

14

反馈与触发器 / 160

想象一下,如果你没有了记忆力,该如何去数数?我们不记得刚刚数过的数,当然也就无法确定下一个数是什么!同理,一个能计数的电路必定需要触发器。本章要介绍的就是各种触发器。

15

字节与十六进制 / 186

在前面的章节中,加法器、锁存器以及数据选择器的输入和输出形式都是8位的数据流,也即数据路径的位宽为8,为什么要定义为8位呢?为什么不是6位、7位、9位或10位?本章就要解释其中的缘由。

16

存储器组织 / 197

每天清晨,我们将自己从沉睡中唤醒,这时大脑的空白会很快被记忆填满。我们立刻会意识到自己身在何方,最近做了些什么事情,有什么计划打算。有的事情我们很快就能想起来,但有时并非如此。我们可以借助许多工具来记录信息,比如笔和纸、磁带,当然现在还可以使用存储器。

17

自动操作 / 215

人类的本性中带有一些懒惰的特质。我们总是抵触繁重的工作,对枯燥的、重复性的工作深恶痛绝。所以,当你必须用前面搭建的加法器计算100个数,甚至更多个数的加法时,有一种念头就会不可遏制地从脑子里冒出来:怎样让加法器自动地完成数据输入和计算呢?办法肯定是有的,那就是编写程序。

18

从算盘到芯片 / 252

算盘、滑尺、纳皮尔骨架、差分机、解析机、继电器、电子管、晶体管、芯片、计算机;甘特、帕斯卡、莱布尼兹、杰奎德、巴贝芝、图灵、冯·诺依曼、香农;IBM、贝尔实验室……你觉得应接不暇了吗?把这些你或熟悉或生疏的名词和名字串起来,就是人类的计算工具发展史。让时光倒流,去看看那些精巧的工具,感受天才们的巧思吧!

19

两种典型的微处理器 / 276

将中央处理器的所有构成组件封装到一块硅芯片上，就得到了微处理器。第一片微处理器芯片诞生于1971年，即Intel 4004系列，其中集成了2300个晶体管，你或许觉得可笑——如今家用计算机的微处理器上所安置的晶体管数量已经以亿为计量单位了。但是，从本质上来说，微处理器实际所做的工作并没有变。在本章，我们就来看看两种有着辉煌历史的典型微处理器：Intel 8080 和 Motorola 6800。

20

ASCII 码和字符转换 / 307

计算机中的存储器唯一可以存储的形式是比特，因此如果想在计算机上处理信息，就必须把它们转换为比特的形式来存储。我们已经掌握了如何用比特来表示数字和机器码。如何用它来存储文本呢？毕竟，人类所积累的大部分信息，都以各种文本形式保存的。下面就轮到 ASCII 码出场了！

21

总线 / 325

一台计算机包括很多部件：中央处理器、存储器、输入/输出设备等。通常这些部件按照功能被分别安装在两个或更多的电路板上。这些电路板之间通过总线（bus）通信。如果对总线做一个简单的概括，可以认为总线就是数字信号的集合，而这些信号被提供给计算机上的每块电路板。

22

操作系统 / 346

你或许梦想过自己组装一台近乎完整的计算机，像老木偶匠盖比特雕刻木偶匹诺曹一样，全部亲自动手用小零件完成。不过在你的机器能完成你想要的操作之前，还差一个重要的东西——操作系统！

23

定点数和浮点数 / 365

整数、分数以及百分数等各种类型的数字与我们形影不离，它几乎出现在我们生活的所有角落。例如你加班 2.75 小时，而公司按正常工作时间的 1.5 支付你工资，你用这些钱买了半盒鸡蛋并交了 8.25% 的销售税。在计算机的

内存里，所有的数都表示为二进制形式。通过前面的学习，我们知道 2 用二进制可以表示为 10_2 ，可是 2.75 用二进制怎样表示呢？这就是本章的主题。

24

高级语言与低级语言 / 381

第 22 章介绍了如何编写一段简单的程序，让我们可以利用键盘将十六进制机器码输入计算机，以及通过视频显示设备来检查这些代码。但是使用机器码编写程序就如同用牙签吃东西，伸出手臂费半天劲刺向食物，但每次都只取到小小的一块，用这种低级语言编写程序既费力又费时，有悖于我们发明计算机的初衷。不过，人们想出了一种效率更高的编程方法——使用高级语言。

25

图形化革命 / 398

回顾历史，从第一台继电器计算器到现在为止，六七十年过去了，计算机的处理速度飞速增长。不过要充分利用计算机日益增长的运算和处理能力，就必须不断改进计算机系统中的用户接口（User Interface），因为它是人机交互的轴心。图形化革命来了！



1

至亲密友

你今年 10 岁，你最好的朋友就住在街对过。事实上，你们各自卧室的窗户正好彼此相对。每当夜幕降临，父母就如同往常一样，早早地催促你该上床睡觉了，但是你和你的朋友还想交流想法，交换见闻，分享各自的秘密，或者扯扯闲话，开开玩笑，聊聊梦想。这本无可厚非。无论怎样，渴望交流本来就是人类最主要的天性之一。

当卧室里的灯依然亮着的时候，你可以和朋友互相挥手，使用各种手势或简单的肢体语言，来表达一两个想法。但是，要表达复杂的想法可能就比较困难了。而且一旦父母宣布“关灯”，这种交流似乎也不可能继续下去。

如何交流呢？或许可以打电话？10 岁小孩的房间里有电话吗？即使有，无论电话在哪里，你们的谈话都有可能被偷听。如果你家里的电脑连接了电话线，它可能帮上忙，而且不会发出声响，但是——等等，电脑也不会你的房间里。

你和朋友所采用的方法就是使用手电筒。众所周知，手电筒是为了让孩子们能够躲在被子下看书而发明的；在天黑后用手电筒来交流信息似乎也是理想的选择。它们当然是很安静的，并且光线是高度定向的，同时光线也不会渗漏出卧室而引起家人的疑心。

手电筒能用来交谈吗？这当然值得一试。我们在一年级的時候学习怎样在纸上写字

母和单词，因此，把同样的方法运用到手电筒上似乎也是有道理的。只需要站在窗户边，用光来画出字母。对于字母“O”，打开手电筒，在空气中划一个圈，然后关掉手电筒。对于字母“I”，竖着划一下。但是，你很快就会发现，这个方法也不太行得通。当你看着朋友的手电筒在空中圈圈点点时，会发现很难在头脑中组合出那么多复杂的笔画。这些旋转和倾斜的光线都太不准确了。

或许大家都曾经在电影里看到这样的情节，两个水手在海上通过灯的亮灭来互相发送信号。而在另一部电影里，一个间谍转动一面镜子将太阳光反射到另一个房间里，从而向被俘的同伙传递信息。或许那正是解决问题的办法。如此一来，你就可以发明一种简单的技术。在这个方案里，字符表里的每个字符对应一连串的手电筒闪烁。“A”是闪一次，“B”是闪两次，“C”是闪三次，依此类推，“Z”就是闪 26 次。单词 BAD 可以用闪 2 次，闪 1 次，闪 4 次这样的组合来表示，而且在字符之间设置的小停顿使这个单词不至于被误认为是闪 7 次的字母“G”。另外，单词之间停顿可以稍长些。

这似乎很有希望，采用这种方案的好处是你不必在空中比划手电筒了，只要对准方向和按开关就行了。但是这种方案也有一个不足，那就是如果你想发送的第一个消息是“How are you?”，那么你将总计需要让手电筒闪 131 次！而且，这还是忽略了标点符号的，你还没有设计闪多少次来对应一个问号。

但是这已经离答案更近一些了。我们能够肯定的是，在此之前一定有人也遇到过类似的问题，而你解决这个问题的思路也是非常正确的。等到了白天，跑一趟图书馆，查查资料，你发现了一个被称为莫尔斯电码（Morse Code）的伟大发明。这正是你想找的，尽管你现在必须重新学习如何去“写”字母表里的字母。

它们的不同之处在于：在你发明的系统里，字母表里的每个字母就是用一定数目的闪光表示的，闪 1 下为“A”，闪 26 下为“Z”。而在莫尔斯电码里，则有两种闪烁——短闪和长闪。当然这使得莫尔斯电码更加复杂，但是在实际应用里它被证明是更为有效的。句子“How are you?”现在只要闪 32 下（包含一些短闪和一些长闪），而不再是 131 下，而且这其中还包括了一个代表问号的编码。

当问及莫尔斯电码是如何工作的时候，人们并不会谈论“短闪”与“长闪”。相反，他们使用“点（dot）”和“划（dash）”，因为这是在打印纸上显示编码的一个便利方法。在莫尔斯电码里，字母表里的每个字母都与一个点划序列相对应，正如下表所示。

A	· —	J	· — — — —	S	· · ·
B	— — — · · ·	K	— — — · — —	T	— —
C	— — — · — — — ·	L	· — — — · ·	U	· · · — —
D	— — — · ·	M	— — — — —	V	· · · · — —
E	·	N	— — —	W	· — — — —
F	· · — — —	O	— — — — — —	X	— — — · — — —
G	— — — — ·	P	· — — — — —	Y	— — — — — —
H	· · · ·	Q	— — — — — — —	Z	— — — — — ·
I	· ·	R	· — — —		

虽然莫尔斯电码和计算机毫无关系，但是，熟悉编码的本质对于深入理解计算机软硬件内部结构以及隐匿在其后的语言将大有裨益。

在这本书里，编码这个词的意思是指一种用来在机器和人之间传递信息的方式。换句话说，编码就是交流。有时候我们认为编码就是指秘密的东西（密码）。但是大部分编码不是这样的。毕竟，大部分编码必须易于理解，因为它们是人类交流的基础。

在《百年孤独》这本书的开篇里，加西亚·马尔克斯回忆了一个时代，那时“这个世界刚刚出现，以至于很多东西缺乏命名，这时就有必要亲自用手指明这些事物”。我们赋予这些东西名字时往往是很随意的。这就好比说为什么猫不被叫做“狗”而狗不被叫做“猫”一样，没有什么理由可言。你可以说英语词汇就是一类编码。

对任何能听见我们的声音并理解我们所说的语言的人来说，我们发出的声音所形成的词语是一种可识别的编码。我们将这个编码称为“口头话语（the spoken word）”或“言辞（speech）”。对于写在纸上（或刻在石头上、木头上，或者在空气中比划）的词，我们还有其他的编码方式。这种编码以手写字符或打印在报纸、杂志以及书本上的字符形式出现。我们叫它“书面语言（the written word）”或“文本（text）”。在许多语言里，语言和文字之间存在着很紧密的联系。例如，英语中的字母和字母组合与它们的发音（或多或少）存在一定的对应性。

对于那些丧失听说能力的的聋哑人而言，人们发明了另一种编码来帮助他们进行面对面的交流。这就是手语。手语通过手和臂膀形成的动作和姿势来传达词语中的单个字符或者整个词语，以及基本的概念。对于那些失明的人，书面语言可以用布莱叶盲文（Braille）来替代。这种文字使用一系列凸起的点来代表字母、字母串以及整个单词。当

话语必须快速抄成文本时，缩写和速记都是很有用的。

我们使用各种不同的编码来为我们自己的交流服务，因为有些编码有时比其他编码更便捷。例如，话语的编码不能存储在纸上，因此，书写的编码就被用来替代话语的编码。如果在黑暗的环境下，而且交流的双方之间有一定的距离，那么通过讲话或者文字来进行秘密的信息交换几乎是不可能的，因此，莫尔斯编码就成了一个方便的选择。如果一种编码可以用在其他编码无法取代的地方，那么它就是一种有用的编码。

我们将会看见，各种类型的编码也用在计算机里来存储和传递数字、声音、音乐、图片和电影。计算机不能直接处理人类的编码，因为计算机无法通过与人类的眼睛、耳朵、嘴巴和手指完全相同的方式来接收人类发出的信息。然而，计算机技术的一个最新趋势，已使得我们的个人计算机能够获取、存储、处理和呈现一切用于与人类沟通的信息，无论视觉信息（文字和图片），还是听觉信息（口语、声音和音乐），或两者的相结合（动画和电影）。所有这些类型的信息都需要它们各自的编码，就像人类说话需要一套器官（嘴和耳朵）而写作和阅读需要另一套（手和眼）一样。

甚至前面所列的莫尔斯编码表，其本身就是一种类型的编码。在这个表中，每个字母由一系列的“点”和“划”来表示。然而实际上我们不能发送“点”和“划”，相反，我们发送与“点”和“划”对应的闪烁光。

当使用手电筒发送莫尔斯编码时，迅速地打开和关闭开关代表一个“点”（快闪），让闪光时间保持得相对长一些代表“划”（慢速闪光）。例如在发送字母 A 时，首先以非常快的速度打开并关闭手电，然后再以稍慢的速度进行一次。在发送下一个字符前，需要暂停一会。在此约定，一个“划”的时长是“点”的 3 倍。例如，如果一个“点”的时长是 1 秒钟，那么一个“划”的时长就应当是 3 秒钟。（在现实中，莫尔斯编码的传输速度远比这要快得多）。接收者看到一个短促的闪烁和一个拖长的闪光后，就知道这是一个 A 了。

在莫尔斯编码中，“点”和“划”之间的停顿是至关重要的。例如，当发送一个字母 A 时，在发送的“点”和“划”之间，手电筒要关闭一段时间，这相当于一个点的闪烁时长（如果“点”的时长是 1 秒钟，那么“点”和“划”之间的停顿也应该是 1 秒钟）。对于同一个单词中的字母，则通过较长的停顿来分隔，这大约相当于一个“划”的时长（或者说是 3 秒钟，如果一个“划”的时长就是 3 秒钟的话）。下图以“hello”的莫尔斯编码

为例，说明字母之间的停顿（间隙）。

● ● ● ● ● ● ■■■■ ● ● ● ■■■■ ● ● ■■■■ ■■■■ ■■■■

单词之间则通过相当于两个“划”的时长来分隔（如果一个“划”的时长为3秒钟，那么这个停顿就应当为6秒钟）。例如下图是“hi there”所对应的编码。

● ● ● ● ● ● ■■■■ ● ● ● ● ● ● ■■■■ ● ●

手电筒开和关的时长并没有限定。它们都与一个“点”的时长相关，而这依赖于手电筒开关能以多快的速度扳动，以及莫尔斯编码的发送者能以多快的速度记下一个指定字符。对于一个快速发送者，他的“划”的时长，也许只相当于一个慢速发送者的“点”的时长。这会带来一个小问题，即阅读莫尔斯编码信息会变得很困难，但是经过一两个字母之后，接收者通常就可以分清楚哪一个是“点”，哪一个是“划”了。

乍一看，莫尔斯编码的定义与打字机字母的排列一样都是随意的。这里的定义是指字母表中的字母与各种“点”和“划”的组合序列之间的对应关系。然而，仔细研究之后，我们就会发现事实并非完全如此。简单且短促的编码，被分配给字母表中使用频率较高的字母，例如E和T。拼字比赛选手和“幸运轮”的粉丝们可能会很快意识到这一点。而不太常用的字母，比如Q和Z（这只会让选手在拼字中得到10分）则被分配以较长的编码。

几乎每个人都知道一点儿莫尔斯编码。三个点、三个划，再加三个点就表示SOS，即国际求救信号。SOS不是一个缩写，这只是一个易于记忆的莫尔斯编码序列。在第二次世界大战期间，英国广播公司一些无线电广播以贝多芬第五交响曲的片段来作为节目的前奏——BAH, BAH, BAH, BAHMMMMM，而贝多芬绝不会想到他所谱写的这段乐曲可以表示为莫尔斯编码的V，即胜利（Victory）之意。

莫尔斯编码的一个缺点是，它没有区分大写字母和小写字母。但是除了可以代表字母外，莫尔斯编码还使用一组由点和划组成的五元序列来表示数字，如下所示。

1	• • • • •	6	• • • • •
2	• • • • •	7	• • • • •
3	• • • • •	8	• • • • •
4	• • • • •	9	• • • • •
5	• • • • •	0	• • • • •

这些编码至少比字母编码更有规律一些。大部分标点符号由 5 个、6 个或者 7 个“点”和“划”的组合序列来表示，如下所示。

.	• • • • •	!	• • • • •
,	• • • • •	(• • • • •
?	• • • • •)	• • • • •
:	• • • • •	=	• • • • •
;	• • • • •	+	• • • • •
-	• • • • •	\$	• • • • •
/	• • • • •	¶	• • • • •
!"	• • • • •	-	• • • • •

还有一些编码则用来表示某些欧洲语言中的重音字母，以及用于特殊目的的速记序列。如 SOS 的编码就是这样的—一个速记序列：必须连续地传递这三个字母，而且字母之间只有一个点的停顿时间。

如果有一个专门为此设计的手电筒的话，你会发现和朋友之间通过莫尔斯编码交流将变得更加简单。除了一个可以滑动的开关外，这种手电筒还有一个按钮开关用来控制灯光的明灭。在一些实际应用中，可以达到每分钟传递 5 到 6 个单词的速度——虽然这仍比讲话慢得多（讲话的速度在每分钟 100 个单词左右），但这已经足够了。

当你和朋友最终熟记了莫尔斯编码之后（这是能熟练地收发编码的唯一方法），你们甚至可以在口语中使用它，用来取代正常的语言。为了使编码发送的速度最快，你可以把“点”读作“嘀（dib）”，把“划”读作“嗒（dab）”。文字也可以用同样的方式简化成“点”和“划”的序列，莫尔斯编码的口语版把讲话内容缩减到只剩下两个声音了。

问题的关键就在于数字 2。两种闪烁，两种声音。事实上，两个不同的事物，只要经过适当的组合，就可以表示所有类型的信息，这的确是千真万确的。

2

编码与组合

莫尔斯码 (Morse Code) 是由塞缪尔·莫尔斯发明的 (1791-1872)，在本书的其他章节中我们还将频繁地提到他。莫尔斯码其实是伴随着电报机的问世而被发明的，关于电报机，我们在后面也将做详细的探讨。正如通过研究莫尔斯码我们可以很方便地理解编码的本质一样，通过电报机来了解计算机硬件也是个不错的途径。

大多数人都会发现莫尔斯码的发送比接收更为简单。即使你并没有熟记莫尔斯码，也可以很方便地使用下面这张按字母表顺序排列的表格。

A	·—	J	·— ·— ·—	S	...
B	— ...	K	— ·— —	T	—
C	— ·— ·—	L	·— ...	U	··—
D	— ·—	M	— —	V	...—
E	·	N	— ·	W	·— —
F	··— ·—	O	— — —	X	— ·— —
G	— — ·—	P	·— — ·—	Y	— ·— — —
H	Q	— ·— ·—	Z	— — ...
I	··	R	·— ·—		

比起发送莫尔斯码，接收编码并进行解码要费时费力得多，因为译码者不得不根据

一串由“点”、“划”组成的晦涩的编码序列来反查字母。例如，如果你接收到一串形如“划-点-划-划”的编码，那么你就必须从表的第一个字母开始逐个搜寻，直到找到与这串编码相符的字母“Y”为止。

问题就出在这里，因为我们现在只有一张提供“字母 → 莫尔斯码”的编码表，而缺少一张可以实现反向查询的“莫尔斯码 → 字母”译码表。在开始学习莫尔斯码的初级阶段，如果有这样的一个表无疑将是很方便的。但是要建立这样一张表，谈何容易。似乎这些字母对应的“点-划”序列并没有什么规律。

所以忘掉字母序列吧。或许根据编码中所包含点、划的多少来对其进行分组，是一个更好的组织这些编码的方法。例如，一个仅包含一个点或一个划的莫尔斯码只能代表两个字母：“E”或“T”。

一组含有 2 个点或划的编码组合，可以给我们呈现出 4 个字母——I, A, N 和 M。

•	E	..	I	---	N
-	T	.-	A	--	M

一组含有 3 个点或划的莫尔斯码可以为我们表示更多的字母。

...	S	---..	D
..---	U	------	K
...--	R	---..-	G
...---	W	------	O

最后（如果我们不想考虑存在数字和标点符号的莫尔斯码的情况），一串由 4 个点或划组成的莫尔斯码就可以表示 16 个字符。

....	H	---...	B
...---	V	---...	X
...--.	F	---..-	C
...---	Ü	------	Y
...---	L	---...	Z
...---	Ä	------	Q
...---	P	---..-	Ö
...---	J	------	Ş

综合以上数据来看，这四张表包含了 $2+4+8+16$ 组码字，总共表示了 30 个字母，比拉丁字母表的 26 个字母还要多出 4 个。所以，你会注意到最后一个表中有 4 组编码是用来表示重音字母的。

当有人给你发送莫尔斯码的时候，上述四张表可能会让你的解码工作变得轻松很多。当接收到某一代表特定字母的码字后，你就可以知道其中所包含的“点”和“划”的数目，那么你至少可以很快找到对应的表格去进行查找。每个表格都组织得很规整，全部是“点”的码字被排在左上角，而全部是“划”的则被排在右下角。

你发现这四张表格在大小上的规律了么？注意看，每个表格所包含的码字数目都是前一张的两倍。这其实很好理解：每个表格所含有的码字，可以看成是在前一张表格所包含的全部码字上再加一个“点”，或者再加一个“划”而组成的新码字。

我们可以用如下这样一个列表来总结这个有趣的规律。

点和划的数目	码字的数目
1	2
2	4
3	8
4	16

在这四个表中，每张表的码字数都是前一张表码字数量的两倍，因此如果第一张表含有 2 个码字，那么第二张表则含有 2×2 个码字，而第三个表就有 $2 \times 2 \times 2$ 个码字。下面用另一种方式呈现这个规律。

点和划的数目	码字的数目
1	2
2	2×2
3	$2 \times 2 \times 2$
4	$2 \times 2 \times 2 \times 2$

当然，如果我们遇到了数字的自乘，就可以通过幂的方式来表示它。例如， $2 \times 2 \times 2 \times 2$ 可以记作 2^4 （2 的 4 次幂）。数字 2、4、8 和 16 都是以 2 为底数的幂值，因为你可以通过使其自身乘 2 来得到它们。由此我们的总结列表也可以写成下面这种样子。

点和划的数目	码字的数目
1	2^1
2	2^2
3	2^3
4	2^4

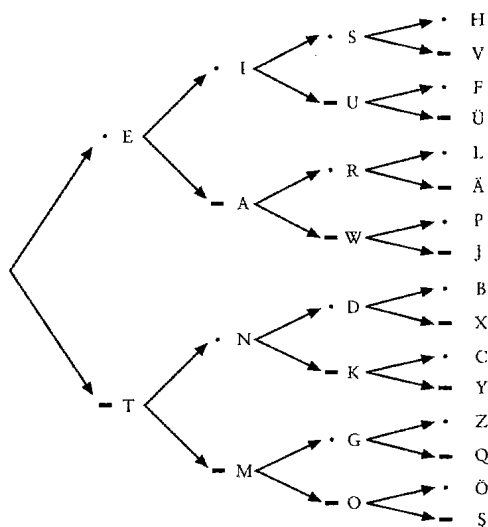
现在这个表已经变得很简洁了。如果知道了码字中“点”和“划”的数目，那么以这个数目为指数的 2 的幂运算结果就是其总共可以表示的码字数。我们可以用下面这个简单的公式来概括上述表格所表示的内容：

$$\text{码字的数目} = 2^{\text{“点”和“划”的数目}}$$

使用 2 的幂值的形式可以表示很多码字，在下一章中，我们还将接触另外一个例子。

为了让莫尔斯码的解码过程更加简单，或许画张图会有所帮助，例如下面这张树型图。

这张图给出了所有字母及其所对应的由“点”和“划”组成的连续序列。当对一串码字进行解码时，我们需要沿着箭头从左向右进行搜寻。以“点-划-点”的码字为例来说，当你需要找出这串码字所代表的字母时，应首先从图的左边开始，选择“点”的分支；然后继续沿着箭头向右选择“划”，接着又是一个“点”。找到最后一个“点”时结果就会紧随其后出现了，没错就是字母“R”。



如果仔细想一想，你就会发现构建这样一个表对于定义莫尔斯码规范来说是很必要

的。首先，它确保了我们不会对不同的字母定义相同的码字。其次，通过这个表我们可以用尽可能短的码字来表示所有的字母，而避免产生编码长度上的浪费。

我们可以继续加长码字至 5 位或者更长，不过这可能超出页面打印边界。一串由 5 个“点”或“划”组成的编码串可以为我们提供 32 ($2 \times 2 \times 2 \times 2 \times 2$, 或 2^5) 种扩展的码字。对于莫尔斯码中定义的 10 个数字和 16 个标点符号来说，通常这已经足够了，而实际上数字确实就是使用 5 位的莫尔斯码来表示的。但是在很多其他编码方式中，5 位码字常用来表示重音字母而不是标点符号。

为了把所有的标点符号也都包含进去，编码系统必须要扩展到 6 位了！扩展后将为我们提供 64 ($2 \times 2 \times 2 \times 2 \times 2 \times 2$, 即 2^6) 种新增的码字，这样总共的码字就达到了 $2 + 4 + 8 + 16 + 32 + 64$ ，也就是 126 种！这对莫尔斯码来说有点太多了，甚至还留下了很多“未定义”的码字。这里“未定义”用来表示那些不代表任何字符的码字。如果你在接收莫尔斯码的时候收到了一个未定义码字，可以肯定发送方一定是出了差错。

我们很容易就能得到这样一个小公式：

$$\text{码字的数目} = 2^{\text{编码的位数}}$$

利用它就可以继续计算出更长位数的点划序列所能表示的码字数目了。

点划的数目	码字的数目
1	$2^1 = 2$
2	$2^2 = 4$
3	$2^3 = 8$
4	$2^4 = 16$
5	$2^5 = 32$
6	$2^6 = 64$
7	$2^7 = 128$
8	$2^8 = 256$
9	$2^9 = 512$
10	$2^{10} = 1024$

幸运的是，我们并不需要写出所有可能的码字来计算码字的总数目。我们需要做的只是让 2 不断地与自己相乘。

莫尔斯码也被称作二进制码 (Binary Code)，因为这种编码的组成元素只有两个——“点”和“划”。这跟硬币有些类似，因为硬币落到地上只能是正面朝上或反面朝上。二元对象（例如硬币）和二进制码（例如莫尔斯码）常常使用 2 的乘方来进行描述。

上面所做的关于二进制编码的分析工作，其实是数学的一个分支，称作“组合学”或“组合分析”，而我们所作的分析则只能说是一个简单的练习。传统意义上来说，因为组合分析涉及类似像扔硬币、掷骰子这样的需要对其组合数目进行推算的问题，所以它经常被应用到概率和统计学中。但是它对于我们理解码字的组合与分解也是十分有帮助的。