل عملیات، منفی شود. اگر حاصل، مثبت باشد، فلگ S برابر صفر می‌شود.

فلگ V یا OF(Overflow): این فلگ وقتی یک می‌شود که حاصل عملیات سرریز باشد. یعنی حاصل در بازهٔ مجاز نباشد.

فلگ C(Carry): این فلگ وقتی یک می‌شود که از بیت سمت چپ، رقم نقلی خارج شود.

سیستم مکمل ۲

تقریباً تمام سیستم‌های دیجیتال از روش مکمل ۲ برای نمایش اعداد صحیح استفاده می‌کنند. همانطور که دیدیم در این سیستم، همهٔ بیت‌ها همانند سیستم بی‌علامت، دارای ارزش هستند فقط ارزش بیت سمت چپ، منفی است.

مثال ۲۰: مقدار اعداد $(۱۰۰۱)_۲$ و $(۱۱۰۰۱)_۲$ و $(۱۱۱۰۰۱)_۲$ و $(۱۱۱۱)_۲$ و $(۱۱۱۱۱۱)_۲$ و $(۰۱۱۰)_۲$ را با همان تعداد بیت‌های نوشته شده، در سیستم مکمل ۲ بیابید.

جواب مثال ۲۰:

$$(۱۰۰۱)_۲=-۲^۳+۲^۰=-۸+۱=-۷$$

$$(۱۱۰۰۱)_۲=-۲^۴+۲^۳+۲^۰=-۱۶+۸+۱=-۷$$

$$(۱۱۱۰۰۱)_۲=-۲^۵+۲^۴+۲^۳+۲^۰=-۳۲+۱۶+۸+۱=-۷$$

$$(۱۱۱۱)_۲=-۲^۳+۲^۲+۲^۱+۲^۰=-۸+۴+۲+۱=-۱$$

$$(۱۱۱۱۱۱)_۲=-۲^۵+۲^۴+۲^۳+۲^۲+۲^۱+۲^۰=-۱$$

$$(۰۱۱۰)_۲=۲^۲+۲^۱=+۶$$

نتیجه: در سیستم مکمل ۲، اگر بیت سمت چپ عدد، تکرار شود، مقدار عدد، عوض نمی‌شود. مـثلاً با ۴ بیت $-۷ = (۱۰۰۱)_۲$ و با ۸ بیت $-۷ = (۱۱۱۱۱۰۰۱)_۲$ و یـا بـا ۴ بیـت $+۶ = (۰۱۱۰)_۲$ و بـا ۸ بیت $+۶ = (۰۰۰۰۰۱۱۰)_۲$.

جمع در سیستم مکمل ۲: برای جمع دو عدد، دو عدد را زیر هم بنویسیم و جمع کنـیم! یعنـی مهم نیست که اعداد مثبت یا منفی هستند، همانند جمع دو عدد بی‌علامت، دو عدد سیستم مکمل ۲ را جمع می‌کنیم و از رقم نقلی خروجی صرفنظر می‌کنیم.

مثـال ۲۱: بـا ۴ بیـت عملیـات ۱۰۰۱+۰۱۱۱ و ۰۰۱۱+۰۱۰۰ و ۱۱۰۱+۱۱۰۰ و ۱۱۰۱+۰۱۱۰ و ۰۱۰۱+۰۱۱۰ و ۱۰۰۱+۱۰۱۰ را در سیستم مکمل ۲ انجام دهید و وضعیت پرچم‌ها را مشخص کنید.

جواب مثال ۲۱:

از بیت سمت چپ، نقلی خارج شده است: $C = ۱$	۱ ۱ ۱
حاصل عملیات صفر است : $Z = ۱$	$۱\ ۰\ ۰\ ۱ = -۷$
حاصل عملیات مثبت است : $S = ۰$	$+۰\ ۱\ ۱\ ۱ = +۷$
سرریز وجود ندارد. جمع -۷ با $+۷$ برابر صفر شده است : $V = ۰$	$۰\ ۰\ ۰\ ۰ = ۰$

از بیت سمت چپ، نقلی خارج نشده است: $C = ۰$	$۰\ ۰\ ۱\ ۱ = +۳$
حاصل عملیات صفر نیست : $Z = ۰$	$+۰\ ۱\ ۰\ ۰ = +۴$
حاصل عملیات مثبت است : $S = ۰$	$۰\ ۱\ ۱\ ۱ = +۷$
سرریز وجود ندارد : $V = ۰$	

از بیت سمت چپ، نقلی خارج شده است: $C = ۱$	۱
حاصل عملیات صفر نیست : $Z = ۰$	$۱\ ۱\ ۰\ ۱ = -۳$
حاصل عملیات منفی است : $S = ۱$	$+۱\ ۱\ ۰\ ۰ = -۴$
سرریز وجود ندارد : $V = ۰$	$۱\ ۰\ ۰\ ۱ = -۷$

از بیت سمت چپ، نقلی خارج شده است: $C = ۱$	۱
حاصل عملیات صفر نیست : $Z = ۰$	$۱\ ۱\ ۰\ ۱ = -۳$
حاصل عملیات مثبت است : $S = ۰$	$+۰\ ۱\ ۱\ ۰ = +۶$
سرریز وجود ندارد : $V = ۰$	$۰\ ۰\ ۱\ ۱ = +۳$

از بیت سمت چپ، نقلی خارج نشده است: $C=0$	①
حاصل عملیات صفر نیست : $Z=0$	$\begin{array}{r} 0\ 1\ 1\ 0 = +6 \\ +\ 0\ 1\ 0\ 1 = +5 \\ \hline 1\ 0\ 1\ 1 = -5 \end{array}$
حاصل عملیات منفی است : $S=1$	
سرریز می‌باشد. جمع دو عدد مثبت، منفی شده است : $V=1$	

از بیت سمت چپ، نقلی خارج شده است: $C=1$	$\begin{array}{r} 1\ 0\ 0\ 1 = -7 \\ +\ 1\ 0\ 1\ 0 = -6 \\ \hline 0\ 0\ 1\ 1 = +3 \end{array}$
حاصل عملیات صفر نیست : $Z=0$	
حاصل عملیات مثبت است : $S=0$	
سرریز می‌باشد. جمع دو عدد منفی، مثبت شده است : $V=1$	

نکته

اگر A و B در سیستم مکمل ۲ باشند آنگاه برای تشخیص سرریز $A+B$ دو روش می‌توان ارائه داد:

۱- هرگاه A و B مثبت باشند و حاصل جمع منفی باشد، یا A و B منفی باشند و حاصل جمع مثبت باشد آنگاه سرریز است و برعکس. توجه کنید اگر A و B مختلف‌العلامه باشند آنگاه $A+B$ سرریز نمی‌شود:

$$\begin{array}{rlllll} \text{carries} = c_n & c_{n-1} & c_{n-2} & \dots & c_1 & c_0 \\ A = & a_{n-1} & a_{n-2} & \dots & a_1 & a_0 \\ +\quad B = & b_{n-1} & b_{n-2} & \dots & b_1 & b_0 \\ \hline & s_{n-1} & s_{n-2} & \dots & s_1 & s_0 \end{array}$$

$$[(a_{n-1}=b_{n-1}=0,\ s_{n-1}=1)\ \text{or}\ (a_{n-1}=b_{n-1}=1,\ s_{n-1}=0)] \Leftrightarrow V=1$$

پس می‌توان فرمول تشخیص سرریز را به شکل $V=\bar{a}_{n-1}\cdot\bar{b}_{n-1}\cdot s_{n-1}+a_{n-1}\cdot b_{n-1}\cdot\bar{s}_{n-1}$ نوشت.
(با عملیات بولی در فصل بعد آشنا می‌شویم)

۲- هرگاه رقم نقلی وارد شده به بیت سمت چپ (c_{n-1}) با رقم نقلی خارج شده از آن (c_n) مساوی نباشد آنگاه سرریز است و برعکس:

$$(c_n \neq c_{n-1} \Leftrightarrow V=1)$$

پس می‌توان فرمول $V=c_n \oplus c_{n-1}$ را نیز برای تشخیص سرریز ارائه داد. (عملگر $\oplus$ علامت xor می‌باشد و وقتی یک می‌شود که دو عملوند آن، مخالف باشند)

نکته

بررسی کنید فرمول $V=\bar{a}_{n-1}\cdot\bar{b}_{n-1}\cdot c_{n-1}+a_{n-1}\cdot b_{n-1}\cdot\bar{c}_{n-1}$ نیز برای تشخیص سرریز صحیح است.

تفریق سیستم مکمل ۲: برای انجام تفریق $A-B$ می‌توان دو روش ارائه داد:

روش ۱: A را با مکمل ۲ عدد B جمع کنیم و از رقم نقلی خروجی (در صورت وجود) صرفنظر کنیم:

$$A-B=A+(-B)=A+\overline{B}+1$$

لازم به ذکر است $-B$ یعنی مکمل ۲ عدد B یعنی $\overline{B}+1$.

روش ۲: B را زیر A بنویسیم و با قرض گرفتن کم کنیم.

مثال ۲۲: با ۴ بیت تفریق ۱۰۰۱−۱۱۰۱ را با دو روش در سیستم مکمل ۲ انجام دهید و وضعیت فلگ‌ها را مشخص کنید.

جواب مثال ۲۲: روش ۱:

$$\underbrace{1101}_{-3}-\underbrace{1001}_{-7}=1101+\overline{1001}+1=1101+0110+1=1101+0111=\underset{\text{carryout}}{\underset{\downarrow}{1}}\,0100=+4$$

فلگ‌ها عبارتند از: $V=0$ و $S=0$ و $Z=0$ و $C=1$

روش ۲: در این روش فلگ C یعنی قرض. در واقع C وقتی یک می‌شود که چپ‌ترین بیت نیاز به قرض داشته باشد.

$$\begin{array}{r} 1101 \\ -1001 \\ \hline 0100 \end{array} \qquad C=0\,,Z=0\,,S=0\,,V=0$$

توجه کنید فلگ C در این دو روش متفاوت است ولی سایر فلگ‌ها یکسان هستند.

مثال ۲۳: تفریق ۴ بیتی ۰۰۰۰−۱۱۰۱ را در سیستم مکمل ۲ با دو روش انجام دهید و فلگ‌ها را مشخص کنید.

جواب مثال ۲۳: روش ۱:

$$1101-0000=1101+\overline{0000}+1=1101+1111+1=\underset{\text{carryout}}{\underset{\downarrow}{1}}\,1101$$

فلگ‌ها عبارتند از: $C=1$ و $Z=0$ و $S=1$ و $V=0$

روش ۲: $1101-0000=1101$

فلگ‌ها عبارتند از: $C=0$ و $Z=0$ و $S=1$ و $V=0$

مثال ۲۴: تفریق ۴ بیتی ۱۱۰۱−۱۰۰۱ را در سیستم مکمل ۲ با دو روش انجام دهید و فلگ‌ها را مشخص کنید.

جواب مثال ۲۴: روش ۱:

$$\underbrace{۱۰۰۱}_{-۷}-\underbrace{۱۱۰۱}_{-۳}=۱۰۰۱+۰۰۱۰+۱=\underbrace{۱۱۰۰}_{-۴}$$

$C=۰$ و $Z=۰$ و $S=۱$ و $V=۰$

روش ۲:

$$\begin{array}{rrrr} & ۱۰ & & \\ & \cancel{۰} & ۱۰ & \\ \cancel{۱} & \cancel{۰} & ۰ & ۱ \\ -۱ & ۱ & ۰ & ۱ \\ \hline ۱ & ۱ & ۰ & ۰ \end{array}$$

بیت سمت چپ قرض گرفته است پس $C=۱$. سایر فلگ‌ها مثل روش ۱ است.

نکته

اگر A و B در سیستم مکمل ۲ باشند. پس از انجام عمل $A-B$ اگر $A\geq B$ آنگاه انتظار داریم که حاصل مثبت باشد یعنی $S=۰$ البته به شرطی که سرریز پیش نیاید یعنی $V=۰$. حال اگر سرریز اتفاق بیافتد یعنی $V=۱$، حاصل منفی می‌شود یعنی $S=۱$. پس شرط لازم و کافی برای آنکه $A\geq B$ این است که $S=V$ یعنی $S\oplus V=۰$. همچنین شرط لازم و کافی برای آنکه $A<B$ این است که $S\neq V$ یعنی $S\oplus V=۱$

نکته

اگر A و B دو عدد n بیتی سیستم مکمل ۲ باشند، حاصل $A\pm B$ وقتی سرریز است که $A\pm B>۲^{n-۱}-۱$ یا $A\pm B<-۲^{n-۱}$. یعنی حاصل از بازهٔ مجاز خارج باشد. زیرا می‌دانیم با n بیت بازهٔ مجاز $[-۲^{n-۱},۲^{n-۱}-۱]$ می‌باشد.

نکته

اگر A و B دو عدد n بیتی سیستم مکمل ۲ باشند، $A-B$ در صورتی سرریز است که A مثبت باشد و B منفی باشد و حاصل منفی باشد، یا A منفی باشد و B مثبت باشد و حاصل مثبت باشد.

سیستم مکمل ۱

در این سیستم، همهٔ بیت‌ها دارای ارزش هستند. اگر عدد n بیتی باشد، ارزش بیت سمت چپ $-(۲^{n-۱}-۱)$ می‌باشد. ارزش سایر بیت‌ها همانند سیستم بی‌علامت است.

مثال ۲۵: مقدار اعداد $(۱۰۰۱)_۲$ و $(۱۱۰۰۱)_۲$ و $(۱۱۱۰۰۱)_۲$ و $(۱۱۱۱)_۲$ و $(۱۱۱۱۱۱)_۲$ و $(۰۱۱۰)_۲$ را با همان تعداد بیت‌های نوشته شده، در سیستم مکمل ۱ بیابید.

جواب مثال ۲۵:

$(۱۰۰۱)_۲=-۷+۱=-۶$

$(۱۱۰۰۱)_۲=-۱۵+۸+۱=-۶$

$(۱۱۱۰۰۱)_۲=-۳۱+۱۶+۸+۱=-۶$

$(۱۱۱۱)_۲=-۷+۴+۲+۱=-۰$

$(۱۱۱۱۱۱)_۲=-۳۱+۱۶+۸+۴+۲+۱=-۰$

$(۰۱۱۰)_۲=۴+۲=+۶$

نتیجه: در سیستم مکمل ۱، همانند سیستم مکمل ۲ اگر بیت سمت چپ عدد را تکرار کنیم، مقدار عدد حفظ می‌شود.

جمع در سیستم مکمل ۱: همانند سیستم مکمل ۲، عمل جمع را انجام می‌دهیم. با این تفاوت که، رقم نقلی خروجی را مجدداً با حاصل جمع، جمع می‌کنیم.

مثال ۲۶: حاصل جمع‌های ۰۱۰۱+۱۰۰۱ و ۱۰۱۰+۰۱۱۰ و ۱۱۰۰+۱۰۱۱ و ۱۰۱۰+۱۰۰۱ را در سیستم مکمل ۱ با ۴ بیت بیابید.

جواب مثال ۲۶:

$$\begin{array}{r} ۰۱۰۱=+۵ \\ +\ ۱۰۰۱=-۶ \\ \hline ۱۱۱۰=-۱ \end{array} \qquad \begin{array}{r} ۱۰۱۰=-۵ \\ +\ ۰۱۱۰=+۶ \\ \hline (۱)\ ۰۰۰۰ \\ +\ ۱ \\ \hline ۰۰۰۱=+۱ \end{array} \qquad \begin{array}{r} ۱۱۰۰=-۳ \\ +\ ۱۰۱۱=-۴ \\ \hline (۱)\ ۰۱۱۱ \\ +\ ۱ \\ \hline ۱۰۰۰=-۷ \end{array} \qquad \begin{array}{r} ۱۰۱۰=-۵ \\ +\ ۱۰۰۱=-۶ \\ \hline (۱)\ ۰۰۱۱ \\ +\ ۱ \\ \hline ۰۱۰۰=+۴ \end{array}$$

آخرین جمع، سرریز است چون جمع دو عدد منفی، مثبت شده است.

نکته

در سیستم مکمل ۱، شرط لازم و کافی برای وقوع سرریز آن است که جمع دو عدد منفی، مثبت شود یا جمع دو عدد مثبت، منفی شود.

تفریق سیستم مکمل ۱: برای انجام عمل $A-B$، A را با مکمل ۱ عدد B $(\bar{B})$ جمع می‌کنیم و رقم نقلی خروجی را با حاصل جمع، جمع می‌کنیم.

مثال ۲۷: با ۴ بیت تفریق ۱۰۰۱−۱۰۰۰ را در سیستم مکمل ۱ انجام دهید.

جواب مثال ۲۷:

$$\begin{array}{r} ۱۰۰۱ \\ +\ ۰۱۱۱ \\ \hline (۱)\ ۰۰۰۰ \\ +\ ۱ \\ \hline ۰۰۰۱ \end{array} \Rightarrow \underset{-۶}{۱۰۰۱} - \underset{-۷}{۱۰۰۰} = \underset{+۱}{۰۰۰۱}$$

سیستم علامت مقدار

در این سیستم، بیت سمت چپ فقط علامت است و فاقد ارزش است. جمع و تفریق در این سیستم برای کامپیوتر کمی پیچیده است. مثلاً با ۴ بیت می‌خواهیم ۰۰۱۱+۱۱۰۱ را انجام دهیم. نمی‌توان این دو عدد را زیر هم نوشت و با هم جمع کرد. باید ابتدا علامت‌ها بررسی شوند و چون دو عدد هم علامت نیستند پس باید ارزش‌ها از هم کم شوند و در نهایت علامتِ عددِ با قدر مطلقِ بزرگتر، انتخاب شود. در واقع ۵−=۱۱۰۱ و ۳+=۰۰۱۱ که حاصل جمعِ این دو عدد برابرِ ۲−=۱۰۱۰ می‌باشد. ولی انجام این عملیات برای کامپیوتر پیچیده‌تر از سایر سیستم‌هاست.

سیستم بی‌علامت

فرض کنید A و B دو عدد n بیتی بی‌علامت هستند و می‌خواهیم $A+B$ را محاسبه کنیم و برای حاصل جمع، n بیت فضا در نظر گرفته‌ایم. حاصل جمعِ دو عددِ n بیتی، حداکثر $n+۱$ بیتی است، پس اگر حاصل جمع را در یک فضای n بیتی قرار دهیم آنگاه در صورتی که رقم نقلیِ خروجی، یک شود، یعنی سرریز اتفاق افتاده است. برای انجام عمل تفریق $A-B$ دو روش وجود دارد: **روش ۱:** عدد اول را با مکمل ۲ عدد دوم جمع کنیم یعنی $A-B=A+\overline{B}+۱$. **روش ۲:** دو عدد را با شیوهٔ قلم و کاغذ و با قرض گرفتن، تفریق کنیم.

اگر تفریق با روش جمع انجام شود یعنی $A-B=A+\overline{B}+۱$ آنگاه

$C_{out}=۱\leftrightarrow A\geq B$ و $C_{out}=۰\leftrightarrow A<B$

اگر تفریق با روش قرض انجام شود آنگاه:

$C_{out}=۱\leftrightarrow A<B$ و $C_{out}=۰\leftrightarrow A\geq B$

کدهای باینری برای اعداد دسیمال

می‌خواهیم ارقام دسیمال ۰ تا ۹ را به صورت باینری کدگذاری کنیم. برای این منظور حداقل ۴ بیت نیاز داریم (برای کد کردن n حالت یا n تا کاراکتر نیاز به $\lceil \lg n \rceil$ بیت می‌باشد) در جدول زیر، تعدادی کد ۴ بیتی برای نمایش ۰ تا ۹ آمده است. طبیعی‌ترینِ این کدها، کد BCD (یا NBCD: Natural Binary Coded Decimal) است که ۰ تا ۹ را به صورت اعداد بی‌علامت باینری ۴ بیتی، یعنی از ۰۰۰۰ تا ۱۰۰۱ نمایش می‌دهد. پس کلمه کدهای ۱۰۱۰ تا ۱۱۱۱ در BCD استفاده نمی‌شوند (به هر رشتهٔ دودویی مثل ۰۰۰۰ یا ۰۰۰۱ یک کلمه کد (code word) گفته می‌شود)

Decimal digit	BCD(۸۴۲۱)	excess۳	$۸۴\overline{۲}\,\overline{۱}$	۲۴۲۱	۶۳۱۱
۰	۰۰۰۰	۰۰۱۱	۰۰۰۰	۰۰۰۰	۰۰۰۰
۱	۰۰۰۱	۰۱۰۰	۰۱۱۱	۰۰۰۱	۰۰۰۱
۲	۰۰۱۰	۰۱۰۱	۰۱۱۰	۰۰۱۰	۰۰۱۱
۳	۰۰۱۱	۰۱۱۰	۰۱۰۱	۰۰۱۱	۰۱۰۰
۴	۰۱۰۰	۰۱۱۱	۰۱۰۰	۰۱۰۰	۰۱۰۱
۵	۰۱۰۱	۱۰۰۰	۱۰۱۱	۱۰۱۱	۰۱۱۱
۶	۰۱۱۰	۱۰۰۱	۱۰۱۰	۱۱۰۰	۱۰۰۰
۷	۰۱۱۱	۱۰۱۰	۱۰۰۱	۱۱۰۱	۱۰۰۱
۸	۱۰۰۰	۱۰۱۱	۱۰۰۰	۱۱۱۰	۱۰۱۱
۹	۱۰۰۱	۱۱۰۰	۱۱۱۱	۱۱۱۱	۱۱۰۰

لازم به ذکر است، تعداد کدهای ۴ بیتی زیادی برای ۰ تا ۹ می‌توان ساخت، بررسی کنید تعداد این نوع کدها $\frac{۱۶!}{۶!}=۲۹۰۵۹۴۳۰۴۰۰$ می‌باشد. ما در این جدول فقط ۵ تا از این کدها را آورده‌ایم.

در این جدول، به غیر از کد excess۳ (مازاد ۳، افزونی ۳) سایر کدها، وزن‌دار (weighted) هستند. مثلاً BCD وزن‌دار است و وزن آن به شکل ۸۴۲۱ است، بنابراین $(۰۱۱۱)_{BCD}=۰\times۸+۱\times۴+۱\times۲+۱\times۱=۷$. به طور کلی اگر وزن بیت‌ها در کد C برابر $w_۳w_۲w_۱w_۰$ باشد آنگاه مقدار کلمهٔ کد $A=(a_۳a_۲a_۱a_۰)_C$ در کد C برابر است با:

$$A=a_۳\times w_۳+a_۲\times w_۲+a_۱\times w_۱+a_۰\times w_۰$$

مثلاً کد $۸۴\overline{۲}\,\overline{۱}$ وزن‌دار است و ارزش یا وزن با ارزش‌ترین بیت آن ۸ و بیت دوم ۴ و بیت سوم ۲- و بیت چهارم ۱- می‌باشد. پس:

$$(۰۱۱۱)_{۸۴\overline{۲}\,\overline{۱}}=۰\times۸+۱\times۴+۱\times-۲+۱\times-۱=۱$$

کد افزونی ۳ (excess3) از افزودن $(۰۰۱۱)_۲=۳$ به BCD، حاصل می‌شود.

مثلاً رقم ۵ در کد افزونی ۳ به شکل ۰۱۰۱+۰۰۱۱ یعنی ۱۰۰۰ نمایش داده می‌شود.

برخی کدها اصطلاحاً خود مکمل (self complement) هستند. کدی را خود مکمل گویند که اگر هر کلمهٔ کد را که به شکل دودویی است، مکمل کنیم (مکمل یک کنیم) آنگاه رقم نظیر آن کلمهٔ کد، مکمل ۹ می‌شود (یعنی از ۹ کم می‌شود) کد excess3 خود مکمل است، مثلاً $(۰۰۱۱)_{excess۳}=۰$ و $(۱۱۰۰)_{excess۳}=۹$ $=\overline{۰۰۱۱}$ یا $(۰۱۰۰)_{excess۳}=۱$ و $(۱۰۱۱)_{excess۳}=۸$.

در جدول فوق، کدهای excess3 و $۸۴\overline{۲}\,\overline{۱}$ و ۲۴۲۱ خود مکمل هستند. شرط لازم برای آنکه یک کد وزن‌دار بتواند خود مکمل باشد، آن است که جمع وزن‌های آن برابر ۹ باشد. مثلاً جمع

وزن‌های کد $۸\,۴\,\bar{۲}\,\bar{۱}$ برابر $۸+۴+(-۲)+(-۱)=۹$ می‌باشد و این کد خود مکمل است. جمع وزن‌های کد ۲۴۲۱ نیز برابر ۹ است.

دقت کنید excess۳ وزن‌دار نیست و ارتباطی به نکتهٔ گفته شده ندارد. لازم به ذکر است که خود مکمل بودن، یک مزیت است و در انجام محاسبات مثل تفریق، کدهای خود مکمل، سخت‌افزار کمتری نیاز دارند.

کدِ BCD کدِ معروفی است ولی خود مکمل نیست. برای یافتن مکمل ۹ در کد BCD، دو روش که قابل پیاده‌سازی است، ارائه می‌شود:

روش ۱: ابتدا کلمهٔ کد BCD را با ۶ جمع کنید و سپس حاصل را مکمل ۱ (not) کنید مثلاً:

$$۲=(۰۰۱۰)_{BCD}\xrightarrow{+۶}۱۰۰۰\xrightarrow{\text{مکمل ۱}}(۰۱۱۱)_{BCD}=۷$$

مکمل ۹

برای اثبات این روش، فرض کنید a یک کلمهٔ کد در BCD است، اگر با ۶ جمع شود، $a+۶$ حاصل می‌شود و حال اگر مکمل ۱ شود $۱۵-(a+۶)=۹-a$ حاصل می‌شود که مکمل ۹ کلمهٔ کد a می‌باشد.

روش ۲: ابتدا کلمهٔ کد BCD را مکمل ۱ کنید سپس نتیجه را با ده جمع کنید و از رقم نقلی خروجی، صرفنظر کنید مثلاً:

$$۲=(۰۰۱۰)_{BCD}\xrightarrow{\text{مکمل ۱}}۱۱۰۱\xrightarrow{+۱۰}\underset{\substack{\downarrow\\ \text{صرفنظر}}}{۱}(۰۱۱۱)_{BCD}=۷$$

مکمل ۹

برای اثبات این روش، فرض کنید a یک کلمهٔ کد در BCD است، اگر a را مکمل ۱ کنید، $۱۵-a$ حاصل می‌شود که حال اگر نتیجه را با ده جمع کنید $(۱۵-a)+۱۰=۲۵-a$ حاصل می‌شود که می‌توان نوشت $(۲۵-a)=۱۶+(۹-a)$ که ۱۶ رقم نقلی خروجی است (۱۰۰۰۰) که صرفنظر می‌شود و $(۹-a)$ مکمل ۹ کلمهٔ a می‌باشد.

مثال ۲۸: عدد ۶۸ را با ۸ بیت به صورت دودویی، BCD، excess3 و $۸\,۴\,\bar{۲}\,\bar{۱}$ نمایش دهید.

جواب مثال ۲۸: برای نمایش دودویی، ۶۸ را به مبنای ۲ تبدیل می‌کنیم. ولی برای سایر کدهای گفته شده، باید به ازای هر رقم، ۴ بیت قرار دهیم:

$$۶۸=۶۴+۴=(۰۱۰۰۰۱۰۰)_{binary}$$

$$۶۸=(\underbrace{۰۱۱۰}_{۶}\underbrace{۱۰۰۰}_{۸})_{BCD}$$

$$۶۸ = (\underbrace{۱۰۰۱}_{۶}\underbrace{۱۰۱۱}_{۸})_{excess۳}$$

$$۶۸ = (\underbrace{۱۰۱۰}_{۶}\underbrace{۱۰۰۰}_{۸})_{۸۴\bar{۲}\bar{۱}}$$

(توجه کنید در کد افزونی ۳، (۱۰۰۱) معادل ۶ و (۱۰۱۱) معادل ۸ می‌باشد)

توجه: برای انجام عملیات جمع و تفریق در کدهای اعداد دسیمال، باید حاصل را تصحیح کنیم که در هر کد روش تصحیح متفاوت است. مثلاً در کد BCD پس از جمع دو رقم، باید حاصل بـا ۶ جمع شود که تصحیح شود مثلاً عمل جمع ۴ و ۸ را در کد BCD در نظر بگیرید:

$$\begin{array}{rl} & ۴ = (۰۱۰۰)_{BCD} \\ + & ۸ = (۱۰۰۰)_{BCD} \\ \hline & ۱۲ = (۱۱۰۰)_{binary} \\ + & ۶ = (۰۱۱۰)_{۲} \\ \hline & ۱۲ = (۱۰۰۱۰)_{BCD} \end{array}$$

حاصل جمع پس از تصحیح (جمع با ۶) برابر ۱۰۰۱۰ می‌باشد که برابر ۱۲ است.

مثال ۲۹: حاصل جمع $(۰۱۱۰۱۰۰۰)_{BCD} + (۰۱۰۰۱۰۰۱)_{BCD}$ را در کد BCD بیابید.

جواب مثـال ۲۹: در واقـع $(۰۱۱۰۱۰۰۰)_{BCD}$ برابـر ۶۸ و $(۰۱۰۰۱۰۰۱)_{BCD}$ برابر ۴۹ مـی‌باشـد و حاصلجمع ۴۹+۶۸ برابر ۱۱۷ می‌باشد که در کد BCD بـه شـکل $(۱۰۰۰۱۰۱۱۱)_{BCD}$ نمـایش داده می‌شود.

برای نمایش ارقام دسیمال غیر از کدهای ۴ بیتی که دیدیم، کدهای معـروف دیگـری کـه از ۴ بیت بزرگتر هستند نیز وجود دارد که در جدول زیر، چند نمونه آمده است:

Decimal digit	Biquinary	$\frac{۱}{۱۰}$ code	$\frac{۲}{۵}$ code ۰۷۴۲۱	$\frac{۲}{۷}$ code ۰۵۰۱۲۳۴
۰	۰۱۰۰۰۰۱	۱۰۰۰۰۰۰۰۰۰	۰۱۱۰۰	۱۰۱۰۰۰۰
۱	۰۱۰۰۰۱۰	۰۱۰۰۰۰۰۰۰۰	۱۰۰۰۱	۱۰۰۱۰۰۰
۲	۰۱۰۰۱۰۰	۰۰۱۰۰۰۰۰۰۰	۱۰۰۱۰	۱۰۰۰۱۰۰
۳	۰۱۰۱۰۰۰	۰۰۰۱۰۰۰۰۰۰	۰۰۰۱۱	۱۰۰۰۰۱۰
۴	۰۱۱۰۰۰۰	۰۰۰۰۱۰۰۰۰۰	۱۰۱۰۰	۱۰۰۰۰۰۱
۵	۱۰۰۰۰۰۱	۰۰۰۰۰۱۰۰۰۰	۰۰۱۰۱	۰۱۱۰۰۰۰
۶	۱۰۰۰۰۱۰	۰۰۰۰۰۰۱۰۰۰	۰۰۱۱۰	۰۱۰۱۰۰۰
۷	۱۰۰۰۱۰۰	۰۰۰۰۰۰۰۱۰۰	۱۱۰۰۰	۰۱۰۰۱۰۰
۸	۱۰۰۱۰۰۰	۰۰۰۰۰۰۰۰۱۰	۰۱۰۰۱	۰۱۰۰۰۱۰
۹	۱۰۱۰۰۰۰	۰۰۰۰۰۰۰۰۰۱	۰۱۰۱۰	۰۱۰۰۰۰۱

در کد دو پنجی (biquinary) که هفت بیتی است. دو بیت اول (از سمت چپ) نشان می‌دهد که عدد در بازه ۰ تا ۴ یا ۵ تا ۹ می‌باشد (۰۱ یعنی ۰ تا ۴ و ۱۰ یعنی ۵ تا ۹) و پنج بیت بعدی یکی از ۵ عدد موجود در آن بازه را نشان می‌دهد.

مهمترین مزیت کدهایی که از ۴ بیت بیشتر هستند، امکان تشخیص خطا است. کد دو پنجی ۷ بیتی است و ۷ بیت ۱۲۸ حالت دارد که از این تعداد فقط ۱۰ حالت آن مجاز است. پس هر خطایی که باعث می‌شود یکی از حالات غیرمجاز تولید شود، قابل تشخیص است.

در کد $\frac{۱}{۱۰}$ (1 out of 10)، فقط یک بیت از ده بیت برابر یک می‌باشد.

کد $\frac{۲}{۵}$، پنج بیتی است که دو بیت از پنج بیت برابرِ یک و سایر بیت‌ها، صفر هستند. این کد نیمه وزن‌دار است. به غیر از ۰ سایر ارقام از وزن پیروی می‌کنند. وزن بیت‌ها ۰۷۴۲۱ می‌باشد پس $(۰۱۰۱۰)_{\frac{۲}{۵}} = ۱\times ۷ + ۱\times ۲ = ۹$.

کد $\frac{۲}{۷}$ هفت بیتی است که دو بیت از هفت برابر یک و سایر بیت‌ها، صفر هستند. این کد وزن‌دار است و وزن بیت‌ها ۰۵۰۱۲۳۴ می‌باشد پس $(۰۱۰۰۰۰۱)_{\frac{۲}{۷}} = ۱\times ۵ + ۱\times ۴ = ۹$.

مثال ۳۰: اگر احتمال خطا در یک بیت برابر $۱۰^{-۴}$ باشد و عدد ۱۰۱۰۰ در کد $\frac{۲}{۵}$ دریافت شده باشد، چقدر احتمال دارد که این عدد خطا باشد؟ واضح است که این خطا در صورت وجود، قابل تشخیص نیست چون ۱۰۱۰۰ یک کلمهٔ کد معتبر است.

جواب مثال ۳۰: دو حالت امکان دارد که خطا پیش آمده و قابل تشخیص نباشد:

حالت ۱: یک بیت صفر به یک و یک بیت یک به صفر تبدیل شده باشد. احتمال این حالت برابر $۲\times ۱۰^{-۴} \times ۳\times ۱۰^{-۴} = ۶\times ۱۰^{-۸}$ می‌باشد.

حالت ۲: دو بیت صفر به یک و دو بیت یک به صفر تبدیل شده باشد که احتمال این حالت برابر $\binom{۳}{۲}\times ۱۰^{-۴}\times ۱۰^{-۴}\times \binom{۲}{۲}\times ۱۰^{-۴}\times ۱۰^{-۴} = ۳\times ۱۰^{-۱۶}$ است.

جواب، برابر مجموع این دو حالت یعنی $۶\times ۱۰^{-۸} + ۳\times ۱۰^{-۱۶}$ می‌باشد.

کد گِری (gray)

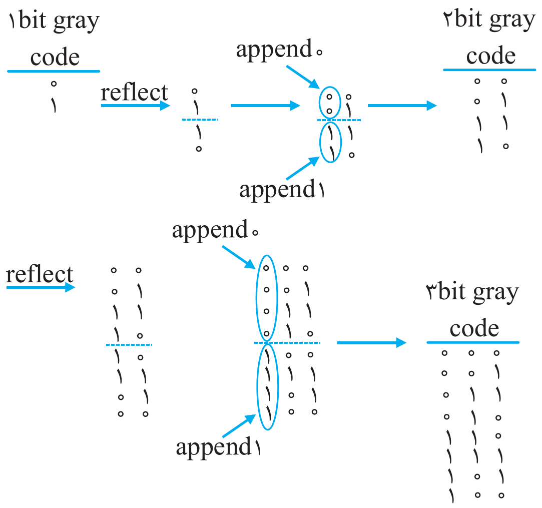
در این کد هر دو کلمهٔ کدِ متوالی، فقط در یک بیت با هم تفاوت دارند. به عبارتی فاصله (distance) بین هر دو کلمهٔ کد متوالی برابر ۱ است. در جدول زیر، کد باینری و کد گری سه بیتی آمده است. در کد باینری تعداد اختلاف بیت $(۰۰۱)_۲ = ۱$ و $(۰۱۰)_۲ = ۲$ دو بیت است: $\begin{array}{ccc} ۰ & ۰ & ۱ \\ & \updownarrow & \updownarrow \\ ۰ & ۱ & ۰ \end{array}$

ولی در کد گری تعداد اختلاف بیت بین هر دو کلمه کد متوالی، فقط یک بیت است.

Decimal number	Binary code	Gray code
۰	۰۰۰	۰۰۰
۱	۰۰۱	۰۰۱
۲	۰۱۰	۰۱۱
۳	۰۱۱	۰۱۰
۴	۱۰۰	۱۱۰
۵	۱۰۱	۱۱۱
۶	۱۱۰	۱۰۱
۷	۱۱۱	۱۰۰

برای نوشتن کلمه کدهای گری، می‌توان از این واقعیت کمک گرفت که این کد، اصطلاحاً کد انعکاسی (reflected code) است. به این معنی که برای یافتن کلمه کدهای $n+1$ بیتی، کافی است کلمه کدهای n بیتی گری را زیر هم بنویسیم و سپس زیر این کلمات، تصویر آینه‌ای همین کلمات را بنویسیم و سپس قبل از n کلمهٔ اول، صفر و قبل از n کلمه دوم، یک قرار دهیم.

به نحوه ساخت کد گری ۲ و ۳ بیتی دقت کنید:



ولی این شیوهٔ ساخت، زمان‌گیر است. زیرا برای ساخت کد گری $n+1$ بیتی باید ابتدا تا کد گری n بیتی را ساخته باشیم. می‌توان برای تبدیل کد باینری به گری، فرمول ارائه داد. قبل از ارائه فرمول، عملگری به نام XOR (exclusive OR : ⊕) را معرفی می‌کنیم. این عملگر، ۲ بیت دریافت می‌کند و اگر مخالف باشند، یک و در غیر این صورت، صفر تولید می‌کند:

ab	a ⊕ b
۰۰	۰
۰۱	۱
۱۰	۱
۱۱	۰

۰⊕۰=۰ و ۰⊕۱=۱ و ۱⊕۰=۱ و ۱⊕۱=۰

فرمول تبدیل کد باینری به کد گری به شکل زیر است:

$$(b_{n-۱}\ b_{n-۲}\ b_{n-۳}\ \cdots\ b_۱\ b_۰)_۲ = (g_{n-۱}\ g_{n-۲}\ g_{n-۳}\ \cdots\ g_۱\ g_۰)_{gray}$$

$$g_{n-۱} = b_{n-۱} \quad \text{و} \quad g_{n-۲} = b_{n-۱} \oplus b_{n-۲} \quad \text{و} \quad g_{n-۳} = b_{n-۲} \oplus b_{n-۳}, \ldots, g_۰ = b_۱ \oplus b_۰$$

یعنی بیت سمت چپ، عوض نمی‌شود و برای یافتن هر بیت مثل g_k در کد گری، کافی است بیت b_k را با $b_{k+۱}$ در باینری، XOR کنیم. می‌توان فرمول کلی تبدیل باینری به گری را به شکل $g_k = b_k \oplus b_{k+۱}$، $۰ \le k \le n-۱$ نوشت و $b_n = ۰$ تعریف کرد.

مثال ۳۱: عدد ۷۰ را با حداقل تعداد بیت در کد گری نمایش دهید.

جواب مثال ۳۱: ابتدا ۷۰ را به کد باینری تبدیل می‌کنیم و سپس با فرمول گفته شده، به کد گری تبدیل می‌کنیم:

$$۷۰ = ۶۴+۴+۲ = (۱\ ۰\ ۰\ ۰\ ۱\ ۱\ ۰)_۲ = (۱\ ۱\ ۰\ ۰\ ۱\ ۰\ ۱)_{gray}$$

برای تبدیل از کد گری به باینری می‌توان فرمول زیر را ارائه داد:

$$(g_{n-۱}\ g_{n-۲}\ g_{n-۳}\ \cdots\ g_۱\ g_۰)_{gray} \rightarrow (b_{n-۱}\ b_{n-۲}\ b_{n-۳}\ \cdots\ b_۱\ b_۰)_۲$$

$$b_{n-۱} = g_{n-۱} \quad \text{و} \quad b_{n-۲} = b_{n-۱} \oplus g_{n-۲} \quad \text{و} \quad b_{n-۳} = b_{n-۲} \oplus g_{n-۳}, \ldots, b_۰ = b_۱ \oplus g_۰$$

با جایگذاری هر فرمول در فرمول بعدی، می‌توان به فرمول‌های زیر رسید:

$$\begin{aligned}
b_{n-۱} &= g_{n-۱} \\
b_{n-۲} &= b_{n-۱} \oplus g_{n-۲} = g_{n-۱} \oplus g_{n-۲} \\
b_{n-۳} &= b_{n-۲} \oplus g_{n-۳} = g_{n-۱} \oplus g_{n-۲} \oplus g_{n-۳} \\
&\vdots \\
b_۰ &= b_۱ \oplus g_۰ = g_{n-۱} \oplus g_{n-۲} \oplus \ldots \oplus g_۰
\end{aligned}$$

لازم به ذکر است، عمل XOR $(\oplus)$ یک تابع فرد (odd function) است یعنی اگر تعداد یک‌ها فرد باشد، حاصل این عمل برابر یک است و در غیر این صورت، حاصل برابر صفر است مثلاً: $۱ \oplus ۱ \oplus ۱ \oplus ۰ \oplus ۱ = ۰$ و $۱ \oplus ۰ \oplus ۱ \oplus ۱ = ۱$. می‌توان گفت عمل XOR همه بیت‌ها را با هم جمع می‌کند و سپس باقیماندهٔ نتیجه را به ۲ محاسبه می‌کند. مثلاً

$$۱ \oplus ۰ \oplus ۱ \oplus ۱ = (۱+۰+۱+۱) \bmod ۲ = ۳ \bmod ۲ = ۱$$

و

$$۱\oplus۱\oplus۱\oplus۰\oplus۱=(۱+۱+۱+۰+۱)\bmod ۲=۴\bmod ۲=۰$$

پس می‌توان گفت

$$b_۰=g_{n-۱}\oplus g_{n-۲}\oplus\ldots\oplus g_۰=(g_{n-۱}+g_{n-۲}+\ldots+g_۰)\bmod ۲$$

مثال ۳۲: عدد ۸ بیتی $(۱۰۰۱۱۰۱۱)_{gray}$ در کد گری نوشته شده است. این عدد را به دهدهی تبدیل کنید.

جواب مثال ۳۲: ابتدا به باینری تبدیل می‌کنیم و سپس باینری را به دهدهی تبدیل می‌کنیم:

$$(۱۰۰۱۱۰۱۱)_{gray}=(b_۷b_۶b_۵b_۴b_۳b_۲b_۱b_۰)_۲$$

$$b_۷=۱ \quad و \quad b_۶=۱\oplus۰=۱ \quad و \quad b_۵=۱\oplus۰\oplus۰=۱ \quad و \quad b_۴=۱\oplus۰\oplus۰\oplus۱=۰$$

$$b_۳=۱\oplus۰\oplus۰\oplus۱\oplus۱=۱ \quad و \quad b_۲=۱\oplus۰\oplus۰\oplus۱\oplus۱\oplus۰=۱$$

$$b_۱=۱\oplus۰\oplus۰\oplus۱\oplus۱\oplus۰\oplus۱=۰ \quad و \quad b_۰=۱\oplus۰\oplus۰\oplus۱\oplus۱\oplus۰\oplus۱\oplus۱=۱$$

پس حاصل به شکل باینری $(۱۱۱۰۱۱۰۱)_۲$ می‌باشد که به دهدهی تبدیل می‌کنیم:

$$(۱۱۱۰۱۱۰۱)_۲=۱۲۸+۶۴+۳۲+۸+۴+۱=۲۳۷$$

کدهای تشخیص و تصحیح خطا

شرط آنکه در یک کد، امکان تشخیص خطا وجود داشته باشد، آن است که در کد مورد نظر، هم کلمات معتبر و هم کلمات نامعتبر وجود داشته باشند. مثلاً در کد BCD، کلمهٔ ۱۰۰۰ معتبر است ولی ۱۱۰۰ معتبر نیست. پس اگر خطا باعث شود کلمهٔ ۱۰۰۰ به ۱۱۰۰ تبدیل شود، آن خطا قابل تشخیص است. البته در کد BCD همهٔ خطاها قابل تشخیص نیستند مثلاً اگر خطا باعث شود، کلمهٔ ۱۰۰۰ به ۱۰۰۱ تبدیل شود، این خطا قابل تشخیص نیست چون ۱۰۰۱ در کد BCD، یک کلمهٔ معتبر است.

فاصله (distance) بین دو کلمهٔ کد، برابر تعداد اختلاف بیت‌های آن دو کلمه است. مثلاً اگر $I=۱۱۰۱۰$ و $J=۱۰۰۰۱$ دو کلمهٔ کد باشند آنگاه فاصله این دو برابر $d(I,J)=۳$ است زیرا I و J در ۳ بیت با هم تفاوت دارند:

$$\begin{array}{ccccc} ۱ & ۱ & ۰ & ۱ & ۰ \\ & \updownarrow & & \updownarrow & \updownarrow \\ ۱ & ۰ & ۰ & ۰ & ۱ \end{array}$$

در یک کد، حداقلِ فاصله، برابر با حداقل فاصلهٔ بین کلمات آن کد است. تمام کدهایی که تاکنون معرفی شده‌اند، حداقل فاصله‌شان برابر یک است چون حداقل ۲ کلمه وجود دارد که فاصلهٔ آنها برابر یک باشد. در چنین کدهایی اگر یک بیت خطا پیش آید، ممکن است باعث شود یک کلمهٔ معتبر به یک کلمهٔ معتبر دیگر تبدیل شود و خطا قابل تشخیص نباشد. مثلاً در کد BCD اگر در کلمهٔ کد ۰۰۰۱ یک بیت خطا شود، کلمه‌ای که به وجود می‌آید یک کلمهٔ دیگر در BCD است و خطا قابل تشخیص نیست.

شرط تشخیص خطا این است که کدی بسازیم با حداقل فاصلهٔ ۲. برای ساختن چنین کدی، کافی است یک بیت به اسم توازن (parity) به کلمات اضافه کنیم. دو نوع توازن وجود دارد: زوج و فرد. توازن زوج (even) بیتی است که به هر کلمهٔ کد اضافه می‌شود و باعث می‌شود کلمات n بیتی تبدیل به کلمات $n+1$ بیتی تبدیل شوند و تعدادِ یک در هر کلمه، زوج شود. توازنِ فرد (odd) نیز باعث می‌شود، تعدادِ یک در هر کلمه، فرد شود.

جدول مقابل به کلمه کدهای سه بیتی باینری، توازن زوج (p_e) اضافه کرده است. با توجه به جدول، اکنون داده‌های ما ۴ بیتی هستند. فرض کنید بخواهیم کلمهٔ $xyzp_e = ۱۰۰۱$ را برای گیرنده‌ای ارسال کنیم و روی بیت دوم (از سمت چپ) خطا پیش آید و گیرنده ۱۱۰۱ دریافت کند. گیرنده تشخیص می‌دهد که خطا رخ داده است زیرا کلمهٔ ۱۱۰۱ معتبر نیست (زیرا در تمامی کلمات به شکل $xyzp_e$، تعداد یک‌ها زوج هستند). واضح است که اگر روی ۲ بیت خطا پیش آید قابل تشخیص نیست. مثلاً اگر ۱۱۱۱ با خطا تبدیل به ۰۰۱۱ (۲ بیت سمت چپ خطا شده) شود، قابل تشخیص نیست.

x	y	z	p_e
۰	۰	۰	۰
۰	۰	۱	۱
۰	۱	۰	۱
۰	۱	۱	۰
۱	۰	۰	۱
۱	۰	۱	۰
۱	۱	۰	۰
۱	۱	۱	۱

برای محاسبهٔ p_e و افزودن آنها به داده‌های اولیه، کافی است همه بیت‌های اولیه را با هم XOR کنیم، یعنی $p_e = x \oplus y \oplus z$. گیرنده برای تشخیص خطا می‌تواند هر ۴ بیت را با هم XOR کند یعنی $C = x \oplus y \oplus z \oplus p_e$ اگر $C = ۱$ یعنی خطا پیش آمده است (حاصل XOR وقتی یک می‌شود که تعداد یک‌ها فرد باشد که به معنی خطاست).

نتیجه: با داشتن کدی که حداقل فاصله‌اش برابر ۲ است می‌توان یک بیت خطا را تشخیص داد. به طور کلی با داشتن کدی که حداقل فاصله‌اش برابر d است، می‌توان تا حداکثر $d-۱$ خطا را تشخیص داد زیرا اگر $d-۱$ بیت از یک کلمهٔ معتبر عوض شوند، این کلمه تبدیل به یک کلمهٔ معتبر دیگر نمی‌شود زیرا فاصله بین کلمات حداقل d است.

برای تصحیح خطا باید فاصلهٔ کد، حداقل برابر ۳ باشد. اگر فاصلهٔ کد (حداقل فاصلهٔ بین کلمات کد) برابر ۳ باشد آنگاه یک بیت خطا، باعث می‌شود که یک کلمهٔ معتبر به یک کلمهٔ نامعتبر تبدیل شود که فاصله این کلمهٔ نامعتبر با کلمهٔ اصلی‌اش برابر یک و با سایر کلمات کد حداقل ۲ است، پس می‌توان آن را تصحیح کرد یعنی کلمهٔ نامعتبر را به کلمه‌ای که نزدیک‌تر است (فاصله‌اش کمتر است) تبدیل کرد.

مثلاً فرض کنید در یک کد با فاصلهٔ ۳، دو کلمهٔ معتبرِ $I = ۱۱۰۱۰$ و $J = ۱۰۰۰۱$ وجود دارند. حال اگر بیت دوم کلمه I خطا شود، کلمه نامعتبر $I' = ۱۰۰۱۰$ بوجود می‌آید که فاصله I' تا I برابر یک و فاصله I' تا J برابر ۲ است پس I' باید به I تبدیل شود که این یعنی تصحیح خطا.

نتیجه: با داشتن کدی که حداقل فاصله آن d است می‌توان حداکثر $\left\lfloor \frac{d-۱}{۲} \right\rfloor$ خطا را تصحیح کرد.

نتیجه: اگر حداقل فاصلهٔ کد برابر $۲c+d+۱$ باشد آنگاه می‌توان تا c خطا را تصحیح کـرد و تـا d خطـای اضـافی را تشـخیص داد. مـثلاً کـدی کـه حـداقل فاصـله ۵ دارد $(۲c+d+۱=۵)$ دارای قابلیت‌های مقابل است:

تشخیص خطا در ۴ بیت $\rightarrow d=۴$ و $c=۰$

تصحیح یک بیت خطا و تشخیص ۲ خطای اضافی $\rightarrow d=۲$ و $c=۱$

تصحیح دو بیت خطا $\rightarrow d=۰$ و $c=۲$

کد همینگ (Hamming): در سـال ۱۹۵۰ R.W.Hamming یـک روش کلـی بـرای سـاخت کدهایی با حداقل فاصلهٔ ۳ ارائه کرد که به کد همینگ معروف شد، که توضیح می‌دهیم.

برای آنکه بتوان حداقل فاصلهٔ یک کد را برابر ۳ کرد و در نتیجه یک خطا را تصحیح کـرد، بایـد به کلماتِ کد، بیت اضافه شود. اگر داده‌های اولیه ۴ بیتی باشند باید به آنها ۳ بیـت اضـافه کنـیم و اگر داده‌های اولیه ۸ بیتی باشند، به آنها ۴ بیت اضافه می‌کنیم (علت را خواهیم دید).

ما برای داده‌های ۴ بیتی کد همینگ می‌سـازیم. پـس بـرای داده‌هـای ۴ بیتـی، ۳ بیـت اضـافه می‌کنیم و در نهایت ۷ بیت را برای گیرنده ارسال مـی‌کنـیم (فـرض کنیـد مـا فرسـتنده هسـتیم).

بیت‌ها را از سمت چپ و شروع از یک شماره‌گـذاری مـی‌کنـیم

۱	۲	۳	۴	۵	۶	۷

سـه بیـتِ اضافه شده را در مکان‌هایی که شماره‌شان توانی از ۲ است قرار می‌دهیم یعنی در مکان‌هـای ۱ و ۲ و ۴ (سه بیت اضافه شده را $c_۱$ و $c_۲$ و $c_۴$) می‌نامیم. ۴ بیت اولیه را در مکـان‌هـای ۳ و ۵ و ۶ و ۷ قرار می‌دهیم ($b_۳b_۵b_۶b_۷$ دادهٔ ۴ بیتی اولیه است):

۱	۲	۳	۴	۵	۶	۷
$c_۱$	$c_۲$	$b_۳$	$c_۴$	$b_۵$	$b_۶$	$b_۷$

برای یافتن $c_۱$ و $c_۲$ و $c_۴$، تعدادی از بیت‌های دادهٔ اصلی را با هم XOR می‌کنیم:

$$\begin{cases} c_۱ = b_۳ \oplus b_۵ \oplus b_۷ = XOR(۳,۵,۷) \\ c_۲ = b_۳ \oplus b_۶ \oplus b_۷ = XOR(۳,۶,۷) \\ c_۴ = b_۵ \oplus b_۶ \oplus b_۷ = XOR(۵,۶,۷) \end{cases}$$

(برای درک این فرمول‌ها، شمارهٔ بیت‌های دادهٔ اولیه یعنی ۳ و ۵ و ۶ و ۷ را به صـورت مجمـوع اعداد ۱ و ۲ و ۴ بنویسید: $۳=۲+۱$ و $۵=۴+۱$ و $۶=۴+۲$ و $۷=۴+۲+۱$، حال شماره‌هـایی که در آنها ۱ وجود دارد یعنی ۳ و ۵ و ۷ سازندهٔ $c_۱$ هستند. شماره‌هایی کـه در آنهـا ۲ وجـود دارد یعنی ۳ و ۶ و ۷ سازندهٔ $c_۲$ هستند و شماره‌هایی که در آنها ۴ وجود دارد یعنی ۵ و ۶ و ۷ سـازندهٔ $c_۴$ هستند)

مثال ۳۳: به دادهٔ ۴ بیتی ۰۰۱۱، سه بیت همینگ را اضافه کنید و ۷ بیت برای گیرنده ارسال کنید:

جواب مثال ۳۳: $b_3b_5b_6b_7 = 0011$ یعنی $\begin{array}{|c|c|c|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} & \multicolumn{1}{c}{7} \\ \hline c_1 & c_2 & 0 & c_4 & 0 & 1 & 1 \\ \hline \end{array}$. با توجه به فرمول‌های c_1 و c_2 و c_4 داریم:

$$c_1 = XOR(3,5,7) = XOR(0,0,1) = 1$$
$$c_2 = XOR(3,6,7) = XOR(0,1,1) = 0$$
$$c_4 = XOR(5,6,7) = XOR(0,1,1) = 0$$

پس ۷ بیت ارسالی برای گیرنده عبارت است از: ۱۰۰۰۰۱۱

حال بررسی می‌کنیم گیرنده پس از دریافت ۷ بیت چگونه خطا را در صورت وجود تصحیح می‌کند. گیرنده پس از دریافت ۷ بیت با کمک فرمول‌های زیر، محل خطا را در صورت وجود، تشخیص می‌دهد و آن را تصحیح می‌کند:

$$\begin{cases} d_1 = c_1 \oplus b_3 \oplus b_5 \oplus b_7 = XOR(1,3,5,7) \\ d_2 = c_2 \oplus b_3 \oplus b_6 \oplus b_7 = XOR(2,3,6,7) \\ d_4 = c_4 \oplus b_5 \oplus b_6 \oplus b_7 = XOR(4,5,6,7) \end{cases}$$

حال $d_4d_2d_1$ یک عدد ۳ بیتی است که اگر ۰۰۰ باشد یعنی عدم وجود خطا. و در غیر این صورت، نشان می‌دهد کدام بیت خطا است:

d_4 d_2 d_1		
۰ ۰ ۰	→	عدم وجود خطا
۰ ۰ ۱	→	بیت شمارهٔ ۱ (c_1) خطاست
۰ ۱ ۰	→	بیت شمارهٔ ۲ (c_2) خطاست
۰ ۱ ۱	→	بیت شمارهٔ ۳ (b_3) خطاست
۱ ۰ ۰	→	بیت شمارهٔ ۴ (c_4) خطاست
۱ ۰ ۱	→	بیت شمارهٔ ۵ (b_5) خطاست
۱ ۱ ۰	→	بیت شمارهٔ ۶ (b_6) خطاست
۱ ۱ ۱	→	بیت شمارهٔ ۷ (b_7) خطاست

مثال ۳۴: گیرنده عدد ۷ بیتی ۱۰۱۰۰۱۱ را دریافت کرده است. در صورت وجود خطا آن را تصحیح کنید.

جواب مثال ۳۴: گیرنده $\begin{array}{|c|c|c|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} & \multicolumn{1}{c}{7} \\ \hline 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ \hline \end{array}$ را دریافت کرده است. d_1 و d_2 و d_4 را محاسبه می‌کنیم:

$$d_۱ = XOR(۱,۳,۵,۷) = XOR(۱,۱,۰,۱) = ۱$$
$$d_۲ = XOR(۲,۳,۶,۷) = XOR(۰,۱,۱,۱) = ۱$$
$$d_۴ = XOR(۴,۵,۶,۷) = XOR(۰,۰,۱,۱) = ۰$$

پس $d_۴d_۲d_۱ = ۰۱۱$ یعنی بیت شماره ۳ خطاست یعنی ۷ بیتی که توسط فرستنده ارسال شده است ۱۰۰۰۰۱۱ می‌باشد. در واقع این مثال ادامهٔ مثال قبلی است که فرستنده ۱۰۰۰۰۱۱ را ارسال کرد و خطایی پیش آمد و گیرنده متوجه شده و خطا را تصحیح کرد. ■

حال می‌خواهیم برای داده‌های ۸ بیتی، کد همینگ بسازیم. همانطور که گفته شد باید ۴ بیت اضافه کنیم. یعنی باید ۱۲ بیت برای گیرنده ارسال کنیم. بیت‌های اضافه شده $(c_۸, c_۴, c_۲, c_۱)$ در مکان‌هایی قرار می‌گیرند که شماره‌شان توانی از ۲ است:

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
$c_۱$	$c_۲$		$c_۴$				$c_۸$				

با توجه به اینکه ۱+۲=۳، ۱+۴=۵، ۲+۴=۶، ۱+۲+۴=۷، ۱+۸=۹، ۲+۸=۱۰، ۱+۲+۸=۱۱ و ۴+۸=۱۲ پس:

$$\begin{cases} c_۱ = XOR(۳,۵,۷,۹,۱۱) \\ c_۲ = XOR(۳,۶,۷,۱۰,۱۱) \\ c_۴ = XOR(۵,۶,۷,۱۲) \\ c_۸ = XOR(۹,۱۰,۱۱,۱۲) \end{cases}$$

پس گیرنده با دریافت ۱۲ بیت و با کمک فرمول‌های زیر، محل خطا را در صورت وجود، تشخیص می‌دهد و آن را تصحیح می‌کند:

$$\begin{cases} d_۱ = XOR(۱,۳,۵,۷,۹,۱۱) \\ d_۲ = XOR(۲,۳,۶,۷,۱۰,۱۱) \\ d_۴ = XOR(۴,۵,۶,۷,۱۲) \\ d_۸ = XOR(۸,۹,۱۰,۱۱,۱۲) \end{cases}$$

اگر $d_۸d_۴d_۲d_۱ = ۰۰۰۰$ یعنی عدم وجود خطا، در غیر این صورت اگر $d_۸d_۴d_۲d_۱ = ۰۰۰۱..۱۱۰۰$ یعنی بیت شمارهٔ ۱ تا ۱۲ خطاست. و اگر $d_۸d_۴d_۲d_۱ = ۱۱۰۱..۱۱۱۱$ یعنی خطا در بیش از یک بیت رخ داده است و قابل تصحیح نیست (البته این لازم و کافی نیست. یعنی ممکن است خطا در بیش از یک بیت رخ دهد و $d_۸d_۴d_۲d_۱$ برابر ۱۱۰۱ یا ۱۱۱۰ یا ۱۱۱۱ نباشد)

حال می‌خواهیم بررسی کنیم، چرا به داده‌های ۴ بیتی ۳ بیت اضافه کردیم و به داده‌های ۸ بیتی، ۴ بیت اضافه کردیم. فرض کنید داده‌ها n بیتی هستند و k بیت به آنها اضافه می‌کنیم و $n+k$ بیت می‌سازیم و به گیرنده ارسال می‌کنیم. k بیتی که اضافه می‌کنیم برای گیرنده ۲^k

حالت ایجاد می‌کند که یک حالت آن (همهٔ d_iها صفر باشند) نشان دهندهٔ عدم وجود خطاست و $۲^k - ۱$ حالت دیگر باید بتواند نشان دهد کدام یک از $n+k$ بیت خطاست. پس باید $\boxed{۲^k - ۱ \ge n+k}$.

مثال ۳۵: به داده‌های ۲۰ بیتی حداقل چند بیت اضافه کنیم تا بتوان یک خطا را تصحیح کرد؟

جواب مثال ۳۵: باید از نامساوی $۲^k - ۱ \ge ۲۰ + k$، حداقل مقدار k را بیابیم که با بررسی جواب $k = ۵\text{bit}$ است.

مثال ۳۶: به داده‌های n بیتی، ۵ بیت اضافه کرده‌ایم تا بتوان یک خطا را تصحیح کرد، بهترین مقادیر برای n را بیابید.

جواب مثال ۳۶: با توجه به رابطهٔ $۲^k - ۱ \ge n + k$، اگر $k = ۵$ قرار دهیم، حداکثر مقدار n حاصل می‌شود: $۲^۵ - ۱ \ge n + ۵ \rightarrow n \le ۲۶$. برای یافتن حداقل مقدار برای n، به جای k مقدار ۴ قرار می‌دهیم: $۲^۴ - ۱ \ge n + ۴ \rightarrow n \le ۱۱$ یعنی به داده‌های حداکثر ۱۱ بیتی باید ۴ بیت اضافه کنیم. پس به داده‌های حداقل ۱۲ بیتی باید ۵ بیت اضافه کنیم. بنابراین بهترین مقادیر برای n که $k = ۵$ بیت به آنها اضافه کنیم $۱۲ \le n \le ۲۶$ می‌باشد.

نتیجه: برای اعداد n بیتی باید k بیت توازن اضافه شود که حداقل فاصلهٔ کد برابر ۳ شود و در جدول زیر مقادیر مناسب n به ازای kهای مختلف آمده است:

Information Bist(n)	Parity Bits(k)
$n = ۱$	$k = ۲$
$۲ \le n \le ۴$	$k = ۳$
$۵ \le n \le ۱۱$	$k = ۴$
$۱۲ \le n \le ۲۶$	$k = ۵$
$۲۷ \le n \le ۵۷$	$k = ۶$
$۵۸ \le n \le ۱۲۰$	$k = ۷$

نکته

به راحتی می‌توان کد همینگ با حداقل فاصلهٔ ۳ را به کدی با حداقل فاصلهٔ ۴ تبدیل کرد. کافی است به کد همینگ با فاصلهٔ ۳، یک بیت توازن زوج اضافه کنیم. با افزودن این بیتِ توازن، فاصلهٔ کد برابر ۴ می‌شود و می‌توان حداکثر ۳ خطا را تشخیص داد و همچنین قابلیت‌های دیگری نیز در کد ایجاد می‌شود. فرض کنید ما به اعداد ۸ بیتی، ۴ بیت اضافه کرده‌ایم و کدی ۱۲ بیتی با فاصله ۳ ساخته‌ایم. حال به این ۱۲ بیت، یک بیت توازن زوج به اسم $p_{۱۳}$ اضافه می‌کنیم یعنی $p_{۱۳} = XOR(۱,۲,..,۱۲)$ و ۱۳ بیت برای گیرنده