مدارهای منطقی

سری کتاب‌های کمک آموزشی کارشناسی ارشد

مجموعه مهندسی کامپیوتر

مؤلف: گروه مولفان

ویراستار علمی:

سمانه سرورنژاد

سرشناسه	: گروه مولفان
عنوان	: مدارهای منطقی
مشخصات نشر	: تهران : مشاوران صعود ماهان، ۱۴۰۱
مشخصات ظاهری	: ۱۶۹ ص
فروست	: سری کتاب‌های کمک آموزشی کارشناسی ارشد
شابک	: ۹۷۸-۶۰۰-۴۵۸-۸۰۹-۶
وضعیت فهرست نویسی	: فیپای مختصر
یادداشت	:این مدرک در آدرس http://opac.nlai.ir قابل دسترسی است.
شناسه افزوده	: سمانه سرورنژاد، ۱۳۶۲ ، ویراستار علمی
شماره کتابشناسی ملی	: ۳۷۷۱۸۸۴

نام کتاب:... مدارهای منطقی

مولف:... گروه مولفان

مدیر تولید محتوی .... سمیه بیگی

ویراستار علمی ....سمانه سرورنژاد

ناشر:... مشاوران صعود ماهان

نوبت و تاریخ چاپ: ....اول / ۱۴۰۱

تیراژ: .... ۱۰۰۰ نسخه

قیمت:... ۲/۰۹۰/۰۰۰

شابک: .... ISBN: ۹۷۸-۶۰۰-۴۵۸-۸۰۹-۶

انتشـــارات مشاوران صعود ماهان: خیابان ولیعصر، بالاتر از تقـاطع مطهری،

روبروی قنادی هتل بزرگ تهـــران، جنب بانک ملی، پلاک ۲۰۵۰

تلفن: ۴- ۸۸۱۰۰۱۱۳

سخن ناشر

«ن والقلم و ما یسطرون»

کلمه نزد خدا بود و خدا آن را با قلم بر ما نازل کرد.

به پاس تشکر از چنین موهبت الهی، موسسه ماهان درصدد برآمده است تا در راستای انتقال دانش و مفاهیم با کمک اساتید مجرب و مجموعه کتب آموزشی خود برای شما داوطلبان ادامه تحصیل در مقطع کارشناسی ارشد گام موثری بردارد. امید است تلاش‌های خدمتگزاران شما در این موسسه پایه‌گذار گام‌های بلند فردای شما باشد.

مجموعه کتاب‌های کمک آموزشی ماهان به‌منظور استفاده داوطلبان کنکور کارشناسی ارشد سراسری و آزاد تالیف شده‌اند. در این کتاب‌ها سعی کرده‌ایم با بهره‌گیری از تجربه اساتید بزرگ و کتب معتبر داوطلبان را از مطالعه کتاب‌های متعدد در هر درس بی‌نیاز کنیم.

دیگر تالیفات ماهان برای سایر دانشجویان به‌صورت ذیل می‌باشد.

- **مجموعه کتاب‌های ۸ آزمون:** شامل ۵ مرحله کنکور کارشناسی ارشد ۵ سال اخیر به همراه ۳ مرحله آزمون تالیفی ماهان همراه با پاسخ تشریحی می‌باشد که برای آشنایی با نمونه سوالات کنکور طراحی شده است. این مجموعه کتاب‌ها با توجه به تحلیل ۳ ساله اخیر کنکور و بودجه‌بندی مباحث در هریک از دروس، اطلاعات مناسبی جهت برنامه‌ریزی درسی در اختیار دانشجو قرار می‌دهد.
- **مجموعه کتاب‌های کوچک:** شامل کلیه نکات کاربردی در گرایش‌های مختلف کنکور کارشناسی ارشد می‌باشد که برای دانشجویان جهت جمع‌بندی مباحث در ۲ ماهه آخر قبل از کنکور مفید می‌باشد.

بدین‌وسیله از مجموعه اساتید، مولفان و همکاران محترم خانواده بزرگ ماهان که در تولید و به‌روزرسانی تالیفات ماهان نقش موثری داشته‌اند، صمیمانه تقدیر و تشکر می‌نماییم.

دانشجویان عزیز و اساتید محترم می‌توانند هرگونه انتقاد و پیشنهاد درخصوص تالیفات ماهان را از طریق سایت ماهان به آدرس mahan.ac.ir با ما در میان بگذارند.

موسسه آموزش عالی آزاد ماهان

سخن مولف

کتاب حاضر، پس از چندین بار ویرایش، برای ارایه به دوستان متقاضی کنکور کارشناسی ارشد رشته‌های حوزه کامپیوتر آماده گردیده است. این کتاب در ۵ فصل تنظیم و خدمت دانشجویان گرامی ارائه شده است. در برخی فصول از جمله فصل ۲، ۳و ۵ مطالب مفید و مهمی ارایه شده که توصیه می‌شود با دقت و درایت بیشتری بررسی و خوانده شود. در فصل یک نیز مطالب پایه‌ای گنجانده شده که هم برای این درس و هم برای درس زبان ماشین و اسمبلی می‌تواند مفید واقع شود. در پایان هر فصل نیز سعی شده سوالات مربوط به سر فصل‌های همان فصل گنجانده شده که می‌تواند برای کمک به یادگیری مطالب همان فصل مفید باشد. امیدوارم این کتاب بتواند شما را به اهدافتان نزدیک‌تر کند.

از کلیه عزیزانی که در گردآوری این کتاب زحمت کشیده‌اند سپاسگزارم.

فهرست

عنوان	صفحه

فصل اول

سیستم اعداد و کدگذاری

- سیستم اعداد
- تبدیل مبنای اعداد
- بررسی سرریز
- شرایط سرریز
- نمایش افزونه یا بایاس شده
- روش‌های کدگذاری
- کدهای تصحیح و تشخیص خطا
- کد توازن فرد/ زوج
- کد همینگ

سیستم اعداد و کدگذاری

هدف از این درس بررسی مدارهای منطقی و چگونگی طراحی آن‌ها می‌باشد. طراحی یک مدار منطقی باید بهینه باشد به‌طوری‌که بتواند رفتار از پیش تعیین شده‌ای را از خود نشان دهد و در عین حال دارای سرعت بالا، هزینه کم و انرژی مصرفی پایینی باشد. در این درس منظور از منطق، منطق ۰ و ۱ می‌باشد که توابع خاص خود را دارد و مدارهای منطقی این توابع را پیاده‌سازی می‌کنند. از آنجایی که منطق ۰ و ۱ بر پایه جبر بول استوار است، بنابراین باید با سیستم جبربول آشنا شوید ولی قبل از آن شما را با نمایش اعداد در قراردادهای مختلف و کدگذاری‌های مختلف آشنا می‌کنیم.

سیستم اعداد

به‌طور کلی در مبناهای مختلف اعداد را به دو صورت زیر نمایش می‌دهند:

الف ـ نمایش مکانی (Positional Notation)

این نمایش همان نمایشی است که در محاسبات روزمره از آن استفاده می‌کنیم. با این فرض که عدد N عددی در مبنای r باشد، نمایش مکانی آن را به‌صورت زیر نمایش می‌دهیم:

$$N = (a_{n-۱}a_{n-۲}....a_۱a_۰ .a_{-۱}a_{-۲}.....a_{-m})_r$$

این عدد دارای n رقم صحیح و m رقم اعشاری می‌باشد. رقم با ارزش (MSD: Most Significant Digital) آن برابر $a_{n-۱}$ و رقم کم ارزش (LSD: Least Significant Digit) آن برابر a_{-m} می‌باشد.

به‌طورکلی اگر r یک مبنای عددی باشد، داریم:

$$۰ \le \text{محدوده یک رقم در مبنای } r \le (r-۱)$$

بنابراین در مبنای ۱۰ داریم: ۹≥ محدود یک رقم در مبنای ده ≥ ۰

تذکر: از آن‌جایی‌که در قراردادهای روزمره همواره مبنای اعداد $r = ۱۰$ می‌باشد بنابراین از ذکر مبنای ده صرف‌نظر می‌کنیم.

ب ـ نمایش چندجمله‌ای (Polynomial Notation)

به این نوع نمایش، نمایش سری‌گونه (Series Represention) نیز می‌گوییم. عدد N که در بالا آمد، در این نمایش به‌صورت زیر می‌باشد:

$$N = a_{n-۱}r^{n-۱} + a_{n-۲}r^{n-۲} + ... + a_۰r^۰ + a_{-۱}r^{-۱} + ... + a_{-m}r^{-m} = \sum_{i=-m}^{n-۱} a_i r^i$$

مبنای ۲ (Binary Base)

ارقام مجاز در مبنای دو $\{0,1\}$ می‌باشند.

نکته: اعمال ریاضی در مبنای r از همان قواعدی که برای اعداد دهدهی حاکم است، پیروی می‌کنند.

مثال: عدد زیر یک عدد دودویی است که نمایش‌های مختلف آن آمده است.

$$(\underset{4}{1}\underset{3}{1}\underset{2}{0}\underset{1}{1}\underset{0}{0}.\underset{-1}{1}\underset{-2}{1}) = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 + 1\times2^{-1} + 1\times2^{-2} = (26.75)_{10}$$

$$(\underset{5}{1}\ \underset{4}{0}\ \underset{3}{1}\ \underset{2}{1}\ \underset{1}{0}\ \underset{0}{1})_2 = 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 = 45$$

مبنای ۸ (Octal Base)

در اینجا چون $r=8$ است، داریم:

$$0 \le \text{محدوده یک رقم در مبنای هشت} \le 7$$

بنابراین ارقام مجاز از صفر تا ۷ خواهد بود.

عدد در مبنای هشت یا هشتی = octal = oct = o

به‌عنوان مثال داریم:

$$(127.4)_8 = 1\times8^2 + 2\times8^1 + 7\times8^0 + 4\times8^{-1} = (87.5)_{10}$$
$$(736)_8 = (736)_{octal} = (736)_{oct} = (736)_o$$

مبنای ۱۶ (Hexadecimal Base)

در اینجا چون $r = 16$ است، داریم:

$$0 \le \text{محدوده یک رقم در مبنای شانزده} \le 15$$

ارقام این اعداد عبارتند از: $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$

به‌عنوان مثال داریم:

$$(A2B)_{16} = (A2B)_{Hexadecimal} = (A2B)_{Hex} = (A2B)_H$$
$$(B65F)_{16} = 11\times16^3 + 6\times16^2 + 5\times16^1 + 15\times16^0 = (46687)_{10}$$

نکته: در سیستم نمایش دودویی واحد زیر متداول است:

$1K(kilo) = 2^{10} = 1,024$ بایت

$1M(Mega) = 2^{20} = 1,048,576$ بایت

$1G(Giga) = 2^{30} = 1,073,741,824$ بایت

در جدول زیر اعداد در مبناهای مختلف آورده شده است:

دهدهی (مبنای ۱۰)	دودویی (مبنای ۲)	هشت‌تایی (مبنای ۸)	شانزده‌تایی (مبنای ۱۶)
۰۰	۰۰۰۰	۰۰	۰
۰۱	۰۰۰۱	۰۱	۱
۰۲	۰۰۱۰	۰۲	۲
۰۳	۰۰۱۱	۰۳	۳
۰۴	۰۱۰۰	۰۴	۴
۰۵	۰۱۰۱	۰۵	۵
۰۶	۰۱۱۰	۰۶	۶
۰۷	۰۱۱۱	۰۷	۷
۰۸	۱۰۰۰	۱۰	۸
۰۹	۱۰۰۱	۱۱	۹
۱۰	۱۰۱۰	۱۲	A
۱۱	۱۰۱۱	۱۳	B
۱۲	۱۱۰۰	۱۴	C
۱۳	۱۱۰۱	۱۵	D
۱۴	۱۱۱۰	۱۶	E
۱۵	۱۱۱۱	۱۷	F

تبدیل مبنای اعداد

تبدیل از هر مبنای دلخواه به مبنای ۱۰

برای این کار کافی است بسط عدد در مبنای داده شده را نوشته و سپس حاصل را محاسبه کنیم، این حاصل، معادل عدد در مبنای ۱۰ است. به‌عنوان مثال:

$$(۱۰۱۰.۰۱۱)_۲ = ۲^۳\times۱+۰\times۲^۲+۱\times۲^۱+۰\times۲^۰+۰\times۲^{-۱}+۱\times۲^{-۱}+۱\times۲^{-۲}=(۱۰/۳۷۵)_{۱۰}$$

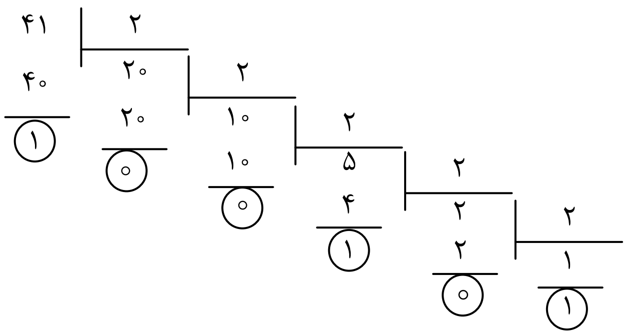
$$۶\times۸^۲+۳\times۸^۱+۰\times۸^۰+۴\times۸^{-۱}=۱$$

روش تبدیل ازمبنای ۱۰ به هر مبنایی

برای این کار از روش تقسیمات متوالی استفاده می‌کنیم.

مثال: عدد $(۴۱)_{۱۰}$ را به مبنای ۲ تبدیل کنید.

حل:



تقسیمات متوالی را تا جایی ادامه می‌دهیم که خارج قسمت برابر یکی از ارقام در مبنای مورد نظر شود. در این‌صورت کل اعدادی که در بالا دور آن‌ها خط کشیده شده است (باقیمانده تمام مراحل به‌علاوه خارج قسمت مرحله آخر) را به‌صورت وارون می‌نویسیم و عدد حاصل معادل باینری عدد داده شده است. $(۴۱)_{۱۰}=(۱۰۱۰۰۱)_۲$

مثال: عدد $(۲۵)_{۱۰}$ را به مبنای ۲ تبدیل کنید.

حل: در اینجا از روش سریع‌تری به‌صورت زیر استفاده می‌کنیم:

$$(۲۵)_{۱۰}=(?)_۲ \Rightarrow \begin{array}{cccccc} ۳۲ & ۱۶ & ۸ & ۴ & ۲ & ۱ \\ \hline & ۱ & ۱ & ۰ & ۰ & ۱ \end{array}$$

همان‌طورکه ملاحظه می‌کنید توان‌های دو را می‌نویسیم و در هر بار از بیشترین توان را که کوچک‌تر از عدد باشد، انتخاب می‌کنیم.

مثال: عدد $(۱۵۳)_{۱۰}$ را به مبنای ۸ تبدیل کنید.

حل: از همان روش تقسیمات متوالی مذکور به‌صورت زیر استفاده می‌کنیم:

$$\begin{array}{r|l} ۱۵۳ & ۸ \\ \cline{2-2} ۱۵۲ & ۱۹ \;\big|\; ۸ \\ \hline ۱ & ۳ \quad\; ۲ \end{array}$$

$(۱۵۳)_{۱۰}=(۲۳۱)_۸$

نکته: به‌طور کلی در تبدیل مبنای ده به مبنای r مراحل را تا جایی ادامه می‌دهیم که خارج قسمت کوچک‌تر از r شود.

نکته: روش تقسیمات متوالی تنها در رابطه با قسمت صحیح اعداد مطرح است. در تبدیل قسمت اعشاری اعداد به مبنای r باید از ضرب‌های متوالی استفاده کنیم و در هر مرحله قسمت صحیح حاصل‌ضرب را در نظر بگیریم. اگر قسمت صحیح رقمی غیر از صفر بود آن را به صفر تبدیل می‌کنیم و کار را ادامه می‌دهیم. برای روشن شدن مطلب به مثال زیر توجه کنید.

مثال: می‌خواهیم عدد $(۰/۶۸۷۵)_{۱۰}$ را به مبنای دو تبدیل کنیم.

	صحیح		کسری	ضریب
$۰/۶۸۷۵\times ۲=$	۱	+	۰/۳۷۵۰	$a_{-۱}=۱$
$۰/۳۷۵۰\times ۲=$	۰	+	۰/۷۵۰۰	$a_{-۲}=۰$
$۰/۷۵۰۰\times ۲=$	۱	+	۰/۵۰۰۰	$a_{-۳}=۱$
$۰/۵۰۰۰\times ۲=$	۱	+	۰/۵۰۰۰	$a_{-۴}=۱$

بنابراین حاصل این تبدیل عبارت است از:

$$(۰/۶۸۷۵)_{۱۰}=(۰/a_{-۱}\,a_{-۲}\,a_{-۳}\,a_{-۴})=(۰/۱۰۱۱)_۲$$

مثال: عدد $(۰/۵۱۳)$ را به مبنای هشت تبدیل کنید.

حل: با استفاده از روش ضرب‌های متوالی به‌صورت زیر عمل می‌کنیم:

	صحیح		کسری	ضریب
$۰/۵۱۳\times ۸=$	۴	+	۰/۱۰۴	$a_{-۱}=۴$
$۰/۱۰۴\times ۸=$	۰	+	۰/۸۳۲	$a_{-۲}=۰$
$۰/۸۳۲\times ۸=$	۶	+	۰/۶۵۶	$a_{-۳}=۶$
$۰/۶۵۶\times ۸=$	۵	+	۰/۲۴۸	$a_{-۴}=۵$
$۰/۲۴۸\times ۸=$	۱	+	۰/۹۸۴	$a_{-۵}=۱$
$۰/۹۸۴\times ۸=$	۷	+	۰/۸۷۲	$a_{-۶}=۷$

بنابراین حاصل این تبدیل تا ۶ رقم اعشار به‌صورت زیر است:

$$(۰/۵۱۳)_{۱۰}=(۰/۴۰۶۵۱۷...)_۸$$

نکته: برای تبدیل اعداد دهدهی که هم دارای قسمت صحیح و هم دارای قسمت اعشاری هستند، قسمت‌های صحیح و اعشاری را به‌طور جداگانه تبدیل می‌کنیم و در آخر کنار هم قرار می‌دهیم. به‌عنوان مثال با توجه به مثال‌های قبل داریم:

$$(۴۱/۶۸۷۵)_{۱۰}=(۱۰۱۰۰۱/۱۰۱۱)_۲$$

✓**تست: کدام‌یک از اعداد زیر در مبنای ۲ دارای نمایش خاتمه‌پذیر است؟** (علوم کامپیوتر سراسری ۸۰)

۱) ۰/۱۵ ۲) ۰/۱۳۵ ۳) ۰/۶۲۵ ۴) ۰/۱

حل: گزینه «۳»

نکته: اگر قسمت اعشاری توانی از ۵ باشد، عدد اعشاری در مبنای ۲ خاتمه‌پذیر خواهد بود.

مثال: بزرگ‌ترین عدد ۳ رقمی را در مبنای دو، هشت و ده بیابید.

حل:

مبنای ۱۰ : $۹۹۹ = ۱۰^{۳} - ۱$

مبنای ۲ : $۱۱۱ = ۲^{۳} - ۱$

مبنای ۸ : $۷۷۷ = ۸^{۳} - ۱$

✓**تست: برای نمایش یک عدد بیست رقمی در مبنای ۲ حداقل به چند بیت نیاز است؟** (علوم کامپیوتر سراسری ۸۱)

۱) ۶۴ ۲) ۶۵ ۳) ۶۶ ۴) ۶۷

حل: گزینه «۴»

بزرگ‌ترین عدد ۲۰ رقمی در مبنای ۱۰ برابر است با $۱۰^{۲۰} - ۱$. همچنین بزرگ‌ترین عدد n بیتی در مبنای ۲ برابر است با $۲^{n} - ۱$.
بنابراین داریم:

$$۲^{n} - ۱ \geq ۱۰^{۲۰} - ۱ \Rightarrow n \log n \geq ۲۰ \Rightarrow n \geq \frac{۲۰}{۰/۳۰۱} \simeq ۶۶/۴۳$$

جمع در مبناهای مختلف عددی

اگر مبنا r باشد برای جمع ۲ رقم دلخواه در مبنای r به تعداد دسته‌های r تایی رقم نقلی تولید می‌شود و باقی‌مانده هر چه بود عیناً پایین نوشته می‌شود که همان رقم مجموع است.

مثال: دو عدد $(۱۷۵۳)_۸$ و $(۶۳۲)_۸$ را با هم جمع کنید.

حل:

$$\begin{array}{r} ۱۱ \\ (۱۷۵۳)_۸ \\ (۶۳۲)_۸ \\ \hline ۲۶۰۵ \end{array}$$

مثال: دو عدد $(A۱B)_{۱۶}$ و $(FA۲)_{۱۶}$ را با هم جمع کنید.

حل:

$$\begin{array}{r} F\,A\,۲ \\ A\,۱\,B \\ \hline ۱۹\,B\,D \end{array}$$

همان‌طورکه در جمع ده‌دهی نیز داشتیم هر عمل جمع از سه بخش مضاف‌الیه (Augend)، مضاف (Addend) و رقم‌های نقلی (Carries) تشکیل شده است. جدول زیر جمع ارقام را در مبنای ۲ نشان می‌دهد.

$A + B = C$

$[A_i]$	$[B_i]$	Carry in	Sum	Carry out
۰	۰	۰	۰	۰
۰	۰	۱	۱	۰
۰	۱	۰	۱	۰
۰	۱	۱	۰	۱
۱	۰	۰	۱	۰
۱	۰	۱	۰	۱
۱	۱	۰	۰	۱
۱	۱	۱	۱	۱

نمایش اعداد منفی

در اینجا سه روش برای نمایش اعداد منفی ذکر می‌کنیم.

روش علامت مقدار (Signed Magnitude Methed)

این روش ساده‌ترین و در عین حال بدترین روش برای نمایش اعداد علامت دار است. اگر $N=\pm(a_{n-1}...a_0.a_{-1}...a_{-m})_r$ عددی در مبنای r باشد آن را به‌صورت زیر نمایش می‌دهیم:

$$N=(sa_{n-1}a_{n-2}...a_0.a_{-1}...a_{-m})_r$$

که در آن رقم s به رقم علامت معروف است. اگر $s=0$ آن‌گاه عدد N مثبت (Positive) است و اگر $s=r-1$ آن‌گاه عدد N منفی (Negative) خواهد بود.

از جمله مشکلات این روش این است که در این روش دو تا صفر داریم. یک صفر مثبت و دیگری صفر منفی.

نکته: با n بیت داریم:

بزرگ‌ترین عدد n بیتی = $011...11 = +2^{n-1}-1$

کوچک‌ترین عدد n بیتی = $111...11 = -(2^{n-1}-1)$

نکته: اشکال اصلی این روش آن است که الگوریتم‌های محاسباتی مانند جمع و تفریق نسبت به سایر روش‌ها پیچیده‌تر هستند. به‌عنوان مثال اگر در این سیستم بخواهیم حاصل ۱۰۱۰+۰۱۱۰ را حساب کنیم باید ابتدا بررسی کنیم که آیا دو عدد هم علامتند یا خیر. با توجه به نتیجه این بررسی و این که می‌خواهیم جمع انجام دهیم یا تفریق، دو عدد را با هم جمع یا تفریق می‌کنیم و حاصل را محاسبه می‌کنیم.

روش مکمل یک

به‌طور کلی مکمل $(r-1)$ عدد a که دارای n رقم صحیح و m رقم اعشاری در مبنای r است برابر است با:

$$[a]_{r-1}=r^n-r^{-m}-a$$

مثال: مکمل ۹ در مبنای ۱۰ عدد $a=127/30$ را به‌دست آورید.

حل:

$[127/30]_9=10^3-10^{-2}-127/30=999/99-127/30=872/69$

مثال: مکمل ۱ در مبنای ۲ عدد $a=101/01$ را به‌دست آورید.

حل:

$[111/01]_1=2^3-2^{-2}-101/01=111/11-101/01=010/10$

نکته: در سیستم مکمل یک برای یافتن قرینه یک عدد مکمل یک آن را به‌دست می‌آوریم.

مثال: عدد ۳- در سیستم مکمل یک با چهار بیت به چه صورتی نوشته می‌شود؟

حل: ابتدا عدد ۳+ را در چهار بیت به‌صورت ۰۰۱۱ می‌نویسیم. حال برای به‌دست آوردن عدد ۳- کافی است از این عدد مکمل یک بگیریم که حاصل برابر ۱۱۰۰ می‌شود.

نکته: در هر سه روش نمایش اعداد، اگر MSB برابر صفر باشد عدد مثبت و در غیراین‌صورت عدد منفی می‌باشد.

نکته: هر سه روش در نمایش اعداد مثبت تفاوتی با یکدیگر ندارند بلکه تفاوت آن‌ها در نمایش اعداد منفی می‌باشد.

مثال: اعداد چهار بیتی در سیستم مکمل یک به‌صورت زیر می‌باشند:

نمایش مکمل ۱	اعداد منفی	نمایش مکمل ۱	اعداد مثبت
۱۱۱۱	۰-	۰۰۰۰	۰+
۱۱۱۰	۱-	۰۰۰۱	۱+
۱۱۰۱	۲-	۰۰۱۰	۲+
۱۱۰۰	۳-	۰۰۱۱	۳+
۱۰۱۱	۴-	۰۱۰۰	۴+
۱۰۱۰	۵-	۰۱۰۱	۵+
۱۰۰۱	۶-	۰۱۱۰	۶+
۱۰۰۰	۷-	۰۱۱۱	۷+

نکته: با n بیت داریم:

بزرگ‌ترین عدد n بیتی = ۰۱۱...۱۱ $= +2^{n-1} - 1$

کوچک‌ترین عدد n بیتی = ۱۰۰...۰۰ $= -(2^{n-1} - 1)$

جمع در سیستم مکمل ۱

برای جمع در این سیستم لزومی به تشخیص علامت اعداد نمی‌باشد. اعداد را مانند سیستم بدون علامت با هم جمع می‌کنیم و رقم نقلی حاصل را با حاصل، جمع می‌کنیم.

مثال: اگر $A = 1010$ و $B = 0100$ حاصل جمع این دو عدد را در سیستم مکمل ۱ حساب کنید.

حل:

$$\begin{array}{rr} & (1010)_2 \\ + & \underline{(0100)_2} \\ & (1110)_2 \end{array}$$

مثال: اگر $A = 1101$ و $B = 0110$ حاصل جمع این دو عدد را در سیستم مکمل ۱ حساب کنید.

حل:

$$\begin{array}{lr} & (1101)_2 \\ + & \underline{(0110)_2} \\ c = 1 & (0011)_2 \\ + & \underline{(0001)_2} \\ & (0100)_2 \end{array}$$

بررسی سرریز (Over flow) در هر سه روش

در جمع دو عدد در دو حالت سرریز رخ داده است:

۱- جمع دو عدد منفی، مثبت شود.

۲- جمع دو عدد مثبت، منفی شود.

در جمع یک عدد مثبت و یک عدد منفی هیچ‌گاه سرریز رخ نمی‌دهد و حاصل همواره در بازه مجاز خواهد بود.

در کامپیوتر سرریز را با یک فلگ به نام V نشان می‌دهند. اگر $V=1$ سرریز رخ داده است و در غیراین‌صورت، سرریز نداریم و حاصل محاسبات صحیح است.

تفریق در سیستم مکمل ۱

به‌طور کلی برای تفریق، عدد اول را با مکمل عدد دوم جمع می‌کنیم. این کار را به‌صورت زیر انجام می‌دهیم:

$$A-B=A+\overline{B}$$

مثال: اگر $A=1010100$ و $B=1000011$ حاصل A-B و B-A را با استفاده از روش مکمل ۱ حساب کنید.

حل: ابتدا از عدد دوم مکمل می‌گیریم و سپس آن را با عدد اول جمع می‌کنیم.

$$\begin{array}{r} A:1010100 \\ +\ 0111100 \\ \hline 10010000 \\ +\qquad 1 \\ \hline 0010001 \end{array}$$

$$\begin{array}{r} B:1000011 \\ 0101011 \\ \hline 1101110 \end{array}$$

$$B-A=1000011-1010100$$

$$B-A=10001$$

روش مکمل دو

به‌طور کلی مکمل r عدد a که دارای n رقم صحیح و m رقم اعشار در مبنای r می‌باشد، برابر است با:

$$[a]_r=r^n-a$$

مثال: مکمل ۱۰ عدد $a=125/70$ را به‌دست آورید.

حل:

$$[125/70]_{10}=10^3-125/70=874/30$$

نکته: برای محاسبه سریع می‌توان از سمت راست اولین رقم غیر صفر را از ۱۰ کم کرد و بقیه ارقام را از ۹ کم نمود.

مثال: مکمل ۲ عدد $a=1110/100$ را به‌دست آورید.

حل:

$$[1110/100]_2=2^4-1110/100=0001/100$$

نکته: برای مکمل ۲ می‌توان ابتدا مکمل یک را حساب نمود و سپس حاصل را با یک جمع نمود. در حالت کلی رابطه زیر برقرار است:

$$[a]_r=[a]_{r-1}+r^{-m}$$

و در حالتی‌که عدد قسمت اعشاری نداشته باشد داریم:

$$[a]_r = [a]_{r-1} + 1$$

نکته: در سیستم مکمل دو برای یافتن قرینه یک عدد مکمل دو آن را به‌دست می‌آوریم.

مثال: عدد ۳- در سیستم مکمل دو با چهار بیت به چه صورتی نوشته می‌شود؟

حل: ابتدا عدد ۳+ را در چهار بیت به‌صورت ۰۰۱۱ می‌نویسیم. حال برای به‌دست آوردن عدد ۳- کافی است از این عدد مکمل دو بگیریم که حاصل برابر ۱۱۰۱ می‌شود.

مثال: اعداد چهار بیتی در سیستم مکمل دو به‌صورت زیر می‌باشند:

نمایش مکمل ۲	اعداد منفی	نمایش مکمل ۲	اعداد مثبت
۰۰۰۰	−۰	۰۰۰۰	+۰
۰۰۰۱	−۱	۰۰۰۱	+۱
۰۰۱۰	−۲	۰۰۱۰	+۲
۰۰۱۱	−۳	۰۰۱۱	+۳
۰۱۰۰	−۴	۰۱۰۰	+۴
۰۱۰۱	−۵	۰۱۰۱	+۵
۰۱۱۰	−۶	۰۱۱۰	+۶
۰۱۱۱	−۷	۰۱۱۱	+۷
۱۰۰۰	−۸		

نکته: بهترین روش برای نمایش اعداد علامت‌دار این سیستم می‌باشد و در اکثر کامپیوترهای امروزی از آن استفاده می‌شود. همان‌طورکه ملاحظه می‌کنید در این سیستم تنها یک نمایش برای صفر وجود دارد.

نکته: با n بیت در سیستم مکمل ۲ داریم:

$$\text{بزرگ‌ترین عدد n بیتی} = 011\ldots11 = +2^{n-1} - 1$$

$$\text{کوچک‌ترین عدد n بیتی} = 100\ldots00 = -2^{n-1}$$

مثال: نمایش عدد $-(18)_{10}$ با سیستم مکمل دو به‌صورت زیر می‌باشد. (با ۸ رقم)

حل:

$$+(18)_{10} = (00010010)_{2cns}$$

$$-(18)_{10} = (11101110)_{2cns}$$

مثال: عدد $(0/479)_{10}$ را به مبنای ۸ تبدیل کنید.

حل:

	صحیح		کسری	ضریب
$0/479 \times 8 =$	۳	+	۰/۸۳۲	$a_{-1} = 3$
$0/832 \times 8 =$	۶	+	۰/۶۵۶	$a_{-2} = 6$
$0/656 \times 8$	۵	+	۰/۲۴۸	$a_{-3} = 5$
$0/248 \times 8 =$	۱	+	۰/۹۸۴	$a_{-4} = 1$

$$\Rightarrow (0/479)_{10} = (0/3651\ldots)_8$$

مثال: عدد $(۱۸/۶)_۹$ را به مبنای ۱۱ ببرید.

حل: برای این کار ابتدا این عدد را به مبنای ۱۰ و سپس حاصل را به مبنای ۱۱ تبدیل می‌کنیم.

$$(۱۸/۶)_۹ = ۱\times ۹^۱ + ۸\times ۹^۰ + ۶\times ۹^{-۱} = ۹+۸+۰/\overline{۶} = (۱۷/\overline{۶})_{۱۰}$$

حال باید قسمت صحیح (۱۷) و قسمت اعشاری (۰/۶۶۶۶....) را به‌طور جداگانه به مبنای ۱۱ تبدیل کنیم.

$$\begin{array}{r|l} ۱۷ & ۱۱ \\ \cline{2-2} \underline{۱۱} & ۱ \\ ۶ & \end{array} \qquad \rightarrow (۱۷)_{۱۰} = (۱۶)_{۱۱}$$

ضریب	کسری		صحیح	
$a_{-۱} = ۷$	۰/۳۲۶	+	۷	۰/۶۶۶×۱۱=
$a_{-۲} = ۳$	۰/۵۸۶	+	۳	۰/۳۲۶×۱۱=
$a_{-۳} = ۶$	۰/۴۴۶	+	۶	۰/۵۸۶×۱۱

$$\Rightarrow (۱۸/۶)_۹ = (۱۶/۷۳۶...)_{۱۱}$$

نکته: برای تبدیل عدد N در مبنای B به مبنای A (با فرض آنکه $B = A^k$ و k عدد صحیح مثبت است) کافی است هر رقم عدد N در مبنای B رابا k رقم در مبنای A جایگزین کنیم.

مثال: عدد $(۰۰۱۰۱۰۱۱۱/۱۰۰)_۲$ را به مبنای ۸ تبدیل کنید.

حل: کافی است ارقام این عدد را به‌صورت زیر به دسته‌های سه تایی تقسیم کنیم:

$(۱۰۱۰۱۰۱۱)_۲ = (۲۵۳)_۸$

$(۰۰۱۰۱۰۱۱۱)_۲ - (۱۲۷/۴)_۸$

مثال: عدد $(۱۱۰۰۰۱۰۱۱۱۱/۱۰۱۱۱)_۲$ را به مبنای ۱۶ تبدیل کنید.

حل: در اینجا دقت کنید که ارقام به‌صورت کامل درون دسته‌های ۴تایی قرار نمی‌گیرند برای حل این مشکل یک رقم صفر در سمت چپ عدد و سه رقم صفر در سمت راست عدد اضافه می‌کنیم تا بتوانیم به‌طور کامل آن را به دسته‌هایی ۴ تایی تقسیم کنیم.

$(۰۱۱۰۰۰۱۰۱۱۱۱/۱۰۱۱۱۰۰۰)_۲ = (۶۲F/B۸)_{۱۶}$

$(۱۰۱۰۱۰۱۱)_۲ = (AB)_{۱۶}$

تذکر: کارکردن با اعداد دودویی، از آن‌جایی‌که تعداد ارقامشان سه یا چهار برابر عدد معادلشان در مبنای ۱۰ می‌باشد، مشکل است. با این وجود کامپیوترهای دیجیتال از اعداد دودویی استفاده می‌کنند و گاهی نیز لازم است اپراتورها، مستقیماً با این اعداد دودویی ارتباط برقرار کنند. یک راه برای نگهداری سیستم دودویی در کامپیوترها این است که از سیستمی که در نکات قبل ذکر شد، استفاده کنیم. با این روش انسان می‌تواند بر حسب اعداد مبنای ۱۶ یا ۸ فکر کرده و در مواقعی که ارتباط مستقیم با کامپیوتر لازم است این اعداد را به مبنای دو تبدیل کند. برای ساده‌سازی عمل تفریق در کامپیوترهای دیجیتال از روش مکمل کردن استفاده می‌کنند. در همین‌جا بر این نکته تأکید می‌کنیم که نمایش اعداد مثبت در هیچ‌یک از این سیستم‌ها با هم تفاوتی ندارند، آنچه که تفاوت بین این سیستم‌ها می‌باشد، نمایش اعداد منفی است.

جمع در سیستم مکمل ۲

برای جمع در این سیستم لزومی به تشخیص علامت اعداد نمی‌باشد. اعداد را مانند سیستم بدون علامت با هم جمع می‌کنیم و از رقم نقلی حاصل صرف‌نظر می‌کنیم.

مثال: اگر $A = ۱۰۱۰$ و $B = ۰۱۰۰$ حاصل جمع این دو عدد را در سیستم مکمل ۲ حساب کنید.

حل:

$$\begin{array}{l} (۱۰۱۰)_۲ \\ \underline{(۰۱۰۰)_۲} \\ (۱۱۱۰)_۲ \end{array}$$

مثال: اگر $A=۱۱۰۱$ و $B=۰۱۱۰$ حاصل جمع این دو عدد را در سیستم مکمل ۲ حساب کنید.

حل:

$$\begin{array}{rr} & (۱۱۰۱)_۲ \\ + & \underline{(۰۱۱۰)_۲} \\ c=۱ & (۰۰۱۱)_۲ \end{array}$$

از رقم نقلی به وجود آمده صرف‌نظر می‌کنیم.

مثال: اگر $A=۱۱۰۱$ و $B=۰۱۱۰$ حاصل جمع این دو عدد را در سیستم مکمل ۲ حساب کنید.

حل:

$$\begin{array}{rr} & (۱۱۰۱)_۲ \\ + & \underline{(۰۱۱۰)_۲} \\ c=۱ & (۰۰۱۱)_۲ \end{array}$$

از رقم نقلی به وجود آمده صرف‌نظر می‌کنیم.

مثال: اگر $A=۱۰۰۱$ و $B=۱۰۱۰$ حاصل جمع این دو عدد را در سیستم مکمل ۲ حساب کنید.

حل:

$$\begin{array}{rr} & (۱۰۰۱)_۲ \\ + & \underline{(۱۰۱۰)_۲} \\ & (۰۰۱۱)_۲ \end{array}$$

از آن‌جایی‌که حاصل جمع دو عدد منفی، مثبت شده است، بنابراین سرریز رخ داده است. روش دیگری برای تشخیص سرریز در سیستم مکمل ۲ این است که هرگاه رقم نقلی مرحله $n-۱$ام مخالف رقم نقلی مرحله n ام باشد، سرریز رخ داده است. بنابراین در این سیستم داریم:

$$V=C_n \oplus C_{n-۱}$$

تفریق در سیستم مکمل ۲

در این سیستم نیز از روش مکمل به‌صورت زیر استفاده می‌کنیم:

$$A-B=A+\overline{B}+۱$$

مثال: اگر $A=۱۰۱۰۱۰۰$ و $B=۱۰۰۰۰۱۱$ حاصل **A-B** و **B-A** را با استفاده از روش مکمل ۱ حساب کنید.

حل: ابتدا از عدد دوم مکمل می‌گیریم و سپس آن را با عدد اول جمع می‌کنیم و حاصل را با یک جمع می‌کنیم.

$$\begin{array}{r} A:۱۰۱۰۱۰۰ \\ +\ \underline{۰۱۱۱۱۰۰} \\ ۰۰۱۰۰۰۰ \\ +\ \underline{\qquad ۱} \\ ۰۰۱۰۰۰۱ \end{array}$$

$$\begin{array}{r} B:۱۰۰۰۰۱۱ \\ \underline{۰۱۰۱۰۱۱} \\ ۱۱۰۱۱۱۰ \end{array}$$

$$B-A=۱۰۰۰۰۱۱-۱۰۱۰۱۰۰$$

$$B-A=۱۰۰۰۱$$

نکته: فلگ‌های سیستم عبارتند از:

V	C	Z	S=N
سرریز	رقم نقلی خروجی	صفر بودن حاصل	علامت

این فلگ‌ها با توجه به نتیجه محاسبات مقداردهی می‌شوند.

نکته: اگر A و B دو عدد در سیستم مکمل ۲ باشند و داشته باشیم $A-B=A+\overline{B}+1$ آن‌گاه وضعیت فلگ‌ها به‌صورت زیر خواهند بود:

رابطه	وضعیت فلگ‌ها
$A \geq B$	$S \oplus V = 0$
$A < B$	$S \oplus V = 1$
$A = B$	$Z = 1$
$A \neq B$	$Z \neq 0$

در حالتی‌که A و B علامت دار هستند در رابطه با C نمی‌توان اظهار نظر کرد.

نکته: اگر A و B دو عدد بی علامت باشند و داشته باشیم $A-B=A+\overline{B}+1$ آن‌گاه وضعیت فلگ‌ها به‌صورت زیر خواهند بود:

رابطه	وضعیت فلگ‌ها
$A \geq B$	$C = 1$
$A < B$	$C = 0$

نکته: اگر A-B را بدون تبدیل به جمع انجام دهیم آن‌گاه C به معنای قرض گرفتن می‌باشد و داریم:

رابطه	وضعیت فلگ‌ها
$A \geq B$	$C = 0$
$A < B$	$C = 1$

اگر در سوال چهارگزینه‌ای عنوان نشده بود که از کدام روش تفریق را انجام دهیم به‌طور پیش فرض از روش اول انجام می‌دهیم.

ضرب دودویی

ضرب اعداد دودویی هم‌چون ضرب اعداد دهدهی انجام می‌شود. هر بیت مضروب در کم‌ارزش‌ترین بیت مضروب فیه ضرب می‌شود. چنین حاصل‌ضربی، حاصل‌ضرب جزیی خوانده می‌شود. حاصل‌ضرب‌های جزیی هر بار یک مکان به چپ انتقال می‌یابند. حاصل‌ضرب نهایی از جمع حاصل‌ضرب‌های جزیی به‌دست می‌آید.

مثال: حاصل‌ضرب $A=11010$ در $B=1010$ را حساب کنید.

حل:

```
        1 1 0 1 0  <-> Multiplicand
      ×   1 0 1 0  <-> Multiplier
-------------------
        0 0 0 0 0
    + 1 1 0 1 0
    +0 0 0 0 0
  +1 1 0 1 0
-------------------
1 0 0 0 0 0 1 0 0  <-> Product
```

حاصل‌ضرب دو عدد دو بیتی در حالت کلی به‌صورت زیر می‌باشد:

$$\begin{array}{cccc} & & B_۱ & B_۰ \\ & & A_۱ & A_۰ \\ \hline & & A_۰B_۱ & A_۰B_۰ \\ & A_۱B_۱ & A_۱B_۰ & \\ \hline C_۳ & C_۲ & C_۱ & C_۰ \end{array}$$

مثال: به جمع و تفریق زیر در مبنای ۸ توجه کنید:

$$\begin{array}{ccccc} ۱ & ۱ & ۱ & & \\ & ۵ & ۴ & ۷ & ۱ \\ + & ۳ & ۷ & ۵ & ۴ \\ \hline (۱ & ۱ & ۴ & ۴ & ۵)_۸ \end{array}$$

$$\begin{array}{ccccc} & ۶ & ۱۲ & ۴ & ۹ \\ & \cancel{۷} & \cancel{۴} & \cancel{۵} & \cancel{۱} \\ - & ۵ & ۶ & ۴ & ۳ \\ \hline & ۱ & ۶ & ۰ & ۶ \end{array}$$

مثال: به جمع و تفریق زیر در مبنای ۱۶ توجه کنید:

$$\begin{array}{ccccc} ۱ & ۰ & ۱ & ۱ & \\ & ۵ & B & A & ۹ \\ + & D & ۰ & ۵ & ۸ \\ \hline ۱ & ۲ & C & ۰ & ۱ \end{array}$$

$$\begin{array}{ccccc} & ۹ & ۲۱ & ۱۰ & ۲۵ \\ & \cancel{A} & \cancel{۵} & \cancel{B} & \cancel{۹} \\ - & ۵ & ۸ & ۰ & D \\ \hline & ۴ & D & A & C \end{array}$$

شرایط سرریز (Over flow Condition)

با توجه به محدوده اعداد در سیستم مکمل ۲، اگر حاصل عملیاتی از $۲^{n-۱}-۱$ بیشتر شود، گوییم سرریز و اگر از $-۲^{n-۱}$ کمتر شود، گوییم زیر ریز رخ داده است. حال می‌خواهیم سه حالت را در جمع و تفریق بررسی کنیم. در ادامه فرض بر این است که $B,C \geq ۰$ داریم:

حالت ۱: $A = B + C$

واضح است که اگر $B + C > ۲^{n-۱} - ۱$ شود، آن‌گاه سرریز رخ داده است. بنابراین در این حالت امکان سرریز وجود دارد و باید به آن توجه شود. مثلاً می‌خواهیم جمع $(۷)_{۱۰}$ و $(۴)_{۱۰}$ را با ۵ بیت در سیستم مکمل ۲ باینری انجام دهیم، داریم:

$$(۷)_{۱۰} = +(۰۱۱۱)_۲ = (۰۰۱۱۱)_{۲cns}$$
$$(۴)_{۱۰} = +(۰۱۰۰)_۲ = (۰۰۱۰۰)_{۲cns}$$
$$(۷)_{۱۰} + (۴)_{۱۰} = (۰۰۱۱۱) + (۰۰۱۰۰) = (۰۱۰۱۱)$$

در اینجا هیچ سرریزی رخ نمی‌دهد، زیرا حاصل $۰۱۰۱۱ = +۱۱$ کوچک‌تر از $۲^۴ - ۱ = ۱۵$ است ولی حال به مثال زیر توجه کنید:

$$(۹)_{۱۰} = +(۱۰۰۱)_۲ = (۰۱۰۰۱)_{۲cns}$$
$$(۸)_{۱۰} = +(۱۰۰۰)_۲ = (۰۱۰۰۰)_{۲cns}$$

در اینجا سرریز رخ داده است بنابراین در حالت کلی داریم:

بنابراین برای تصحیح باید عدد ۶ یا $(۱۱۰)_۲$ به حاصل جمع اضافه شود و در صورت لزوم رقم نقلی نیز تولید خواهد کرد. دلیل این است که اختلاف بین یک رقم نقلی در با ارزش‌ترین مکان بیتی حاصل از جمع دودویی و نقلی دهدهی برابر است با: ۶=۱۰-۱۶ به عبارت دیگر چون ۶ حالت نامجاز دارد باید ۶ تا به آن اضافه شود تا دور زده و به حالت صحیح خود که بین صفر تا ۹ است بازگردد.

$$(۹)_{۱۰} + (۸)_{۱۰} \Rightarrow (۱۰۰۱)_۲ + (۱۰۰۰)_۲ = (۱۰۰۰۱) + ۱۱۰ = \underbrace{۱}_{۱}\underbrace{۰۱۱۱}_{۷} = ۱۷$$

حالت ۲: $A = B - C$

در این حالت می‌نویسیم:

$$\mathbf{A = (B)_۲ + (-(C)_۲) = (B)_۲ + [C]_۲ = (B)_۲ + ۲^n - (C)_۲ = ۲^n + (B-C)_۲}$$

حال دو حالت ممکن است رخ دهد که عبارتند از:

الف) اگر $B > C$ آن‌گاه $A > ۲^n$ و از رقم نقلی صرف‌نظر می‌کنیم. بنابراین:

$$(A)_۲ = (B)_۲ + [C]\Big|_{\text{carry discarded}}$$

ب) اگر $B < C$ آن‌گاه:

$$A = ۲^n - (C-B)_۲ = [C-B]_۲ \Rightarrow A = -(C-B)_۲$$

در این حالت هیچ رقم نقلی وجود نخواهد داشت.

مثال: با فرض داشتن ۵ رقم داریم، بنابراین:

$$(۱۴)_{۱۰} - (۹)_{۱۰} = (۱۴)_{۱۰} + (-(۹)_{۱۰}) = +(۱۱۱۰)_۲ - (۱۰۰۱)_۲ = (۰۱۱۱۰)_{۲cns} + (۱۰۱۱۱) + (۰۰۱۰۱) + \text{carry} + (۰۱۰۱) = +۵$$

همان‌طورکه دیدید، در این مثال Carry را حذف کردیم.

مثال: فرض کنید ۵ رقم داریم. بنابراین:

$$(۹)_{۱۰} - (۱۴)_{۱۰} = (۹)_{۱۰} + (-(۱۴)_{۱۰}) = +(۱۰۰۱)_۲ + (-(۱۱۱۰)_۲) = (۰۱۰۰۱)_{۲cns} + (۱۰۰۱۰)_{۲cns} + (۱۱۰۱۱)_{۲cns}$$
$$= -(۰۱۰۱)_۲ = -(۵)_{۱۰}$$

مثال:

$$(۰۰۱۰۰)_{۲cns} - (۱۰۱۱۰)_{۲cns} = (۰۰۱۰۰)_{۲cns} + (۰۱۰۱۰)_{۲cns} = (۰۱۱۱۰)_{۲cns} = +(۱۱۱۰)_۲ = +(۱۴)_{۱۰}$$

نکته: در حالت ۲ هیچ‌گاه سرریز رخ نمی‌دهد.

حالت ۳: $A = -B - C$

در این حالت با مکمل کردن داریم:

$$A = [B]_۲ + [C]_۲ = ۲^n - (B)_۲ + ۲^n - (C)_۲ = ۲^n + ۲^n - (B+C)_۲ = ۲^n + [B+C]_۲$$

بیت نقلی ۲^n حذف شود.

سرریز زمانی رخ می‌دهد که رقم علامت حاصل صفر شود.

مثال: محاسبات خواسته شده را در ۵ بیت انجام دهید.

$$۱)\ -(۷)_{۱۰} - (۸)_{۱۰} = ?$$

حل:

$$(-(۷)_{۱۰}) + (-(۸)_{۱۰}) = (-(۰۱۱۱)_۲) + (-(۱۰۰۰)_۲) = (۱۱۰۰۱)_{۲cns} + (۱۱۰۰۰)_{۲cns} + (۱۰۰۰۱)_{۲cns} + \text{Carry} = -(۱۱۱۱)_۲ = -(۱۵)_{۱۰}$$

$$۲)\ -(۱۲)_{۱۰} - (۵)_{۱۰} = ?$$

$$= (۰۱۱۱۱)_{۲cns} + \text{Carry}\ (-(۱۲)_{۱۰}) + (-(۵)_{۱۰}) = (-(۱۱۰۰)_۲ + (-(۰۱۰۱)_۲) = (۱۰۱۰۰)_{۲cns} + (۱۱۰۱۱)_{۲cns}$$

از آن‌جایی‌که در این مثال بیت علامت صفر شده است بنابراین خطای سرریز روی داده است.

نکته: در حالت ۳ همواره رقم نقلی به‌وجود می‌آید که از آن صرف‌نظر می‌کنیم. هم‌چنین اگر علامت حاصل مثبت شود، خطای سرریز رخ داده است.

مثال: فرض کنید $A=(25)_{10}$ و $B=-(46)_{10}$، حاصل $A+B$، $A-B$، $B-A$ و $-A-B$ را محاسبه کنید.

حل:

$$A=+(25)_{10}=(00011001)$$
$$-A=(11100111)_{2cns}$$
$$B=(101110)$$
$$-B=-(46)_{10}=(010010)$$
$$A+B=(00011001)_{2cns}+(010010)_{2cns}=(101011)_{2cns}\xrightarrow{\text{مکمل ۲ گرفته}}(010101)=-(21)_{10}$$
$$A-B=A-(-B)=A+B(00011001)_{2cns}+(101110)_{2cns}=(01000111)_{2}=(-71)_{10}$$
$$B-A=B+(-A)=(101110)_{2cns}+(11100111)_{2cns}=(10101)=(21)_{10}$$
$$-A-B=(-A)+(-B)=(11100111)+(10010)=-(10111001)$$

به‌طورکلی برای حالات مختلف سیستم مکمل II داریم:

سرریز ؟	شرایط	بیت علامت	رقم نقلی	حالت
no	$B+C\le 2^{n-1}-1$	0	0	$B+C$
Yes	$B+C\ge 2^{n-1}-1$	1	0	
no	$B\le C$	0	1	$B-C$
no	$B>C$	1	0	
no	$-(B+C)\ge -2^{n+1}$	1	1	$-B-C$
Yes	$-(B+C)<-2^{n+1}$	0	1	

نکته: در حالت کلی در سیستم نمایش مکمل ۲، در عملیات جمع و تفریق از رقم نقلی خروجی صرف‌نظر می‌کنیم .حال سرریز عمل $A=B+C$ در دو حالت زیر رخ می‌دهد:

الف) اگر علامت B و C مثبت باشد و علامت A منفی شود سرریز رخ داده است.

ب) اگر علامت B و C منفی باشد و علامت A مثبت شود سرریز رخ داده است.

کدهای عددی (Numeric Codes)

کد، استفاده سیستماتیک از مجموعه‌ای از نمادها برای نمایش اطلاعات است. به‌عنوان مثال علائم چراغ راهنمایی (قرمز: ایستادن، زرد: اخطار، سبز: رفتن) یک کدگذاری می‌باشد. هر عنصر گسسته مستقل اطلاعاتی را می‌توان با استفاده از کدهای دودویی نشان داد. کدها باید به‌صورت دودویی باشند تا کامپیوترها قادر به نگهداری آن‌ها باشند. کدها، نماد اطلاعاتی که باید ارائه دهند را عوض می‌کنند.

نکته: برای کد کردن 2^n مقدار مستقل، حداقل به n بیت نیاز است.

نکته: از این کدها برای نمایش اعداد استفاده می‌شود و به دو صورت اعداد با ممیز ثابت و اعداد با ممیز شناور تقسیم می‌شوند.

اعداد با ممیز ثابت

برای نمایش اعداد صحیح علامت‌دار یا اعداد اعشاری محض (اعداد اعشاری‌که قسمت صحیح ندارند) می‌توان از این نحوه کدگذاری استفاده کرد. در این سیستم مقدار عددی می‌تواند به‌صورت مکمل I یا مکمل II یا علامت مقدار نوشته شود و فرمت آن به‌صورت زیر است:

(ممیز ضمنی) + (مقدار) + (بیت علامت): فرمت اعداد صحیح علامت‌دار

(مقدار) + (ممیز ضمنی) + (بیت علامت): فرمت اعداد اعشاری محض

از آن‌جایی‌که کامپیوتر می‌داند که عدد نوشته شده صحیح یا اعشاری محض است بنابراین مکان ممیز را می‌داند و ممیز ضمنی نوشته نمی‌شود.

مثلاً عدد صحیح ۰۱۱۱۰۰۱۱ در این فرمت برابر عدد ۱۱۱۰۰۱۱+ است و عدد اعشاری محض ۰۱۱۱۰۰۱۱ در این فرمت برابر عدد ۰/۱۱۱۰۰۱+ است.

نمایش افزونه (Excess) یا بایاس شده (Biased)

افزونه k نمایش یک کُد C، اضافه کردن k به کلمات کد C است. از این روش مکرراً در نمایش توان اعـداد ممیـز شـناور اسـتفاده می‌گردد. در جدول زیر نمایش افزونه ۸ آورده شده است:

افزونه ۸	سیستم مکمل II	کدگذاری دهدهی
۱۱۱۱	۰۱۱۱	+۷
۱۱۱۰	۰۱۱۰	+۶
۱۱۰۱	۰۱۰۱	+۵
۱۱۰۰	۰۱۰۰	+۴
۱۰۱۱	۰۰۱۱	+۳
۱۰۱۰	۰۰۱۰	+۲
۱۰۰۱	۰۰۰۱	+۱
۱۰۰۰	۰۰۰۰	۰
۰۰۰۱	۱۰۰۱	−۱
۰۰۱۰	۱۰۱۰	−۲
۰۰۱۱	۱۰۱۱	−۳
۰۱۰۰	۱۱۰۰	−۴
۰۱۰۱	۱۱۰۱	−۵
۰۱۱۰	۱۱۱۰	−۶
۰۱۱۱	۱۱۱۱	−۷
۰۰۰۰	۱۰۰۰	−۸

نکته: این کدها وزن‌دار نمی‌باشند ولی خود مکمل‌اند.

اعداد ممیز شناور

برای نمایش عدد N در فرمت ممیز شناور آن را به‌صورت $M \times V^E$ نمایش می‌دهیم که در آن، M را مـانتیس (Mantissa)، E را نما (Exponent) و V را مبنا گویند. بنابراین به‌طور کلی اگر $N = \pm(a_{n-۱}\dots a_۰.a_{-۱}\dots a_{-m})_r$ آن‌گاه داریم:

$$N = \pm(.a_{n-۱}\dots a_{-m}) * r^n$$

که در آن معمولاً M را در سیستم علامت مقدار نمایش می‌دهند و $M = (S.a_{n-۱}\dots.a_{-m})_{rsm}$ که در آن S مـی‌توانـد صـفر (برای نمایش مانتیس مثبت) و یا یک (برای نمایش مانتیس منفی) باشد و E اغلب به‌صورت افزونه $۲^{e-۱}$ خواهد بـود کـه در آن e تعداد بیت‌های نما می‌باشد.

نرمال کردن مانتیس

به فرم‌های نمایش زیر توجه کنید:

$$N = (۱۱۰۱/۰۱۰۱)_۲ = (۰/۱۱۰۱۰۱۰۱)_۲ \times ۲^۴ = (۰/۰۱۱۰۱۰۱۰۱)_۲ \times ۲^۵$$

همان‌گونه که ملاحظه می‌کنید، یک عدد را می‌توان به فرم‌های مختلف نمایش داد ولی برای یکتا بودن نمایش همواره مانتیس را به گونه‌ای می‌نویسیم که بیت با ارزش‌تر آن (MSD) یک باشد. به این کار، نرمال کردن مانتیس گویند. بنابراین از نمایش دوم برای مانتیس استفاده می‌کنیم. بنابراین همواره $0/5 \le |M| < 1$ خواهد بود.

استانداردهای IEEE754 برای نمایش اعداد ممیز شناور

این استاندارد به دو صورت زیر می‌باشد:

الف) دقت ساده (Single Precision)

در این سیستم عدد به‌صورت زیر نمایش داده می‌شود.

S	E	M

که S نشان‌دهنده علامت مانتیس است و یک بیت می‌باشد. E، ۸ بیت است و M، ۲۳ بیت می‌باشد. بنابراین در کل عدد ۳۲ بیتی می‌باشد.

ب) دقت مضاعف (Double Precision)

که S نشان‌دهنده علامت مانتیس است و یک بیت می‌باشد. E، ۱۱ بیت است و M، ۵۲ بیت می‌باشد. بنابراین در کل عدد ۶۴ بیتی می‌باشد.

مثال: عدد $(101100.101)_2$ را به فرمت ممیز شناور بنویسید. فرض کنید تعداد بیت‌های نما (E) برابر ۵ و تعداد بیت‌های مانتیس (M) برابر ۱۰ است.

حل: این کار را به‌صورت زیر انجام می‌دهیم:

مانتیس نرمال شده $= 0/101100101 \times 2^6$

نما $= 00110 \xrightarrow[\text{افزونه } 2^{5-1}]{} 10110$

نمایش عدد =

۰	۱۰۱۱۰	۱۰۱۱۰۰۱۰۱۰

نکته: از آن‌جایی‌که مانتیس قسمت اعشاری و نما یک عدد صحیح است، در صورت کم بودن بیت‌های آنها، به سمت راست مانتیس و به سمت چپ نما، صفر اضافه می‌کنیم.

کد وزن‌دار: کدهای وزن‌دار کدهایی هستند که با دانستن وزن بیت‌ها در نمایش دودویی بتوان معادل دهدهی آن‌ها را به‌دست آورد.

کد خود مکمل: کدی است که مکمل I آن در نمایش دودویی برابر مکمل ۹ آن در نمایش دهدهی باشد. شرط لازم و کافی برای خود مکمل بودن یک کد این است که اولاً وزن‌دار باشد و ثانیاً مجموع وزن‌های آن برابر ۹ باشد.

کد انعکاسی: کدهایی که در افزایش و یا کاهش یک واحد، صرفاً نیاز به تغییر یک بیت می‌باشد، کد انعکاسی نام دارند (مانند کدگری).

روش‌های کدگذاری

کد BCD (Binary Coded Decimal)

از این کد برای نمایش رقم‌های دهدهی ۰ تا ۹ استفاده می‌شود و در آن از ۴ بیت استفاده می‌شود. این کد، یک کد وزن‌دار می‌باشد و وزن بیت‌های آن به ترتیب از چپ به راست برابر است با: ۱ و ۲ و ۴ و ۸. بنابراین به آن کُد ۸۴۲۱ نیز می‌گویند. این کد به نام کد NBCD نیز شناخته می‌شود.

کدهای BCD در زیر آمده‌اند:

۰=۰۰۰۰ ۱=۰۰۰۱ ۲=۰۰۱۰ ۳=۰۰۱۱
۴=۰۱۰۰ ۵=۰۱۰۱ ۶=۰۱۱۱ ۷=۰۱۱۱
۸=۱۰۰۱ ۹=۱۰۰۰

از این سیستم در پردازنده‌هایی که محاسبات دهدهی انجام می‌دهند، استفاده می‌شود.

مثال: عدد $(۸۷۹۰)_{۱۰}$ را به کد BCD نمایش دهید.

حل:

۸۴۲۱ ۸۴۲۱ ۸۴۲۱ ۸۴۲۱
۱۰۰۰ , ۰۱۱۱ , ۱۰۰۱ , ۰۰۰۰
نمایش عدد ۸ — نمایش عدد ۷ — نمایش عدد ۹ — نمایش صفر

نکته: کدهای BCD وزن‌دار ولی غیر خود مکمل هستند. برای یافتن مکمل ۹ یک رقم در این کد دو روش وجود دارد:

روش اول: ابتدا مکمل I عدد داده شده را محاسبه می‌کنیم و سپس حاصل را با $(۱۰۱۰)_۲$ جمع می‌کنیم و از رقم نقلی حاصل صرف‌نظر می‌کنیم.

مثال: مکمل ۹ عدد $(۱۰۰۰)_{BCD}$ را بیابید.

حل:

$$۱۰۰۰ \xrightarrow{\text{مکمل I گرفتن}} ۰۱۱۱ \xrightarrow{+۱۰۱۰} ۱۰۰۰۱ \xrightarrow[\text{از رقم نقلی}]{\text{صرف نظر کردن}} ۰۰۰۱$$

روش دوم: ابتدا عدد را با $(۰۱۱۰)_۲$ جمع می‌کنیم و سپس مکمل I عدد حاصل را حساب می‌کنیم و از رقم نقلی حاصل صرف‌نظر می‌کنیم.

مثال: مکمل ۹ عدد $(۱۰۰۰)_{BCD}$ را بیابید.

حل:

$$۱۰۰۰ \xrightarrow{+۰۱۱۰} ۱۱۱۰ \xrightarrow{\text{مکمل I گرفتن}} ۰۰۰۱$$

ASCII (American Standard Code for Information Interchange)

در کدگذاری‌ها اغلب از این سیستم کدگذاری استفاده می‌شود. برای یافتن خطاها در این روش از ۸ بیت (بیت‌های توازن) استفاده می‌شود. قسمتی از جدول ASCII در زیر آمده است:

Character	Binary Code	Hexadecimal Code
D	۱۰۰۰۱۰۰	۴۴
I	۱۱۰۱۰۰۱	۶۹
G	۱۱۰۰۱۱۱	۶۷
T	۱۱۱۰۱۰۰	۷۴
A	۱۱۰۰۰۰۱	۶۱
L	۱۱۰۱۱۰۰	C۶

علاوه بر کدهای توضیح داده شده، سه کد وزن‌دار دیگر نیز در جدول زیر آورده شده است که در یکی از آن‌ها از وزن‌های ۲- و ۱- استفاده شده است که با توجه به این وزن‌ها، مقدار کد محاسبه می‌شود.

کد افزونه ۳

در این کد به هر رقم چهار بیتی سه واحد اضافه می‌شود. افزایش ۳ واحد به کد باعث ایجاد خاصیت خود مکملی می‌شود.

مثال: عدد $(۲۷)_{۱۰}$ را به کد بدون وزن مازاد ۳ ببرید.

حل: ابتدا ۲ و ۷ را جداگانه با ۳ جمع می‌کنیم و سپس آن‌ها را به کد BCD تبدیل می‌کنیم.

(۲۷)
↙ ↘
۵ ۱۰
↓ ↓
۸۴۲۱ ۸۴۲۱
۰۱۰۱ ۱۰۱۰
↘ ↙
(۰۱۰۱۱۰۱۰)

مثال: اگر کد افزونه ۳ برابر (۰۱۰۱۱۰۱۰) باشد، این کد معادل چه عددی در مبنای ده است؟

حل: برای تبدیل از کد بدون وزن مازاد ۳ به مبنای ده ابتدا بیت‌های کد را ۴ تا ۴ تا از سمت راست جدا می‌کنیم و با وزن ۱و۲ و۴و۸ از مبنای دو به ده می‌بریم. سپس از هر کدام به‌طور جداگانه سه واحد کم کرده و ارقام حاصل را بدون جابه‌جایی کنار هم می‌گذاریم.

۸۴۲۱ ۸۴۲۱
$\underbrace{۰۱۰۱}_{۵}\underbrace{۱۰۱۰}_{۱۰}$
$-۳\downarrow$ $-۳\downarrow$
$(۲\quad ۷)_{۱۰}$

مثالی از چندین روش کدگذاری در جدول زیر آمده است:

رقم دهدهی	۸۴۲۱	Excess − ۳	۸۴−۲−۱	۲۴۲۱	۵۰۴۳۲۱۰
۰	۰۰۰۰	۰۰۱۱	۰۰۰۰	۰۰۰۰	۰۱۰۰۰۰۱
۱	۰۰۰۱	۰۱۰۰	۰۱۱۱	۰۰۰۱	۰۱۰۰۰۱۰
۲	۰۰۱۰	۰۱۰۱	۰۱۱۰	۰۰۱۰	۰۱۰۰۱۰۰
۳	۰۰۱۱	۰۱۱۰	۰۱۰۱	۰۰۱۱	۰۱۰۱۰۰۰
۴	۰۱۰۰	۰۱۱۱	۰۱۰۰	۰۱۰۰	۰۱۱۰۰۰۰
۵	۰۱۰۱	۱۰۰۰	۱۰۱۱	۱۰۱۱	۱۰۰۰۰۰۱
۶	۰۱۱۰	۱۰۰۱	۱۰۱۰	۱۱۰۰	۱۰۰۰۰۱۰
۷	۱۱۱	۱۰۱۰	۱۰۰۱	۱۱۰۱	۱۰۰۰۱۰۰
۸	۱۰۰۰	۱۰۱۱	۱۰۰۰	۱۱۱۰	۱۰۰۱۰۰۰
۹	۱۰۰۱	۱۱۰۰	۱۱۱۱	۱۱۱۱	۱۰۱۰۰۰۰

به کد ۵۰۴۳۲۱۰ کد دو پنجی یا Biquinary نیز می‌گویند.

توجه: در وزن‌های منفی فقط کافی است علامت منفی وزن را منظور کنید. گاهی اوقات مشاهده می‌شود که وقتی یک رقم دهدهی را به کد BCD با وزن مثبت می‌بریم برای یک رقم دو کد ایجاد می‌شود. برای مثال اگر کد BCD وزن دار مثبت با وزن (۱و۲و۴و۲) باشد عدد دهدهی ۵ را به دو شکل زیر می‌توان نمایش داد.

$$\frac{۲۴۲۱}{۰۱۰۱}=۴+۱=(۵)_{۱۰} \qquad \frac{۲۴۲۱}{۱۰۱۱}=۲+۲+۱=(۵)_{۱۰}$$

هر دو نمایش صحیح می‌باشند اما یک قانون مهم می‌گوید که باید هر کد دارای نمایش منحصر به فرد باشد. بنابراین حتماً فقط یکی از آن‌ها را باید در نظر گرفت. بنابراین قرارداد پر کردن وزن‌ها به شکل زیر می‌باشد:
از صفرتا ۹ ده کد وجود دارد. آن‌ها را به دو قسمت تقسیم می‌کنیم. پنج کد اول یعنی از صفر تا ۴ از سمت راست به چپ پر می‌شود و پنج کد دوم یعنی از ۵ تا ۹ از سمت چپ به راست پر می‌شوند.
بنابراین در مثال فوق عدد ۵ باید از سمت چپ و به راست پر شود و داریم:

$$\frac{۲۴۲۱}{۱۰۱۱} = ۲+۲+۱ = (۵)_{۱۰}$$

و حالت دیگر غیر قابل قبول خواهد بود.

کد گری (Gray Code)

کدگری یک کد انعکاسی است، یعنی در این روش کدگذاری دو عدد متوالی تنها در یک بیت با هم اختلاف دارند. کدهای گری از صفر تا ۱۵ در زیر آمده است:

عدد	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵
کد گری	۰۰۰۰	۰۰۰۱	۰۰۱۱	۰۰۱۰	۰۱۱۰	۰۱۱۱	۰۱۰۱	۰۱۰۰	۱۱۰۰	۱۱۰۱	۱۱۱۱	۱۱۱۰	۱۰۱۰	۱۰۱۱	۱۰۰۱	۱۰۰۰

از آن‌جایی‌که در هر انتقال بین دو عدد متوالی در کدگری تنها یک تغییر رخ می‌دهد بنابراین خطای کمتری در انتقال رخ خواهد داد. یکی از کاربردهای کدگری، هنگامی است که داده آنالوگ به وسیله تغییر پیوسته شیفت نمایش داده شود. در این موارد می‌توان دوره شیفت را به قطعاتی تقسیم نمود و به هر قطعه عددی تخصیص داد. اگر قطعات مجاور به کدگری مرتبط شوند، ابهام در تفکیک دو ناحیه مجاور کاهش می‌یابد.

تبدیل اعداد دودویی به کدهای گری

از سمت راست هر دو بیت را با هم XOR می‌کنیم و حاصل را عیناً پایین می‌نویسیم. آخرین بیت کد را عیناً در پایین می‌نویسیم.

مثال: عدد $(۰۰۱۰)_{Bin}$ را به کد گری معادل تبدیل کنید.

حل:

۰ ۰ ۱ ۰

↓ ⊕ ⊕ ⊕

۰ ۰ ۱ ۱

راست به چپ ←

مثال: عدد دودویی $(۱۱۰۰۰۱۱۰۰۱۰۰۰۱)_۲$ را به کد گری تبدیل کنید.

حل: کدگری حاصل برابر است با:

$(۱۰۱۰۰۱۰۱۰۱۱۰۰۱)_{Gray}$

تبدیل کدهای گری به اعداد دودویی

اولین بیت از سمت چپ را عیناً در پایین می‌نویسیم. سپس همان بیت با رقم مجاور XOR می‌کنیم و حاصل در پایین نوشته می‌شود. این کار را تا آخرین بیت برای حاصل مرحله قبل و بیت بعدی ادامه می‌دهیم.

چپ به راست →

۰ ۰ ۱ ۱

⊕ ⊕ ⊕

$(۰\ ۰\ ۱\ ۰)_{NBCD}$

مثال: کدگری $(۱۱۰۰۱۱۱۱۱۱۰۰۰۰)_{Gray}$ را به دودویی تبدیل کنید.

حل:

$(۱۰۰۰۱۰۱۰۱۰۰۰۰۰)_۲$

کدهای تصحیح و تشخیص خطا (Error Detection Code & Correction Codes)

در انتقال اطلاعات دودویی از وسایلی نظیر سیم، امواج رادیویی و غیره استفاده می‌شود. هر نویز خارجی که وارد وسایل فیزیکی شود، ممکن است بیت‌های ۰ و ۱ را به هم تبدیل کند. در این بخش به بررسی کدهایی می‌پردازیم که در صورت رخداد خطا می‌توانند آن‌ها را تشخیص دهند و بعضی از آن‌ها حتی قدرت تصحیح این خطا را نیز دارا می‌باشند.

فرض کنید که I و J کلمات باینری n بیتی باشند. وزن هر کلمه برابر تعداد یک‌های آن کلمه می‌باشد. هم‌چنین فاصله دو کلمه I و J تعداد ارقامی است که با هم اختلاف دارند.

مثال: اگر $I = ۰۰۱۱۰۰۱۱$ و $J = ۱۱۱۰۰۱۱۱$ در این‌صورت وزن این دو عدد و فاصله آن‌ها از هم را به‌دست آورید.

حل:

$$W(I) = ۴$$
$$W(J) = ۶$$
$$dist(I,J) = ۴$$

فاصله مینیمم (d_{min}): به حداقل فاصله دو واژه در یک کد فاصله مینیمم آن کد گویند.

نکته مهم: اگر کدی t خطا را تصحیح و s خطا را تشخیص دهد، آن‌گاه s و t باید در نامساوی زیر صدق کنند:

$$۲t + s + ۱ \le d_{min}$$

مثال: اگر فاصله مینیمم در کدی برابر $d_{min} = ۲$ باشد، آن‌گاه این کد چه برخوردی با خطاها دارد؟

حل: با توجه به رابطه بیان شده در نکته قبل، این کد تنها می‌تواند یک خطا را تشخیص داد.

مثال: اگر فاصله مینیمم در کدی برابر $d_{min} = ۳$ باشد، آن‌گاه این کد چه برخوردی با خطاها دارد؟

حل: این کد می‌تواند یک خطا را تصحیح کند.

مثال: اگر فاصله مینیمم در کدی برابر $d_{min} = ۴$ باشد، آن‌گاه این کد چه برخوردی با خطاها دارد؟

حل: این کد می‌تواند یک خطا را تصحیح کند و یک خطا را تشخیص دهد. توجه داشته باشید که این دو خطا مستقل از هم می‌باشند.

بنابراین برای تشخیص و تصحیح خطاها باید از کدهایی با حداقل فاصله دو استفاده کنیم. برای تولید چنین کدهایی، می‌توان از بیت‌های توازن استفاده نمود، که به‌صورت زیر می‌باشند:

کد توازن فرد/زوج (Odd/Even Parity Code)

برای ایجاد کدهای توازن فرد یا زوج کافی است یک بیت توازن (P) را به هر یک از کلمات کد اضافه کنیم تا مجموع وزن کد فرد یا زوج شود. در این‌صورت d_{min} برابر ۲ خواهد شد.

برای تولید بیت توازن زوج کافی است همه بیت‌ها را با هم XOR کنیم و برای بررسی خطا کافی است همه بیت‌ها به همراه بیت توازن تولید شده را با هم XOR کنیم. مثلاً اگر کلمات یک کد دارای سه بیت x و y و z باشند برای تولید بیت توازن زوج از مدار زیر استفاده می‌کنیم: