

2013

# 《软件理论基础（1）》大作业

MVB 设备控制器的 Matlab/Simulink 仿真 之 信号接收单元



2011013236 陈华榕

2011013251 吕婉琪

清华大学 软件学院

2013-6-24

# 目录

---

- 1 背景概述 ..... 2
  - 1.1 数据包传输的工作原理 ..... 2
  - 1.2 数据头与数据尾 ..... 2
  - 1.3 CRC 校验：原理..... 3
  - 1.4 CRC 校验：作用..... 3
  - 1.5 数据的串行传输 ..... 3
  - 1.6 数据的并行传输 ..... 4
- 2 设计架构 ..... 4
  - 2.1 信号发送单元 ..... 4
  - 2.2 并行模型 ..... 4
  - 2.3 串并行混合模型 ..... 5
- 3 结果分析 ..... 6
- 4 结论 ..... 7
- 5 未来工作与心得体会 ..... 7
- 6 致谢 ..... 8

# 1 背景概述

---

MVB 的含义是多功能车辆总线，在总线上挂载了多个设备，设备之间通过总线进行通信，以传递信息。设备之间的通信遵循一定的协议，在数据链路层、物理层的二进制数据被包装成帧发送。数据被分成多组，每组 2 位，只取 01 与 10 两种可能的组合来表示逻辑 0 与逻辑 1。

## 1.1 数据包传输的工作原理

我们做的是信号接收单元，首先考虑数据帧是如何产生的。

待发送数据为 64 位，CRC 校验码 R 位，将这 64+R 位的逻辑 01 串按 0→01，1→10 的规则转化成 128+2R 位的二进制串，再在该串前加“101100011100010101”，串后加 0011 组成最后的发送数据，也就是以“0011”作为结束标志。而 128+2R 位的二进制串以两位为单位，一定不会出现“00”和“11”的片段，所以可以准确判断串的结束位置。

接收单元需要做的就是读取到“101100011100010101”后转入接收状态，依次读取相关数据，在读到“0011”时停止，这样应该可以接收到 128+2R 位的二进制串，将其转化成逻辑 01 串，最后再将数据和 CRC 校验串的部分分离出来进行校验。不符合 CRC 校验结果的将给出报错。

## 1.2 数据头与数据尾

为什么要有数据头与数据尾？

我们知道数据发送与接收都是连续进行的，因此发送方与接收方需要约定一个数据包开始和结束的信号，这样做至少有以下好处：

- 1) 保证接收到的数据帧与发送方发送的相同，不出现增、漏的情况。
- 2) 在非数据帧传输期间保证不会误将噪声信号作为数据帧接收。
- 3) 同一个接收方接收来自不同发送方的数据时，可用于辨别发送方。

因此，给数据帧增加数据头与数据尾是必要的。

为了区分开数据头尾与数据帧内容，就将数据帧通过 0→01，1→10 的规则转化成双倍的长度，并在数据头尾中增加“00”和“11”段。这样虽然会使数据帧大小翻倍，但考虑到以上好处，以及单个数据帧通常都不大的因素，增加数据头尾是非常值得的。

### 1.3 CRC 校验：原理<sup>1</sup>

CRC 校验的根本思想就是在要发送的帧后面附加一个数，生成一个新帧发送给接收方。当然，这个附加的数不是随意的，它要使所生成的新帧能与发送端和接收端共同选定的某个特定数整除（所用的除法是“模 2 除法”）。这样，到达接收端后，把接收到的数据帧除以（同样是“模 2 除法”）这个选定的除数，验证结果没有余数即可。如果有余数，就说明该帧在传输过程中出现了差错。

要达到以上的效果，可以采用这样的方法：

- 1) 选择一个用于在接收端进行校验时，对接收的帧进行除法运算的除数（通常以多项式表示这个除数，称为“生成多项式”），要求该除数的二进制最高位与最低位必须均为 1。
- 2) 设除数的二进制位数为  $k$ ，就在要发送的  $m$  位数据帧后加  $k-1$  位 0，对这个  $m+k-1$  位新帧通过模 2 除法得到余数，该余数就是得到的 CRC 校验码，直接将  $m+k-1$  位的新帧加上该校验码就得到了 CRC 校验后的完整发送数据帧。

### 1.4 CRC 校验：作用

实际上 CRC 校验也是发送、接收双方进行辨别的一种方法，也可以检验接收数据帧的正确性。

此外，由于选择不同的除数能得到不同的校验结果，因此 CRC 校验对提高数据帧传输的安全性也起到一定作用。

在提交的模型中，为了能自定义 CRC 校验的除数，我们选择了一般性的 CRC 校验模型，设置的除数是 10001000000100001，对应  $k=17$ ，CRC 校验码为 16 位，新帧为 80 位，再通过 0->01，1->10 转化并添加数据头尾，每个数据帧就是 182 位。

### 1.5 数据的串行传输

数据的串行传输是最直观的，因为本质来说，信号的传输就是以串行的方式进行的。也正是因为信号传输是串行的，而不是固定维度并行的传输，才需要进行上述 0->01，1->10 转化、添加数据头尾、CRC 校验等步骤来保证数据传输的鲁棒性和安全性。

---

<sup>1</sup> CRC 校验的原理：<http://winda.blog.51cto.com/55153/1063951>

## 1.6 数据的并行传输

对于数据的并行传输，若只考虑传输部分，将极大简化传输流程。由于每次传输的都是多维信号，因此每次采样都能直接得到一个完整的数据帧。不过，数据的完全并行传输在无线信号传输中是很难实现的，无线信号往往还是串行传输。

## 2 设计架构

---

由于通过并行传输的方式能保持数据帧的完整性，我们就首先设计编写了最简单的使用并行传输方式的信号接收单元。考虑到并行方式并不是实际中用到的数据传输方式，我们希望下一步在该并行模型的基础上修改得到串行模型，因此在并行模型中保持了对数据帧 CRC 校验、翻译、添加数据头尾等的支持。

接着，我们在上述并行模型的基础上，将信号传输转化为串行的方式进行。主要是编写了串行信号转并行信号的模块与并行信号转串行信号的模块。

在我们最终完成的信号接收单元中，收到的信号会先通过一个 StateFlow 的 Chart 提取出我们需要的 80 位数据（含 CRC 校验部分），接着通过串行信号转并行信号的模块将 80 位串行数据转化为 1 个 80 维的并行数据，将该 80 维数据传输给 CRC 综合校验模块，得到反 CRC 校验的结果，这就是我们需要的数据。

为了模拟真实环境以完成对我们搭建模型的测试，我们还设计编写了简易的信号发送单元，其中能够将数据封装成符合要求的数据帧并通过并行信号转串行信号的模块将每个数据帧转化成 182 位串行信号。

### 2.1 信号发送单元

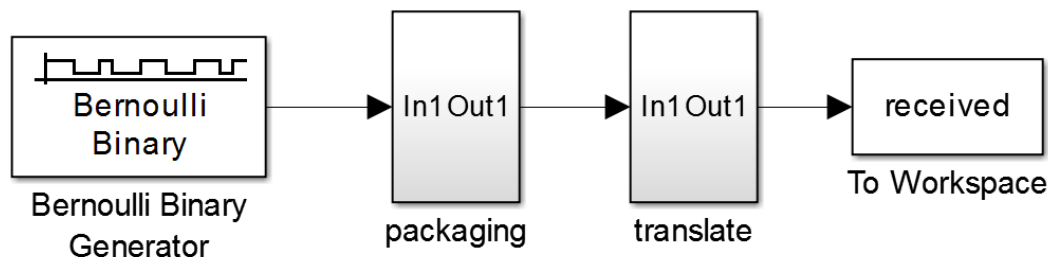
我们通过伯努利二进制序列生成器模块（Bernoulli Binary Generator）生成我们需要传输的 64 维/帧的数据，以串行方式通过一般性的 CRC 校验模块（General CRC Generator）根据指定的除数计算 CRC 校验码并附加到数据帧中得到 80 维/帧的并行数据，再通过自定义 Matlab Fcn 模块，对数据帧进行翻译与添加数据头尾的包装。

若需要发送串行信号，则辅以脉冲信号触发，将并行数据帧输出为连续 182 位串行数据。

### 2.2 并行模型

并行模型模拟的不是真实环境，而是为后续串行模型的完成作铺垫。在这里，我们保持了对数据帧 CRC 校验、翻译、添加数据头尾等的支持。

模型见 [slx/parallelreceiver.slx](#)。

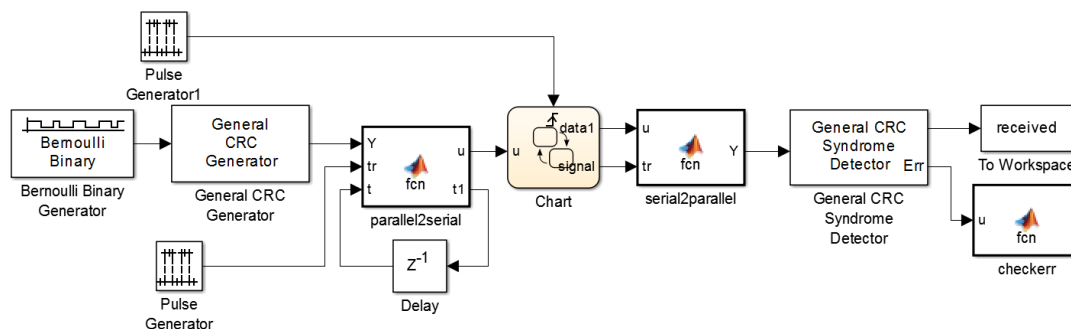


如图所示，待发送数据由伯努利二进制序列生成器产生，以 64 维并行输入 **packaging**，其中通过 CRC 校验得到 80 维并行数据，再通过封装最终输出 182 维并行数据。

假定数据以并行方式发送，当 **translate** 接收到 182 维并行数据后，通过逆封装得到 80 维含 CRC 校验的数据，再通过逆 CRC 校验得到 64 维数据，这正是一开始产生的传输数据，依照要求将其输出至 **workspace**。

## 2.3 串并行混合模型

模型见 [slx/mixreceiver.slx](#)。在上述并行模型基础上，将信号传输转化为串行的方式进行。即在发送方将信号转化为串行方式进行发送，在接收方以串行方式接收。



如图所示，CRC 校验及其之前的模块与并行模型相同，CRC 校验后我们能得到 80 维的并行数据，将不断输入给 **parallel2serial** 模块。**Parallel2serial** 模块保持输出为 0 的状态，当受到 **tr** 触发时立即记录此时的输入值，将其包装为 182 维的数据帧，并在接下来 182 个采样周期中以串行方式逐维输出整个数据帧。

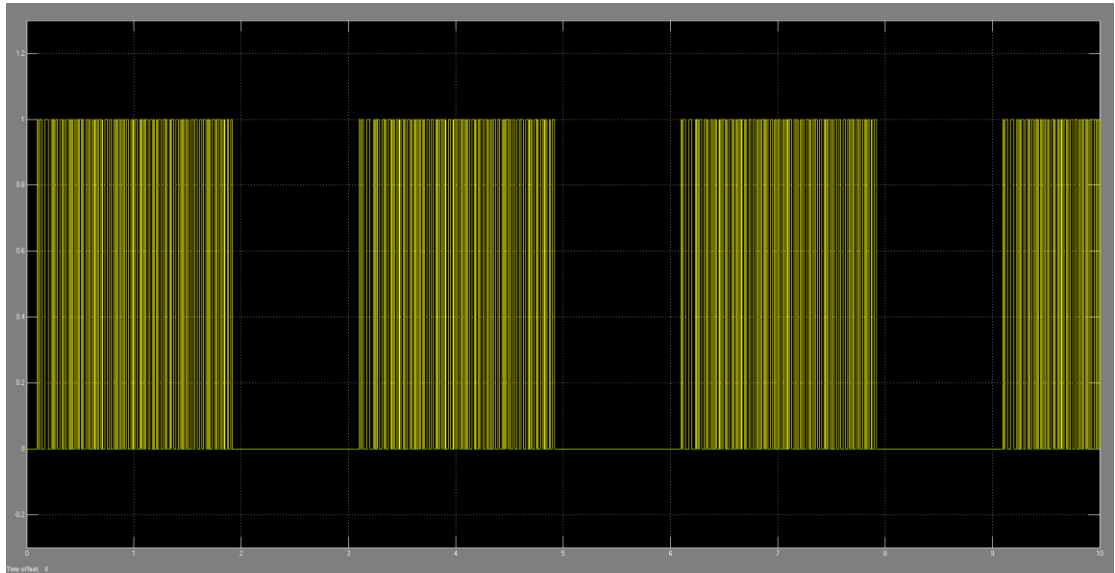
接着通过有限状态机处理接收到的信号。当接收到信号时，信号接收单元从默认的等待接收状态进入校验数据头的中间态。对数据头的校验采用的是用 **stateflow** 中 **chart** 实现的有限状态机。当数据头通过验证后，将转入接收状态，同时进行数据的翻译和检验是否结束。对数据的翻译和对数据尾也采用了有限状态机的实现方式。其具体原理为，当输入为 0->1 时，离开 1 状态时发出 **data1=0** 的信号，即将 01 翻译为 0；同理，当输入为 1->0 时，离开 0 的状态时发出 **data1=1** 的信号，将 10 翻译为 1。循环翻译，直至出现数据尾。若出现其他不合法的数据输入，则跳转到 **error** 状态，将直接进入等待下一组数据的等待接收状态。

接收到数据帧后转化为 80 维数据并从 data1 输出口串行输出，在完整接收到一个数据帧后提供一个采样周期的 signal 信号以触发后续的 serial2parallel 模块。

在 serial2parallel 模块中维护一个 81 维的队列，每次接收到 data1 就将队列中最早的数据出队并将 data1 入队。在收到 signal 触发时，由于数据尾的存在，实际在队列中混入了最新的一个 data1，而该 data1 不属于数据帧，因此将队列中除去最新的 data1 以外的数据按入队顺序依次组成一个 80 维向量，并行输出给逆 CRC 校验模块，校验后得到的 64 维数据正是一开始产生的传输数据，依照要求将其输出至 workspace。

### 3 结果分析

我们对完成的上述两个模型进行了大量的实验测试，总结起来，模型在正确性、鲁棒性、安全性等方面都有不错的表现。现在举一个实际实验测试的例子。



如上图所示，信号发送方发送了这样的串行数据。考虑第一个数据帧，在 1.91s 时，serial2parallel 输出了：10110111000110110000110101000111010001000010110010110010100011010101011101101110。

经检验，0101011101101110 确实是数据 101101110001101100001101010001110100100001011001011001010001101 的以 10001000000100001 为除数的 CRC 校验码，且输出至 workspace 的 received 在 1.91s 记录的确实是这个数据。

同样可验证其他的数据帧传输也是正确的。此外，我们还尝试了在数据帧传输间隔中添加一些噪声，确实不会对接收到的数据帧产生影响。而在数据帧传输时添加的噪声则有比较大的可能使数据帧不符合要求而被废弃。不仅如此，我们还尝试修改了添加 CRC 校验码后的数据帧，导致了模型通过 checkerr 模块在 workspace 中输出“CRC Verification Failed!”。

以上实验结果都是符合我们期望的。

## 4 结论

---

根据串并行的特点与实际情况，我们正确地选择了以串行方式进行信号的发送与接收，以并行方式进行发送或接收单元内的 CRC 校验与逆校验，并通过并行数据与串行数据的互相转换完成了衔接。从实际效果来看，这样的设计方法是经得起仿真环境考验的，在实际测试中，我们搭建的模型表现出了让人满意的性能。

这次的大作业通过实际动手的方式加深了我们对数据传输原理的理解与认识。我们设计编写了一个涵盖完整信号发送、接收体系的模型，测试过程中验证了添加数据头尾对噪声的过滤作用以及 CRC 校验在排除干扰、增强安全方面起到的作用。

我们认识到，理论的推演与实际环境常常会有较大差别，通过计算机仿真的方法是将理论结果应用于实际的第一大步。

我们对本次大作业中搭建的模型有信心，虽然可能其中存在一些我们没有注意到的问题，但我们仍期望并相信它能在经过一些修正后，被应用到实际无线硬件设备的信号接收过程中。

## 5 未来工作与心得体会

---

在我们设计编写的混合模型中，验证数据头的有限状态机的实现采用的是比较繁琐的实现方式，涉及到 17 个中间态，各状态间的各种转换也很多，虽然逻辑比较简单但是实现起来非常复杂，并且难以调试。今后可以考虑利用 Chart 中的 local 变量，将正确的数据头存储为一个向量，然后按位比对转换状态，其实现类似于基于有限状态机的字符串匹配算法。这样做的逻辑会比较复杂且不直观，但具体实现将变得简单，状态模型看起来也更为简洁，同时便于调试。

此外，由于实际芯片存在传输延迟时间，可能会影响模型的性能，因此可能需要由此进行一些调整。而且，串并行数据互相转换的效率有进一步提高的空间，整体的模型也有优化的可能。

通过这次大作业，我们对 Matlab/Simulink 有了更奇妙的认识。我们再次感受到 Matlab/Simulink 的强大，正是它们让我们能以比较方便的方式完成了本次大作业中的模型搭建，使我们能将更多的注意力放在模型的性能优化上。而借着这次作业，我们也学习了信号传输的皮毛知识，激发了我们的兴趣，对未来相关课程的学习应该会有帮助。

在作业完成的过程中，我们也多次遇到棘手的问题，对问题的理解也可谓是一波三折，这反映了我们的知识、经验都不够丰富，未来需要更加努力！



## 6 致谢

---

老师一学期以来为这门课付出了辛苦的汗水。作为新开的课，老师为了能让我们理解课堂内容、掌握相关知识，付出了很多额外的时间。而我们通过一学期三十次课的学习，不仅开阔了眼界，还树立了系统建模的意识，学到了很多模型参数辨识的方法等。更重要的是，计算机仿真的思想与方法已经根植于我们的脑海中，这在我们未来的求学、工作，甚至生活中一定会有所帮助。

两位助教对我们学习相关知识的巨大帮助，还耐心解答了我们的每一个问题，细心批改了我们的每一次作业，为此付出了很多时间。两位助教对我们这次大作业的完成也提供了非常重要的帮助，我们对问题的正确理解很大程度上要归功于助教的耐心。不仅如此。两位助教在本学期还教会了我们一些很巧妙或很奇特的方法，让人大开眼界，启发了我们的思考。

在此特别感谢老师与两位助教在这个学期的辛苦付出，感谢你们给予我们的莫大帮助，祝未来一切顺利！