**第 15 次课 学时 2**

|  |  |
| --- | --- |
| **授课章节内容** | 第八章 有噪信道编码 |
|  | 8.6 循环码 |
| **教学目标** | 教学目标5 |
| **支撑毕业要求** | 毕业要求2-1 |

**教学要求：**

1. 知识目标

* 掌握循环码的概念，生成多项式，编译码实现电路。

1. 能力目标

* 具备利用生成多项式设计循环码的能力。

1. 素质目标

* 激发学生对电子信息相关专业课程的学习热情，通过对循环码的学习，提升运用理论指导实践的工程素质。

**教学重点与难点**：

循环码生成多项式，编译码实现电路

**教学过程设计：**

课前预习，观看视频8.6，导入新课，强调循环码是线性分组码的重要子类，只是增加一个循环特性而已

| **讲授与指导内容** | | **讲课、互动内容设计** |
| --- | --- | --- |
| 8.6 循环码  二进制循环码是一种重要的线性分组码，这种码可以通过简单的反馈移位寄存器来实现，伴随式的计算也可以简单地通过相似的反馈移位寄存器实现。循环码所具有的代数结构使它具有元组是子空间*S*的一个码字，那么，经由循环移位得到的也同样是*S*中的一个码字；或者，一般来说，经过次循环移位得到的也是*S*中的一个码字。  码字的各个分量可以看做是多项式的系数，即  （8.48）  可以将多项式函数看做是码字*U*中的数字的“占位符”，即将一个*n*元矢量用*n*-1阶或更低阶的多项式来描述。每一项的存在或不存在对应了*nn*元组中相应的位置为1或0。如果非0，那么多项式的阶为*n*-1.在讨论循环码的代数结构时，将会看到把码字描述为多项式的作用。  8.6.1循环码的代数结构  把一个码表示成多项式的形式后，码的循环特性将以如下方式表示出来。如果是一个（*n*-1）阶的码字多项式，那么，，即除以得到的余项同样是一个码字，即  （8.49*a*）  或者，两边同乘以，  （8.49*b*）  同样可以将它描述成取模的算术形式，即  模 （8.50）  其中，模定义为除以后得到的余式。下面对的情形验证式（8.50）的有效性：      现在加减；或者，由于使用的是模2算术运算，将加两次，    因为是*n*-1阶的，它不能被除，这样，根据式（8.49*a*），有  模  经过扩展，得到式（8.50），即  模  【**例8.26**】 码矢量的循环移位  对，设，把码字表示成多项式的形式，并且根据式（8.50），求出循环移位3比特后得到的码字。  解：  （多项式按阶数从低到高排列）  其中  将除以，用多项式除法求出余式：    将余式写成低阶到高阶的形式：，码字是*U*=1101循环移位3位的结果。注意，对于二进制码，加法操作是模2运算，所以+1=-1，因此计算中没有出现任何负号。  8.6.2二进制循环码的特性  类似于用生成矩阵生成分组码，也可以用生成多项式（generator polynomial）生成循环码。对于一个循环码，生成多项式是唯一的，具有如下形式：  （8.51）  其中，和都必须等于1。子空间中的任何一个码字多项式都满足，其中，是一个阶或更低阶的多项式。消息多项式写为  （8.52）  共有个码字多项式，在一个码中有个码矢量。因为对应于每一个码矢量都必须有一个码字多项式，因此  或  因此，正如式（8.51）显示的那样，没有余项时，*U*是子空间*S*中一个有效的码字。  一个循环码的生成多项式是的一个因式；即，例如：    用作为阶数为的一个生成多项式，可以生成一个的循环码。或者当时，使用，可以生成循环码。总之，如果是阶的多项式并且是的一个因子，那么唯一地生成一个循环码。  8.6.3 系统形式的编码  根据循环码的代数性质来建立系统编码的过程，可以把消息矢量用如下多项式表示：  （8.53）  在系统中，消息数字作为码字的一部分。我们可以把消息比特移入码字寄存器的最右边*k*个，而把监督比特加在最左边的个中。这样，对消息多项式进行代数变换，使得它右移比特。如果将乘以，则得到右移的消息多项式  （8.54）  如果再将式（8.54）除以，结果可以表示成  （8.55）  其中，余式可以表示成    同样，可以说  模 （8.56）  在式（8.55）两边同时加上并进行模2运算，可得  （8.57）  式（8.57）的左边是有效的码字多项式，因为它是阶或更低阶的多项式，当它除以时，余式为0，码字可以扩展为它的多项式形式，如下所示：    码多项式对应于码矢量：  （8.58）  【**例8.27**】系统形式的循环码  使用生成多项式，从（7，4）码字集中为消息矢量***m***=1011生成系统码字。  解： ，，，    将用多项式除法除以，可以写为    根据式（8.57），可得      8.6.4 多项式除法电路  如前所述，码字多项式的循环移位和消息多项式的编码需要将一个多项式除以另外一个多项式，这种操作可以由除法电路（反馈移位寄存器）实现。考虑两个多项式和：    和  对于，图8.10的除法电路实现了多项式除以的除法运算，并求出了商和余项：    寄存器的各级首先初始化为0。第一个移入的最高有效系数。在次移位后，商输出是，这是商的最高项。对于商的每一个系数，都要从被除式中减去。图8.10中的反馈连接实现了这个减法。在每次移位时，计算被除式中剩下的最左边项和反馈项之差，并将此作为寄存器中的内容。寄存器每移位一次，差就向前移动一位；阶数最高的项（刚开始时是0）被移出，而中下一个有意义的系数被移进。在总共次进入寄存器的移位之后，商串行输出而余式留在寄存器中。    图8.10 多项式除法电路  【**例8.28**】 除法电路  使用图8.10所示的除法电路将除以，求出商和余项。比较多项式除法的电路实现和手工实现。  解：除法电路需要实现以下操作：    依据图8.10的一般形式，所需要的反馈移位寄存器如图8.11所示。假定寄存器的内容初始化为0，那么电路实现以下操作步骤：   |  |  |  |  | | --- | --- | --- | --- | | 输入序列 | 移位次数 | 寄存器内容 | 输出和反馈 | | 0001011 | 0 | 000 | - | | 000101 | 1 | 100 | 0 | | 00010 | 2 | 110 | 0 | | 0001 | 3 | 011 | 0 | | 000 | 4 | 011 | 1 | | 00 | 5 | 111 | 1 | | 0 | 6 | 101 | 1 | | - | 7 | 100 | 1 |   4次移位后，商的系数串行出现在输出端，是1111，或者商多项式，余式系数是100，或者余数多项式是。总之，电路计算的结果为    多项式除法步骤如下：  移位后的输出，      图8.11 例8.28的除法电路  8.6.5 级移位寄存器的系统编码  循环码的系统编码，需要计算模得到的监督比特，换句话说，需要将信息多项式上移（右移）后与生成多项式做除法。上移是为了给监督比特腾出空间，这些监督比特附加于信息比特后，产生了系统形式的码矢量。将信息比特上移个位置是很普通的操作，实际上并不是由除法电路完成的。事实上，仅仅计算了监督比特，并适当地将其置于信息比特旁边的位置上。监督多项式是除以生成多项式后的余项（remainder），它可以通过图8.11所示的级反馈寄存器进行*n*次移位后得到。注意，寄存器中最初的次移位只是为了填满寄存器。只有当最右端的一级寄存器也被填上时才可能产生反馈。因此，如图8.12所示，可以将输入数据加入到最后一级寄存器的输出端以缩短移位的次数。另外，进入最左端寄存器的反馈项是输入端和最右端寄存器内容之和。要生成这个和，必须确保对任意生成多项式，都有。反馈电路按照如下生成多项式的系数进行连接：  （8.59）  下面描述了使用图8.12中编码器进行编码的步骤：   1. 开关1在前*k*次移位时闭合，允许将消息比特传输到级编码移位寄存器； 2. 开关2处于下方以允许在前*k*次移位时消息比特直接传送到输出端； 3. 传完*k*个消息比特后，开关1、开关2移到上面的位置； 4. 剩余的次移位通过将监督比特传送到输出寄存器，清空编码寄存器； 5. 总的移位次数等于*n*，寄存器的输出内容是码字多项式     图8.12 级移位寄存器编码  【**例8.29**】 循环码的系统编码  使用图8.12所示形式的反馈移位寄存器，将消息矢量***m***=1011编码成（7,4）码，生成多项式为。  解：          模  对于如图8.13所示的级编码移位寄存器，操作步骤如下：   |  |  |  |  | | --- | --- | --- | --- | | 输入序列 | 移位次数 | 寄存器内容 | 输出 | | 1011 | 0 | 000 | - | | 101 | 1 | 110 | 1 | | 10 | 2 | 101 | 1 | | 1 | 3 | 100 | 0 | | - | 4 | 100 | 1 |   在4次移位之后，开关1打开，开关2移到上面的位置，在寄存器中获得的监督比特移位到输出端。输出的码字为*U*=1001011，码字多项式为。    图8.13 用级移位寄存器进行（7，4）循环码编码例子  8.6.6 级移位寄存器检错  码字在传输中可能受到噪声干扰。因此，接收到的矢量有可能并不是发送的矢量。假设传输的码字具有多项式表达形式，而接收到的矢量其多项式表征为，因为是一个码字多项式，它必是生成多项式与某个多项式的乘积，即  （8.60）  而是受到干扰后的形式，可以写为  （8.61）  其中，是错误图样的多样式。译码器检验是否为一个码字多项式，即它除以后，余项是否为0。这是通过计算接收多项式的伴随式（syndrome）实现的。伴随式等于除以的余项，即  （8.62）  其中，是一个或更低阶的多项式。这样，伴随式是一个元组。由式（8.60）到式（8.62），可得  （8.63）  比较式（8.62）和式（8.63），可以发现伴随式作为模的余式，恰好等于模的余式。这样接收矢量多项式的伴随式就包含了纠正错误图样的信息。伴随式的计算是通过除法电路实现的，几乎与发送器所使用的编码电路相同。图8.14是一个用级移位寄存器计算伴随式的例子，这里码字矢量的生成同例8.29。开关1开始是闭合的，而开关2打开。开始时所有存储单元初始化为0，接收矢量移入寄存器的输入端。当整个接收矢量进入移位寄存器后，寄存器的内容就是伴随式。然后，打开开关1，合上开关2，这样就把伴随式移除寄存器。译码器的操作步骤如下：   |  |  |  | | --- | --- | --- | | 输入序列 | 移位次数 | 寄存器内容 | | 1001011 | 0 | 000 | | 100101 | 1 | 100 | | 10010 | 2 | 110 | | 1001 | 3 | 011 | | 100 | 4 | 011 | | 10 | 5 | 111 | | 1 | 6 | 101 | | - | 7 | 伴随式 |     图8.14 级移位寄存器计算伴随式的例子  如果伴随式是全0矢量，就认为接收到的矢量是正确的码字。如果伴随式是非0矢量，则接收的矢量就受到了干扰，即检测到了错误的发生；这些错误可以通过将错误矢量（由伴随式指明）加于接收矢量得到纠正。这种译码方法对简单的码字是十分有效的。更复杂的码字则需要使用代数技巧来获得实用的译码器。 | | 循环码理论分析完善，编译码电路易于实现  循环性是循环码的本质特征，要求吃透原理  除法电路用于产生监督位 |
| **小结** | 循环码特性，生成多项式，循环编译码的物理实现 | |
| **复习要点** | 循环码生成多项式，循环码编译码电路 | |
| **思考题** | 循环码生成多项式具备哪些特征？ | |
| **作业题** | 8-10 | |

作者签名：

****