$(2^n - N)$

قاعده کلی برای محاسبه متمم ۲ یک عدد: صفرها و اولین رقم غیر صفر از ۲ کم شوند. بقیه ارقام از ۲-۱ کم شوند

قاعده کلی برای محاسبه متمم ۲-۱ یک عدد: همه رقم‌ها از ۲-۱ کم شوند. $(2^n - N - 1)$

مثال:

$N = 01101100$

$$[N]_{2-1} = 2^8 - N - 1 = (100000000)_2 - (01101100)_2 - 1 = 10010011$$

$$\begin{array}{r} 100000000 \\ -\ 01101100 \\ \hline 10010100 \end{array} \qquad \begin{array}{r} 10010100 \\ -\ 00000001 \\ \hline 10010011 \end{array}$$

همه بیت‌ها نسبت به بیت‌های متناظر در عدد N، not شده است

مثال:

$$(50956)_{10} \Rightarrow [50956]_{10} = 49044$$

$$[50956]_{10-1} = 49043$$

در این درس در مورد محاسبات در سیستم متمم ۲-۱ بحث نمی‌کنیم زیرا محاسبات در سیستم متمم ۲ ساده‌تر است و در عمل از سیستم متمم ۲ (۲) استفاده می‌کنیم.

end_around carry

- کدهای عددی: منظور کدگذاری روشی برای نمایش اطلاعات است. به عنوان مثال سه رنگ قرمز، زرد و سبز برای کدگذاری مفاهیم go ، stop ، caution استفاده می‌شود. در این جا با کدهای عددی سر و کار داریم.

- یک کد n بیتی، 2^n ترکیب مختلف از صفر و یک‌ها را به وجود می‌آورد ⇐ یک کد n بیتی یک مجموعه 2^n عنصری را می‌تواند کد کند.

11
10
01
00

کد ۲ بیتی

کد BCD: Binary coded decimal (دهدهی کد شده به صورت باینری)

کد BCD اعداد تک رقمی با عدد باینری چهار بیتی معادل آنها برابر است

رقم	کد BCD
۰	0000
۱	0001
۲	0010
۳	0011
۴	0100
۵	0101
۶	0110
۷	0111
۸	1000
۹	1001

مثال: عدد ۳۵۲ را به صورت BCD کد نمایید.

$$(352)_{10} = (0011\ 0101\ 0010)_{BCD}$$

از کد BCD معمولا در نمایش اعداد در روی صفحه نمایش استفاده می‌شود و یا پردازنده‌هایی که محاسبات را در مبنای ۱۰ انجام می‌دهند.

یک عدد k رقمی دهدهی با 4k بیت به صورت BCD بیان می‌شود.

$$(10)_{10} = (1010)_2 = (0001\ 0000)_{BCD}$$

محاسبات در کدهای BCD:

$$\begin{array}{r} 2 \\ +\ 3 \\ \hline 5 \end{array} \qquad \begin{array}{r} 0010 \\ +\ 0011 \\ \hline 0101 \end{array}$$

$$\begin{array}{r} 4 \\ +\ 6 \\ \hline 10 \end{array} \qquad \begin{array}{r} 0100 \\ 0110 \\ \hline \underbrace{1010}_{10} \end{array} \xrightarrow{+6} \underbrace{00010000}_{16}$$

$$\begin{array}{r} 196 \\ +\ 578 \\ \hline 774 \end{array} \qquad \begin{array}{rrr} 1 & 1 & \\ 0001 & 1001 & 0110 \\ +\ 0101 & 0111 & 1000 \\ \hline 0111 & 10001 & 1110 \\ & 0110 & 0110 \\ \hline & 0111 & 0100 \\ \downarrow & \downarrow & \downarrow \\ 7 & 7 & 4 \end{array}$$

- برای جمع و تفریق اعداد علامت‌دار BCD معمولا از روش متمم ۱۰ استفاده می‌شود.
- علامت عدد مثل ارقام خود عدد با چهار بیت نمایش داده می‌شوند.

$$\begin{array}{r} 196 \\ -\ 578 \\ \hline -382 \end{array} \xrightarrow{\text{در سیستم متمم ۱۰}} \begin{array}{r} 0196 \\ +\ 9422 \\ \hline 9618 \end{array} \longrightarrow$$

حاصل یک عدد منفی است زیرا رقم آخر ۹ شده => متمم ۱۰ می‌گیریم تا مقدار عدد مشخص شود.

$$9618 \xrightarrow{10'c} 0382 \Rightarrow (9618)_{10'c} = (-382)_{10}$$

محاسبات بالا به صورت BCD :

$$\begin{array}{r} 0196 \\ 9422 \\ \hline 9618 \end{array} \qquad \begin{array}{rcccc} & & & 1 & \\ & 0000 & 0001 & 1001 & 0110 \\ + & 1001 & 0100 & 0010 & 0010 \\ \hline & 1001 & 0110 & 1011 & 1000 \\ & & & 0110 & \\ \hline & & & 0001 & \\ & \downarrow & \downarrow & \downarrow & \downarrow \\ & 9 & 6 & 1 & 8 \end{array}$$

- کد اسکی (ASCII) در بسیاری از کاربردها اطلاعات هم به صورت عدد هستند و هم به صورت حروف و کاراکترها
- کد اسکی معمولاً ۷ بیتی است => ۱۲۸ کد اسکی داریم
- اگر به صورت ۸ بیتی استفاده شود می‌توان علائم یونانی و ایتالیک را هم نمایش داد.

A 1000001 → 65
B 1000010 → 66
⋮
Z 1011011
[
]
\
/
=
a 1100001 → 97
b 1100010

کد گری (Gray): یک روش کد کردن اعداد است به طوری که کدهای متناظر با دو عدد همسایه، تنها در یک بیت با هم اختلاف داشته باشند.

کد گری n بیتی:

n=۱	n=۲	n=۳	n=۴
۰ → ۰	۰ → ۰۰	۰ → ۰۰۰	۰۰۰۰
۱ → ۱	۱ → ۰۱	۱ → ۰۰۱	۰۰۰۱
	۲ → ۱۱	۲ → ۰۱۱	۰۰۱۱
	۳ → ۱۰	۳ → ۰۱۰	۰۰۱۰
		۴ → ۱۱۰	۰۱۱۰
		۵ → ۱۱۱	۰۱۱۱
		۶ → ۱۰۱	۰۱۰۱
		۷ → ۱۰۰	۰۱۰۰
			۱۱۰۰
			۱۱۰۱
			۱۱۱۱
			۱۱۱۰
			۱۰۱۰
			۱۰۱۱
			۱۰۰۱
			۱۰۰۰

برای تولید کدهای گری با تعداد بیت دلخواه می‌توان به صورت بازگشتی عمل کرد.

یکی از موارد استفاده از کد گری، جدول کارنو است که در آینده خواهیم دید. این کد در خیلی جاهای دیگر مثل رمزگذاری نیز کاربرد دارد.

fixed point numbers. → در این اعداد محل ممیز ثابت است.

در این نوع اعداد هم می‌توان از روش های اندازه ـ علامت، متمم ۲ و یا متمم ۱-۲ برای نمایش اعداد علامت‌دار استفاده شود.

$۱۱۰/۰۱۰ \xrightarrow{\text{اندازه ـ علامت}} -(۲/۲۵)_{۱۰}$

$۱۱۰/۰۱۰ \xrightarrow{\text{متمم ۲}} -(۰۰۱/۱۱۰)_{۲} = -(۱/۷۵)_{۱۰}$

$۱۱۰/۰۱۰ \xrightarrow{\text{متمم ۱}} -(۰۰۱/۱۰۱)_{۲} = -(۱/۶۲۵)_{۱۰}$

Floating point number — اعداد ممیز شناور

$$N = M \times r^{E}$$

E → exponent — fixed point integer

M → Mantissa → fixed point number

این روش نمایش، مشابه روش نمایش علمی است

$$۸۵۲ = ۰/۸۵۲ \times ۱۰^{+۳} = ۸/۵۲ \times ۱۰^{۲}$$

$$N = \pm(a_{n-1} \cdots a_0 / a_{-1} \cdots a_{-m})_r = \pm(۰/a_{n-1} \cdots a_{-m})_r \cdot r^{n}$$

معمولا مانتیس به صورت اندازه علامت کد می‌شود.

$$\Rightarrow M = (s, a_{n-1} \cdots a_{-m})_{SM}$$

$$= (-1)^s (0.a_{n-1} \cdots a_{-m})_2$$

- مانتیس همیشه به گونه‌ای انتخاب می‌شود که $1 > |M| \geq 0.5$ باشد

(البته این موضوع ممکن است در استانداردهای مختلف متفاوت باشد)

$$N = M \times 2^E$$

$$N = (M/2)\, 2^{E+1}$$

$$N = (M2)\, 2^{E-1}$$

اگر برای exponent ، e بیت درنظر بگیریم:

$$-2^{e-1} \leq E \leq 2^{e-1} - 1$$

$$0 \leq \underbrace{E + 2^{e-1}}_{E_{new}} \leq 2^e - 1 \Rightarrow E_{new} = (b_{e-1} \cdots b_0)_{excess\_k}$$

$$k = 2^{e-1}$$

$$N = \boxed{s \quad b_{e-1} \cdots b_0 \quad a_{n-1} \cdots a_{-m}}$$

$$N = (-1)^s (0.a_{n-1} \cdots a_{-m})_2 \times 2^{(b_{e-1} \cdots b_0) - 2^{e-1}}$$

مثال 8 $n+m=7$ ، $e=5$ انتخاب شود. در این صورت عدد $(11.125)_{10}$ را به صورت ممیز شناور نمایش دهید

$$(1011.001)_2 = (\cancel{\ldots})_2 \times 2^4 \quad 0.1011001$$

$$\Rightarrow M = 0.1011001$$

$$E = (4)_{10} = (100)_2 \Rightarrow E_{new} = 4 + 16 = (20)_{10} = (10100)_2$$

$$\Rightarrow N = \underline{0}\,\underline{10100}\,\underline{1011001}$$

۱۹

IEEE standard

۱	bit	sign	۱	bit	sign
۸	"	E	۱۱	"	E
۲۳	"	M	۵۲	"	M
single precision			double precision		