

第15章 字节与十六进制

上一章中的两个改进的加法机清晰地解释了数据路径的概念。在整个电路中，8位值从一个部件传到另一个部件。它们是加法器、锁存器、选择器的输入，经过运算或操作又从这些部件输出。这些数由开关定义，最后由灯泡来表示结果。可以认为电路中的数据路径的宽度是8位。可是，为什么一定是8位，而不是6位、7位、9位或10位呢？

最简单的回答就是这些加法机是在第12章中最原始的加法机上改进而来的，而最原始的那个加法机就是8位。不过，这个解释似乎很缺乏说服力。实际上，用8位的原因是它表示一个字节。

字节这个词大概是在1956年前后由IBM公司最早提出来的。这个词起源于bite，但用y代替了i，以便不会被人误认为它是bit。曾经有一段时期，字节仅仅简单地表示特定数据路径上数据的位数。但是到了20世纪60年代中期，随着IBM的360系统的发展（一种大型复杂的商用计算机），字节这个词专门用来表示8位二进制数。

作为一个8位数，一个字节可以从00000000取值到11111111。这些数可以代表0~255的正数，也可以表示-128~127范围内的正、负数。总之，一个特定的字节可以代表 2^8 即256种不同事物中的一个。

8位数事实上是很适用的，字节在很多方面都比一位数优越。IBM采用字节的一个原因就是它们易于以BCD（将在第23章中描述）这种格式保存。巧的是，在以后的各章中你会看到字节用于保存文本也是很合适的，因为世界上大部分书面语言都可以用少于256个字符的字符集来表示（除了汉语、日语、韩语等所用的表意文字以外）。用字节表示黑白图像中的灰度也是很合适的，因为人眼大约能区分256种不同程度的灰度。当一个字节不足以表示信息时（如刚才说的表意语言：汉语、日语、韩语等），用两个字节，即 2^{16} 或65536也可以很好地表示。

半个字节，即4位二进制数，有时被称为半位元组，它比起字节而言，用的并不频繁。

由于字节在计算机内部经常出现，所以尽可能简单明了地表示它会带来很大方便。例如，一个8位二进制数10110110的确很直观但是不简明。

当然也可以用十进制数来表示字节，但这要求从二进制换算成十进制，这样做不仅不简单，反而是件很令人讨厌的事。第8章曾描述了一种很直观的方法。每一位二进制数字写到对应的方盒子中，并在其下方标上2的乘方数。把每一列相乘后再相加即可得到对应的十进制数。下面是10110110的转换：

1	0	1	1	0	1	1	0
x128	x64	x32	x16	x8	x4	x2	x1
128	+ 0	+ 32	+ 16	+ 0	+ 4	+ 2	+ 0
							= 182

把十进制数转换为二进制数就更麻烦了。你可以用十进制数去除以递减顺序排列的2的幂，每除一次，商是一个二进制位，而余数则继续去除以下一个最大的2的幂。下面是十进制数182转换成对应二进制数的过程：

182	54	54	22	6	6	2	0
+128	+64	+32	+16	+8	+4	+2	+1
1	0	1	1	0	1	1	0

第8章有关于这个方法的更多描述。总之，在十进制数和二进制数之间进行转换通常不是件十分简单的事。

从第8章中我们还学习了八进制数字系统，八进制数字系统只使用数字 0、1、2、3、4、5、6、7。在八进制数和二进制数之间进行转换却是小菜一碟，你只要记住每个八进制数字对应 3 位二进制数字即可，如下表所示：

二进制	八进制
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

如果已有一个二进制数字（如 10110110），则从最右边的数字开始，每 3 位二进制数字组成一组，即可转换为对应的八进制数：

$$\begin{array}{c} 10110110 \\ \underbrace{\quad\quad\quad}_2 \underbrace{\quad\quad}_6 \underbrace{\quad}_6 \end{array}$$

可见，字节 10110110 可以表示为八进制的 266。这显然已简单了很多，八进制确实是表示字节的一个好方法，但其中仍然有一个问题。

字节的二进制表示范围是从 00000000 ~ 11111111，对应的八进制表示范围是从 000 ~ 377。上例清楚地表示出 3 位二进制数对应于最右边和中间的八进制数，而 2 位二进制数对应于最左边的八进制数，这就表明一个 16 位二进制数的八进制表示和把它分成两个字节后的八进制表示有所不同：

$$\begin{array}{c} 1011001111000101 \\ \underbrace{\quad\quad}_1 \underbrace{\quad\quad}_3 \underbrace{\quad\quad}_1 \underbrace{\quad\quad}_7 \underbrace{\quad\quad}_0 \underbrace{\quad\quad}_5 \\ \\ \underbrace{10110011}_2 \underbrace{0011}_6 \underbrace{\quad}_3 \quad \underbrace{11000101}_3 \underbrace{\quad}_0 \underbrace{\quad}_5 \end{array}$$

为了使多字节值能和单字节值的表示一致，需要的系统应该能使每个字节平分，这意味着应该把每个字节分成 4 组，每组 2 位（以 4 为基数）；或 2 组，每组 4 位（以 16 为基数）。

让我们看看以 16 为基数的情况，这是我们还未接触过的新的记数系统。它被称为“十六进制（hexadecimal）”，这个词本身就让人迷惑，因为大部分以 hexa- 为前缀的词都是指与 6 有关的事物，而这里 hexadecimal 却是指 16。虽然微软公司在技术出版物的格式方面明确地声明不要将十六进制缩写为 hex，但绝大多数人还是使用这种缩写。

在十进制中，我们这样计数：

0 1 2 3 4 5 6 7 8 9 10 11 12

在八进制中，不需要8和9：

0 1 2 3 4 5 6 7 10 11 12

同样，以4为基数的系统不需要4、5、6或7：

0 1 2 3 10 11 12

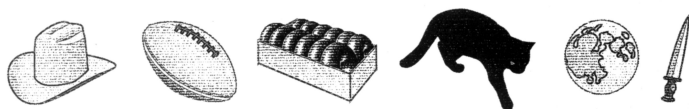
而二进制只需要0和1：

0 1 10 11 100

但是十六进制有些不同，它需要的符号比十进制还要多，如下所示：

0 1 2 3 4 5 6 7 8 9 ? ? ? ? ? 10 11 12

上图中10出现的地方是表示十进制中的数16。此外，还需要6个符号来表示十六进制数，但这些符号是什么呢？它们来自哪里呢？最形象的方法是引入6个新符号，例如：



它们不同于大部分数字用的符号，这些符号的优点是便于记忆，而且从某种意义上代表了它们应该表示的数字。你看，重10加仑的牛仔帽、一个橄榄球（一个橄榄球队有11人）、一打椰子、一只黑猫（和不吉利的13相联系）、一轮满月（一般出现在新月后的第14天晚上）以及让人们联想到Julius Caesar在三月的第15天被暗杀时所用的匕首。

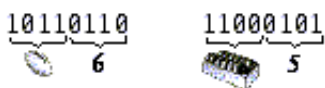
每个字节都可以用两个十六进制数表示，换句话说，一个十六进制数代表4位二进制数，即半个字节。下表描述了在二进制数、十六进制数和十进制数之间的转换：

二进制数	十六进制数	十进制数	二进制数	十六进制数	十进制数
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010		10
0011	3	3	1011		11
0100	4	4	1100		12
0101	5	5	1101		13
0110	6	6	1110		14
0111	7	7	1111		15

下图表示如何用十六进制表示字节10110110：

10110110
6

即使要表示多字节数也很容易：



一个字节总是由一对十六进制数来表示。

不过，我们绝不会真的用橄榄球或匕首来表示十六进制数，事实上，我们用 6 个拉丁字母来表示那 6 个十六进制数，如下所示：

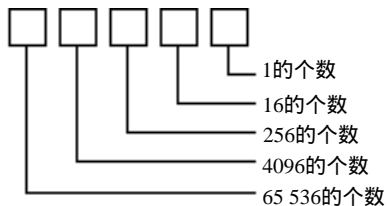
0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12

下表表示出在二进制数、十六进制数和十进制数之间的转换：

二进制数	十六进制数	十进制数
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

字节 10110110 可以由十六进制数 B6 来表示，而不用再画上一个橄榄球。同前面的章一样，用下标来表示记数系统的基数，如：10110110_{TWO} 表示二进制、2312_{FOUR} 表示四进制、266_{EIGHT} 表示八进制、182_{TEN} 表示十进制、B6_{SIXTEEN} 则表示十六进制。不过，这真是挺麻烦的。还好，有更简单、更通用的方法来表示十六进制：B6_{HEX} 还可以进一步简化为 B6h。

在十六进制数中，每一位的位置都对应于 16 的幂：



十六进制数 9A48Ch 是：

$$\begin{aligned}
 9A48Ch &= 9 \times 10000h + \\
 &\quad A \times 1000h + \\
 &\quad 4 \times 100h + \\
 &\quad 8 \times 10h + \\
 &\quad C \times 1h
 \end{aligned}$$

用16的乘方表示为：

$$\begin{aligned} 9A48Ch = & 9 \times 16^4 + \\ & A \times 16^3 + \\ & 4 \times 16^2 + \\ & 8 \times 16^1 + \\ & C \times 16^0 \end{aligned}$$

用对应的十进制数代入为：

$$\begin{aligned} 9A48Ch = & 9 \times 65536 + \\ & A \times 4096 + \\ & 4 \times 256 + \\ & 8 \times 16 + \\ & C \times 1 \end{aligned}$$

注意，在写一个数时，不用下标来表示数的基数也不会引起混淆。9就是9，不管它是十进制数还是十六进制数；而A则显然是个十六进制数，相当于十进制中的10。

把所有数字转换成十进制数需要下列运算：

$$\begin{aligned} 9A48Ch = & 9 \times 65536 + \\ & 10 \times 4096 + \\ & 4 \times 256 + \\ & 8 \times 16 + \\ & 12 \times 1 \end{aligned}$$

答案是631 948。以上是一个十六进制数如何转换成十进制数的过程。

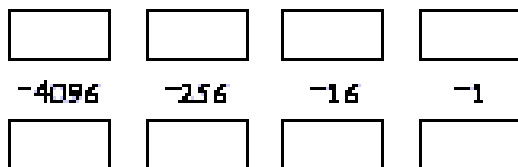
下面是把任何一个4位十六进制数转换成十进制数的模板：

x4096		x256		x16		x1
	+		+		+	
						=

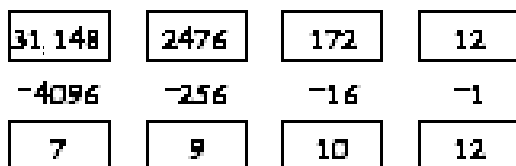
例如，这儿有一个79ACh的转换过程。记住，十六进制中的A和C对应于十进制中的10和12：

把十进制数转换为十六进制数需要做除法。如果数字比255小，则可以用1个字节来表示，对应于两个十六进制数。为了求出这两个数，用原数去除以16得到商和余数。举例说明，182除以16得11，余6，所以十六进制表示为B6h。

如果想要转换的十进制数比65 536小，则十六进制表示会有4位或更少。下面是把这样一个十进制数转换为十六进制数的一个模板：



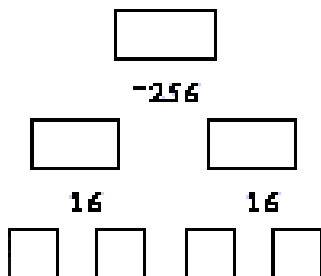
先把整个十进制数放到左上角的盒子里，作为第一个被除数，用它除以 4096（第一个除数），商放到被除数下面的盒子里，余数放到被除数右边的盒子里。余数成为新的被除数，用它除以256。以下是31 148如何转换成十六进制数的过程：



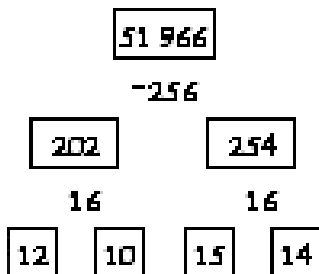
当然，十进制数的10和12用十六进制表示就是A和C，故结果是79ACh。

这个方法的问题是如果你想用一个计算器来做除法运算，它不会显示出余数是多少。如果你用31 148除以4096，计算器给出的结果只能是7.6044921875。为了计算余数你得用4096×7（得到28 672），再从31 148中减去它；或者用4096乘以0.6044921875，即商的小数部分。（不过，有些计算器具有十进制数和十六进制数之间的转换功能。）

把小于65 535的十进制数转换为十六进制数的另一个方法是先把原数除以 256而分为两个字节，对于每个字节，再除以16。下面是模板：



从最上面开始，每做完一次除法，商就进入除数左下方的盒子里，而余数进入右边的盒子里。例如，51 966的转换过程是：



得到的十六进制数字是12、10、15和14，即CAFE，它看起来倒更像一个单词而非一个数字！

下面是和十六进制相关的加法表：

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

可以用这张表和通常的进位规则来对十六进制数做加法：

$$\begin{array}{r}
 4A3378E2 \\
 + 877AB982 \\
 \hline
 D1AE3264
 \end{array}$$

曾在第13章中讲过可以用2的补数来表示负数。如果在二进制中处理8位带符号数，则所有的负数都是以1开头的。在十六进制中，两位带符号数若是以8、9、A、B、C、D、E或F开头则是负数。例如，99h可能表示十进制的153（如果处理的是1个字节的无符号数）或十进制的-103（如果处理的是有符号数）。

字节99h也有可能就是十进制的99，关于这一点会在第23章中解释，但是下一步必须先讲讲存储器。