《通信算法验证应用》

[《通信算法验证应用》 1](#_Toc183430068)

[第一节：通信 3](#_Toc183430069)

[1.1信息表示 4](#_Toc183430070)

[1.1.1通信格式 4](#_Toc183430071)

[1.1.2信号编码 4](#_Toc183430072)

[1.2通信要素与主流通信 5](#_Toc183430073)

[1.2.1通信三要素 5](#_Toc183430074)

[1.2.2主流通信 6](#_Toc183430075)

[1.3扩频通信 6](#_Toc183430076)

[1.3.1扩频通信 7](#_Toc183430077)

[1.3.2 m序列 9](#_Toc183430078)

[1.3.2.1本原始多项式 9](#_Toc183430079)

[1.3.2.2m序列发生器电路 10](#_Toc183430080)

[1.3.2.3m的特性 12](#_Toc183430081)

[第二节：扩频通信算法验证 13](#_Toc183430082)

[2.1扩频调制编码设计 13](#_Toc183430083)

[2.1.1扩频编码结构设计 15](#_Toc183430084)

[2.1.1.1待调制序列发生器电路实现 15](#_Toc183430085)

[2.1.1.2载波序列发生器电路实现 16](#_Toc183430086)

[2.1.1.3调制编码电路实现 17](#_Toc183430087)

[2.2扩频解码 18](#_Toc183430088)

[2.2.1扩频解码架构设计 18](#_Toc183430089)

[2.2.2序列恢复程序实现 19](#_Toc183430090)

[2.2.2.1程序实现 20](#_Toc183430091)

[2.2.3信息恢复程序设计实现 21](#_Toc183430092)

[2.2.3.1汉明检纠错 21](#_Toc183430093)

[2.2.3.2信息恢复程序实现 22](#_Toc183430094)

[2.3扩频抗干扰性验证 23](#_Toc183430095)

[2.3.1高斯噪声 25](#_Toc183430096)

# 第一节：通信

通信是数字电路设计中主要应用领域，简单的说就是在不同电路系统或者设备之间通过电路交换信息的过程。

通信从设计角度看，主要包括2个方面：

信息：有价值的东西，在电子设计中就是有价值（对我们生活有用）的数据，可以这样说只要是电路设计中的数据皆为信息。

交换：通过某种方式不同的电路系统或者设备之间交互价值，交换的方式可以理解为以下2种：

数字通信：将有价值的数据转话位数字信号，通过数字通信接口或者总线，传输信息

载体通信：将有价值的数据以某种波为载体，通过载体传播信息，常见的载体有

声音 ：比如广播

光 ：比如光纤

电信号 ：比如电缆

射频

驱动电路

解密

数字调制

载波调制

射频

驱动电路

载波调制

数字调制

扩频

加密

信息

介质

信息

解扩

在数字电路设计中只要我们涉及到以上应用，我们相对来说参与通信设计

## 1.1信息表示

载体通信

数字通信

信宿

信源

在通信中，信息在传递的时候有一定的格式，比如数字通信USB接口以包作为数据格式

在通信中，往往需要将原始数据编码为编码信息，在进行传输

### 1.1.1通信格式

在一些复杂的通信中，信息一般以帧或者包作为数据结构传递，简单的通信可以以字节为单位传递信息，在数字通信中有的情况可以以比特传递信息。

### 1.1.2信号编码

在一些通信中，往往需要将原始数据进行编码，按照编码的**用途**可以分为信道编码和信源编码。

信源编码：将原始数据进行压缩，提高通信效率，比如H264编码，JPEG编码，

信道编码：将原始数据添加多余的比特，为了提高通信的可靠性，信道编码按照编码方式分为如下2种：

卷积码

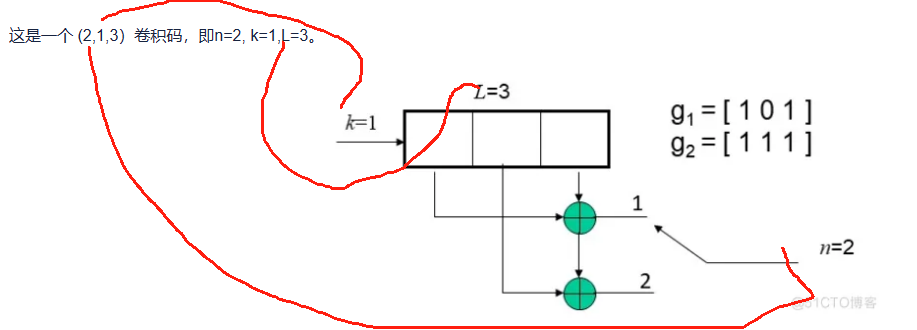
由于它的编码方式，编码不但与当前信息有关，还与之前的编码有关，它的表示为

Code=（N,K,L）

N:编码总长度

K：每个码中信息长度

L:记忆深度



分组码

由于它的编码方式，编码不但与当前信息有关，与之前的编码无关，它的表示为

Code=（N,K）

N:编码总长度

K：每个码中信息长度

## 1.2通信要素与主流通信

### 1.2.1通信三要素

在通信中参与通信的元素，即参与通信物理个体，概括起来如下3种

信源 ：发起信息的电路系统或者设备

信宿 ：信息的目的段的电路系统或者设备

传播介质 ：载体运动的介质，可以是无形或者有形

### 1.2.2主流通信

1. 卫星通信

通信在传播的时候，需要中继，当距离较远的的时候，地面中继建设难度大，以靠非地面中继，以卫星作为中继实现全球通信。

1. 光纤通信

在常见载体中，光是传播最快的，在信源和信宿端加光电转换电路即可提高传输速率

1. 扩频通信

在通信抗干扰解决方案中，扩频通信能够很好的满足抗干扰的要求

## 1.3扩频通信

在通信中抗干扰是影响通信的重要因素，如何能够更好的解决通信抗干扰问题就在通信应用中比较重要。

香农很早提出了香农定理，它很好的阐述了关于通信抗干扰性的规律，定理如下：



Rmax（通信能力） :通信过程中所能达到的最大数据传输量

W（信号自身的一个参数） ：带宽，指的是信号能量集中的频谱宽度

P：db

Wl

Wh

F：hz

**W = wh-wl**

**W:是由信号本身决定**

**信号频率越高，带宽越宽**

S：信号功率

N：噪声功率

S/N：信噪比，值越大信号质量越好，当S/N<1意味信号倍被噪声掩埋，换句话说接收到但是噪声信号，在正常通信过程中有时无法正常通信。

通过以上定理我们可以看出，当S/N值较小的情况，并且Rmax值已经最大（达到最大的通信能力），为了保证正常通信，我们需要将W值变大，即频谱展宽，也就我们所说的扩频。

以上香农定理应用到实际中，就能够解决通信抗干扰性问题，因此现在的通信中可以引入扩频机制，扩频是cdm的基础。

### 1.3.1扩频通信

扩频通信是香农定理的实际应用，本质上就是增大通信信号的带宽，一般原始信息所表示的信号属于**低频窄带信号**。在应用中想办法将低频窄带信号调制到高频宽带信号上即可实现扩频。

实际中扩频方式有如下几种：

1. Dsss（直接序列扩频）

Tz = seq\_s + seq\_x ；

Seq\_s（低频序列）：离散信号，信息的转化为离散信号，离散信号即原始序列，这里我们又叫做待扩频序列

Seq\_x（高频序列）：离散信号吗，输出序列频率固定，叫做载波序列

Tz:调制序列

1. 跳频（FHSS）

Tz = seq\_s + seq\_x ；

Seq\_s（低频序列）：离散信号，信息的转化为离散信号，离散信号即原始序列，这里我们又叫做待扩频序列

Seq\_x（高频序列）：离散信号吗，叫做载波序列，该序列的频率不固定，每隔一段时间让序列的输出频率随机变化（随机码控制随机频率）

Tz:调制序列

1. 跳时（TH）

Tz = seq\_s + seq\_x ；

Seq\_s（低频序列）：离散信号，信息的转化为离散信号，离散信号即原始序列，这里我们又叫做待扩频序列

Seq\_x（高频序列）：离散信号吗，输出序列频率固定，叫做载波序列

Tz:调制序列，随机改变TZ的时长

1. 线性在调制

跳频的一种特殊情况

Tz = seq\_s + seq\_x ；

Seq\_s（低频序列）：离散信号，信息的转化为离散信号，离散信号即原始序列，这里我们又叫做待扩频序列

Seq\_x（高频序列）：离散信号吗，叫做载波序列，该序列的频率不固定，每隔一段时间让序列的输出频率线性变化

Tz:调制序列

### 1.3.2 m序列

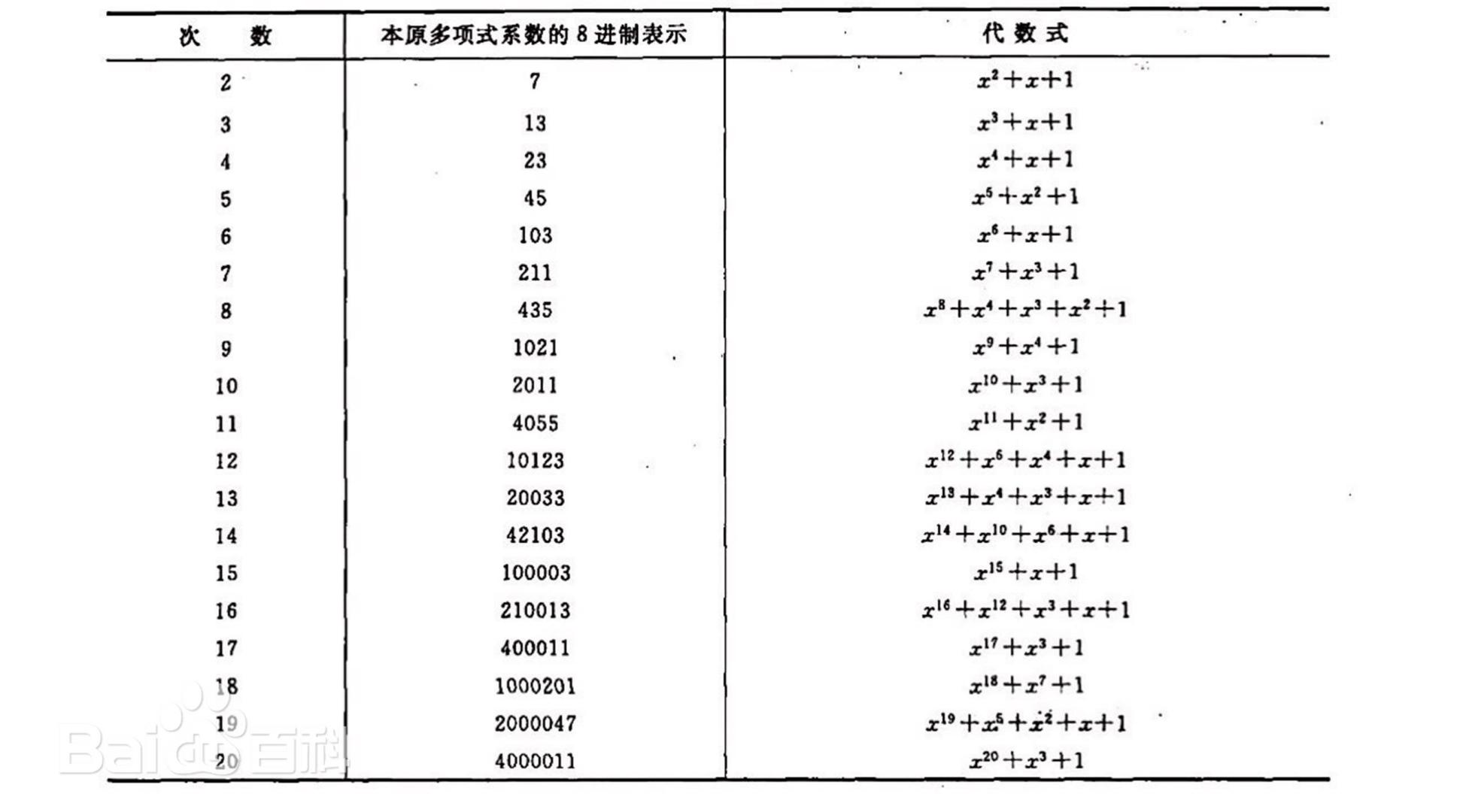
在扩频通信中，有时用随机码或者随机序列，比如直接序列扩频就需要随机序列实现，m序列就是能够用于扩频的随机序列之一。

M序列是通过硬件电路产生的，它通过移位反馈电路产生PN序列，它的电路结构可以用**数学模型**表示。

= 0(N表示阶，A表示移位寄存器。C表示本院多项式系数 ，+ 为模2加)

#### 1.3.2.1本原始多项式

原始多项式是系数的一种表示方式，在扩频m序列电路中，C必须用本原多项式



例如次数为5的本原多项式为x5+x2+1

C5 =1

C4=0

C3=0

C2=1

C1=0

C0=1

#### 1.3.2.2m序列发生器电路

比如N=5

= 0(N表示阶，A表示移位寄存器。C表示本院多项式系数，+ 为模2加 )

A5C0 + A4C1 + A3C2 + A2C3 + A1C4 + A0C5 =0

**A5C0 =-（A4C1 + A3C2 + A2C3 + A1C4 + A0C5）**

**A5 =-（A4C1 + A3C2 + A2C3 + A1C4 + A0C5）**

C1

C2

C3

C4

C5

A0

A1

A2

A3

A4

C1

C2

C3

C4

C5

A1

A2

A3

A4

A0

通过m序列发生器电路可以看出，当N=5的时候一共有31个m序列，我们把这31个m序列叫做**等价相异的m序列**。

练习：

实现31个N=5的m序列发生器电路，产生的这31个序列为等价相异的m序列。

把每个m序列前31比特打印文件中

Always @（posedge clk or negedge rst\_n）

If(~rst\_n)

A40\_1 <= 5’d1;

Else

A40\_1<={ A40\_1[3]^ A40\_1[0], A40\_1[4:1]};

Always @（posedge clk or negedge rst\_n）

If(~rst\_n)

A40\_2 <= 5’d2;

Else

A40\_2<={ A40\_2[3]^ A40\_2[0], A40\_2[4:1]};

….

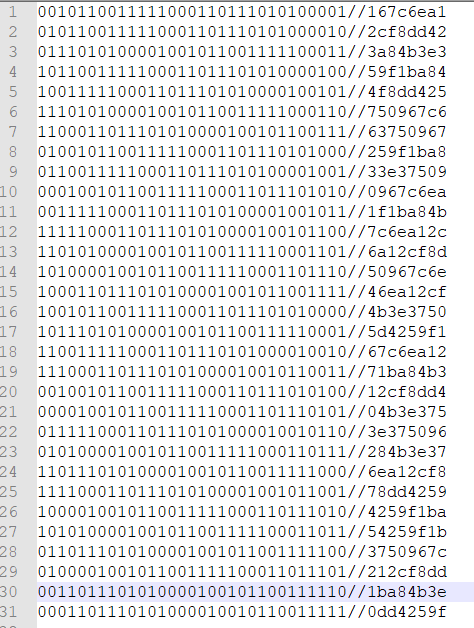
Always @（posedge clk or negedge rst\_n）

If(~rst\_n)

A40\_31 <= 5’d31;

Else

A40\_31<={ A40\_31[3]^ A40\_31[0], A40\_31[4:1]};



#### 1.3.2.3m的特性

M序列所谓PN序列的一种，可以用来扩频，它也有自己的特性

1. **周期性**

M序列的周期是2N-1（N为阶）

例如：

0101010 0101010 0101010 0101010 0101010

1. 均衡性

M序列在每个周期中，1的个数比0的个数多1个

1. **游程特性**‌

在m序列中，状态“0”或“1”连续出现的段称为游程。m序列的一个周期中，游程总数为2n-1，其中“0”和“1”各占一半。这种游程分布表现出随机性。

1. **自相关特性**‌

m序列具有尖锐的自相关特性，即序列与自身的时间延迟版本的相关性很高，而与其他序列的相关性很低。这使得m序列在通信中非常有用，因为它可以减少干扰和误码率。

# 第二节：扩频通信算法验证

扩频调制编码

Dout[K-1:0]

解扩

Noise\_code[N-1:0]

其它通信电路

其它通信电路

（数字）

code[N-1:0]

Info（信息序列）

Din[K-1:0]

数据序列化

并转换

## 2.1扩频调制编码设计

要求：

1. 采用包作为数据传输单位
2. 信道编码采用汉明码，信源编码采用PCM码
3. 通信同步采用最基本位同步的方式
4. 不支持相位同步
5. 通信数据格式

规定每512个信息位为一包数据，即129个汉明码，我们在每包数据之间附加分隔符，也可以当做包头使用，分隔符我们规定7b‘b000000’

7’b0000000

128个汉明码

1. 汉明码

汉明码是最基本一种信道编码，是分组码的一种，表示为(7,4)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 信息位 | | | | 监督位 | | |

D2= D6 + D5 + D4

D1= D6 + D5 + D3

D0= D6 + D4 + D3

1. 通信同步

数字通信：主要依赖于时钟

载体通信：三方（基站，卫星）同步

不管人通信，同步是保证正常数据传输的前提，我们是发送方和接收方也需要进行同步，这里我们我们上电的时候进行一次同步，采用基本位同步方式

位同步 = 空闲序列+同步序列

空闲序列：为了在接收方和发送方之间预留足够的同步时间，这里我们规定10比特的1

同步序列：规定1比特0

### 2.1.1扩频编码结构设计

Info

rst\_n

Clk\_cx

rst\_n

Clk\_c

ready\_info

dsss\_code

code[2:0]

**seq\_s**

flag\_seq\_x

flag\_seq\_s

**seq\_x**

seq\_s\_gen

产生待调制的序列

seq\_x\_gen

**产生载波序列**

tz\_code

调制编码

#### 2.1.1.1待调制序列发生器电路实现

ready\_info

Info

rst\_n

Clk\_c

flag\_seq\_s

seq\_s

seq\_s\_gen

产生待调制的序列

端口说明

|  |  |
| --- | --- |
| 信号名 | 描述 |
| 系统相关 | |
| Clk\_c | 信息时钟 |
| rst\_n | 系统复位 |
| 输入信息序列 | |
| Info | 信息序列 |
| ready\_info | 当ready\_info==1表示准备接收info |
| 输出待调制序列 | |
| seq\_s | 待调制序列 |
| flag\_seq\_s | 当flag\_seq\_s==1表示seq\_s有效 |

Flag\_seq\_s<=1;

Case(cnt)

0:seq\_s<=d0;

1: seq\_s<=d1;

2: seq\_s<=d2;

Endcase

If(cnt ==2) cnt<=0;else cnt<=cnt+1;

If(cnt==2)if(cnt\_hmm==127)

Cnt\_hmm<=0;

Else cnt\_hmm<=cnt\_hmm+1

Seq\_s<=info;

Flag\_seq\_s<=1;

If(cnt ==3) cnt<=0;else cnt<=cnt+1;

If(cnt==2)

Ready\_info<=0

rst\_n

cnt==3

cnt==6

cnt==10

HMM\_JD

HMM\_INFO

PK\_HEDAD

BIT\_SYNC

Seq\_s<=0;

Flag\_seq\_s<=1;

If(cnt ==6) cnt<=0;else cnt<=cnt+1;

If(cnt==5)

Ready\_inf0，=1；

If(cnt == 10)cnt<=0;else cnt<=cnt+1;

If(cnt ==10)begin

Seq\_s<=0;

Flag\_seq\_s<=1;end

Else begin

Seq\_s<=1;

Flag\_seq\_s<=1’end

cnt==2 & cnt\_hmm==127

cnt==2 & cnt\_hmm!=127

always @(posedge clk)begin

If( (state == HMM\_INFO)&(cnt==0)) d3<=info;

If( (state == HMM\_INFO)&(cnt==1)) d4<=info;

If( (state == HMM\_INFO)&(cnt==2)) d5<=info;

If( (state == HMM\_INFO)&(cnt==3)) d6<=info;end

assign d2<= d6 ^d5^4;

assign d1<= d6 ^d5^3;

assign d0<= d6 ^d4^0;

#### 2.1.1.2载波序列发生器电路实现

flag\_seq\_x

seq\_x

seq\_x\_gen

**产生载波序列**

rst\_n

Clk\_cx

端口说明

|  |  |
| --- | --- |
| 信号名 | 描述 |
| 系统相关 | |
| Clk\_cx | 载波序列时钟 |
| rst\_n | 系统复位 |
| 输出待调制序列 | |
| seq\_x | 载波 |
| flag\_seq\_s=x | 当flag\_seq\_x==1表示seq\_x有效 |

#### 2.1.1.3调制编码电路实现

flag\_seq\_x

flag\_seq\_s

seq\_x

seq\_s

tz\_code

调制编码

code[2:0]

|  |  |
| --- | --- |
| 信号名 | 描述 |
| 输入载波序列 | |
| seq\_x | 载波序列 |
| flag\_seq\_x | 当flag\_seq\_x ==1表示seq\_x有效 |
| 输出待调制序列 | |
| seq\_s | 待调制序列 |
| flag\_seq\_s | 当flag\_seq\_s==1表示seq\_s有效 |
| 输出PCM码 | |
| Code[2:0] | PCm编码，默认精度3 |

assign tz= seq\_s ^ seq\_x;

Always @(\*)

If(flag\_seq\_x& flag\_seq\_s)

If(tz)

Code= 3b001;//+1

else

Code= 3b111;//-1

## 2.2扩频解码

扩频解码就是信宿经过一系列的通信把接收到的带噪声的编码数据，根据其特性恢复出原始信息。

code[2:0]

**0,+1,-1**

扩频编码

信源

其他通信

code\_noise[2:0]

[-3:+3]

扩频解码

信宿

其他通信

扩频解码主要完成2个操作，分别是恢复序列，信息恢复

序列恢复：从带噪声的编码数据中，恢复出我们想要的序列

信息恢复：将序列按照包格式，通过信道编码，恢复出原始信息

### 2.2.1扩频解码架构设计

clk\_d

clk\_dx

rst\_n

rst\_n

code\_noise[2:0]

[-3:+3]

Info\_recovery

Info\_data[7:0]

flag\_seq\_rc

seq\_rc

seq\_recovery

flag\_Info\_data

dsss\_decode

### 2.2.2序列恢复程序实现

序列恢复是从带噪声的编码数据中恢复恢复出我们想要的序列，如何才能恢复序列？下来我们就这个问探讨。

待调制原始序列

**低频序列**

0 1 0 1 0 1 1 0 1 0 1

载波序列

**高频序列**

1 0 1 0 1 1 1 0 1 0 1

调制输出序列

**高频序列**

001 111 001 111 001 001 001 111 001 111 001

编码输出

通过以上编码可以看出当带调制的码元为1的时候，编出输出的最高位与当前对应的载波序列相同，码元为0的时候，编出输出的最高位与当前对应的载波序列相反.

**假如高频载波序列的输出频率是待调制的低频序列的N倍**

**cnt =0;**

**If(cnt==N-1)**

**Cnt <=0;**

v

**else Cnt <= Cnt +1;**

有噪声PCM编码

无噪声PCM编码

**sum =0 ;**

**if(cnt ==N-1)**

**sum <=0;**

t

**else if(code\_noise [2] == m[cnt] )**

**sum <= sum + | code\_noise |;**

**else**

**sum <= sum - | code\_noise |;**

序列判定(cnt==N-1)

|  |  |
| --- | --- |
| sum | 恢复序列 |
| >0 | 1 |
| =0 | 不确定 |
| <0 | 0 |

注意:m[cnt] 值是当前序列，解码的时候我们必须需要个用于解码的m序列，为了信息保密，算法中用于解码的m序列只要与编码端的序列相关性较好就可以用来解码。

#### 2.2.2.1程序实现

flag\_seq\_rc

seq\_rc

code\_noise[2:0]

[-3:+3]

seq\_recovery

clk\_dx

rst\_n

端口说明

|  |  |
| --- | --- |
| 信号名 | 说明 |
| 系统相关 | |
| clk\_dx | 序列恢复时钟 |
| rst\_n | 系统复位 |
| 编码输入（带噪声） | |
| code\_noise[2:0] | [-3:+3]  带噪声的编码输入 |
| 输出恢复序列 | |
| seq\_rc | 恢复序列 |
| flag\_seq\_rc | 当flag\_seq\_rc==1表示seq\_rc有效 |

rst\_n

If(cnt==N-1)cnt<=0;else cnt<=cnt+1;

If(cnt==N-1)

Sum0<= 0;

Else if( code\_noise[2] == m0[cnt])

Sum0<= Sum0+ abs\_code\_noise;

Else

Sum0<= Sum0+ abs\_code\_noise;

….

If(cnt==N-1)

Sum30<= 0;

Else if( code\_noise[2] == m30[cnt])

Sum30<= Sum30+ abs\_code\_noise;

Else

Sum30<= Sum30+ abs\_code\_noise;

If(sum0>n-3)mx<=m0;

…

If(sum30>n-3)mx<=m30;

If(cnt==N-1)cnt<=0;else cnt<=cnt+1;

If(cnt==N-1)

Sum0<= 0;

Else if( code\_noise[2] == mx[cnt])

Sum0<= Sum0+ abs\_code\_noise;

Else

Sum0<= Sum0+ abs\_code\_noise;

If(cnt==N-1)

If(sum>=0)

Seq<=1;

Else

Seq<=0;

If(cnt==N-1)cnt<=0;else cnt<=cnt+1;

If(cnt==N-1)

Sum0<= 0;

Else if( code\_noise[2] == mx[cnt])

Sum0<= Sum0+ abs\_code\_noise;

Else

Sum0<= Sum0+ abs\_code\_noise;

cnt==N-1

sum<0

cnt==N-1

((sum0>= (N-3)) |

(sum1>= (N-3)) |

…

(sum30>= (N-3))

SYNC

M\_FIND

WORK

### 2.2.3信息恢复程序设计实现

当我们从带噪声的数据中恢复出的序列，把序列转换为汉明码包数据，**提取有效信息**，并且输出。

信息为（4位）+监督位（位）

….

信息为（4位）+监督位（位）

分隔符（7’d0）

#### 2.2.3.1汉明检纠错

汉明码是一种信道编码编码，这个码具备检错能力（接收方每次收到汉明码，**能够判单收到的码是否正确**），同时具备1比特的纠错能力（当接收方每次接收到的汉明如果7比特数据中错误了1个比特，汉明码是能够将该接收到的码进行矫正，如果7比特数据中错误2个或者2个以上的比特，汉明码是无法矫正）。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 信息位 | | | | 监督位 | | |

D2= D6 + D5 +D4;

编码过程中，D2本质上是D6,D5,D4的偶校验位

D1= D6 + D5 +D3;

编码过程中，D1本质上是D6,D5,D3的偶校验位

D0= D6 + D4 +D3;

**编码过程中，D0本质上是D6,D4,D3的偶校验位**

假如定义S1,S2,S3,表示如下：

S1 = D6 + D5 +D4 + D2;

如果S1=0,D6,D5,D4,D2都是正确的，否则这4个数肯定有错误

S2 = D6 + D5 +D3 + D1;

如果S2=0,D6,D5,D3,D1都是正确的，否则这4个数肯定有错误

S3 = D6 + D4 +D3 + D0;

如果S3=0,D6,D4,D3,D0都是正确的，否则这4个数肯定有错误

这里我们把S1,S2,S3叫做叫校正子，假设汉明码**完全没有错误，或者只错误1比特的情况下，可以得出如下结论**：

|  |  |  |  |
| --- | --- | --- | --- |
| {S1,S2,S3} | 错误位置 | {S1,S2,S3} | 错误位置 |
| 3’b000 | 完全正确 | 3’b011 | D3 |
| 3’b001 | D0 | 3’b101 | D4 |
| 3’b010 | D1 | 3’b110 | D5 |
| 3’b100 | D2 | 3’b111 | D6 |

以上结论不适用于错误2比特或者2比特以上，如果存在这种情况，我们认为我们收到的汉明码是错误的，在实际应用中，我们针对汉明码矫正采取以下措施：

1. 完全正确，错误2比特或者2比特以上，直接输出信息位
2. 错误1比特，按照以上的表格进行矫正（对应错误位取反）

#### 2.2.3.2信息恢复程序实现

rst\_n

flag\_Info\_data

Info\_data[7:0]

flag\_seq\_rc

seq\_rc

Info\_recovery

clk\_d

端口说明

|  |  |
| --- | --- |
| 信号名 | 描述 |
| 系统相关 | |
| clk\_d | 信息时钟 |
| rst\_n | 系统复位，低电平有效 |
| 扩频序列恢复有关 | |
| seq\_rc | 扩频恢复的序列 |
| flag\_seq\_rc | 当flag\_seq\_rc==1表示seq\_rc有效 |
| 有效信息输出 | |
| info\_data [7:0] | 有效信息 |
| flag\_Info\_data | 当flag\_Info\_data==1表示Info\_data有效 |

## 2.3扩频抗干扰性验证

高频时钟，默认它的频率是clk\_c的31倍

clk\_d

clk\_c

信源

clk\_cx

**每次解扩恢复的信息数据按顺序循环写入mem\_dec**

reg [7:0]mem\_dec[`TESTMNUM-1:0];

0:

1；

2：

3：

4;

5；

…

`TESETNUM-1;

并转串

info\_rdy

**1:循环按顺序读取mem\_src数据。**

**2:每次读取的数据进行并串转换，MSB->LSB**

reg [7:0]mem\_src[`TESTMNUM-1:0];

0:

1；

2：

3：

4;

5；

…

`TESETNUM-1;

**噪声采样时钟，默认精度N=3**

wire [2:0] noise;//噪声信号的pcm码，PCM码的值范围[-2:+2]

noise噪声可以使有**规律的信号，或者随机信号**

always @（posedge clk\_c ）

noise <= {$eandom}%3;//随机信号

如果noised 值统计特性符合正态分布则noise为高斯噪声

assign code\_noise = code + noise;

加噪

其它通信

其它通信

code[2:0]

Info\_data[7:0]

flag\_info\_data[7:0]

code\_noise [2:0]

info

解扩

扩频

信宿

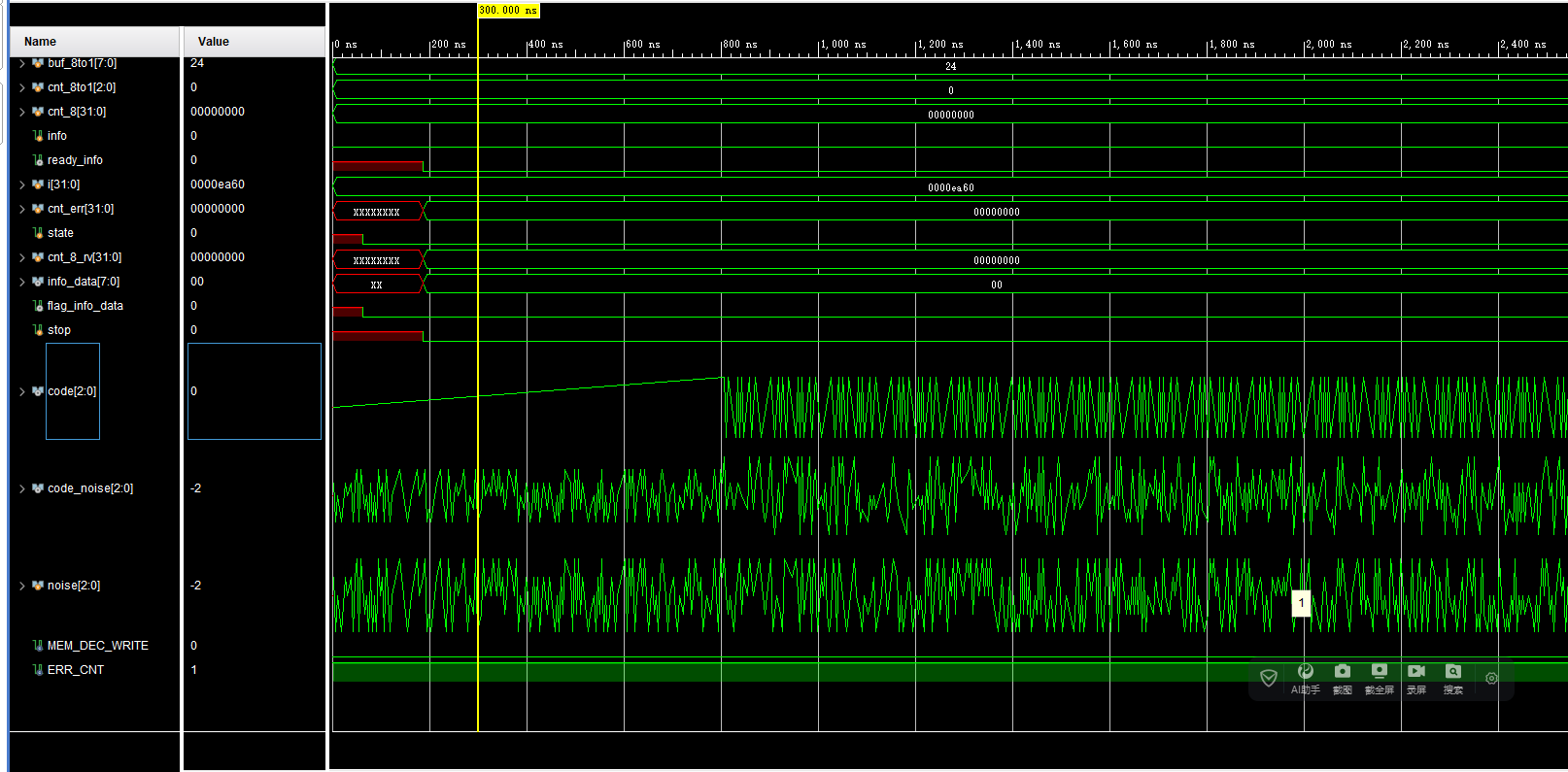
clk\_dx

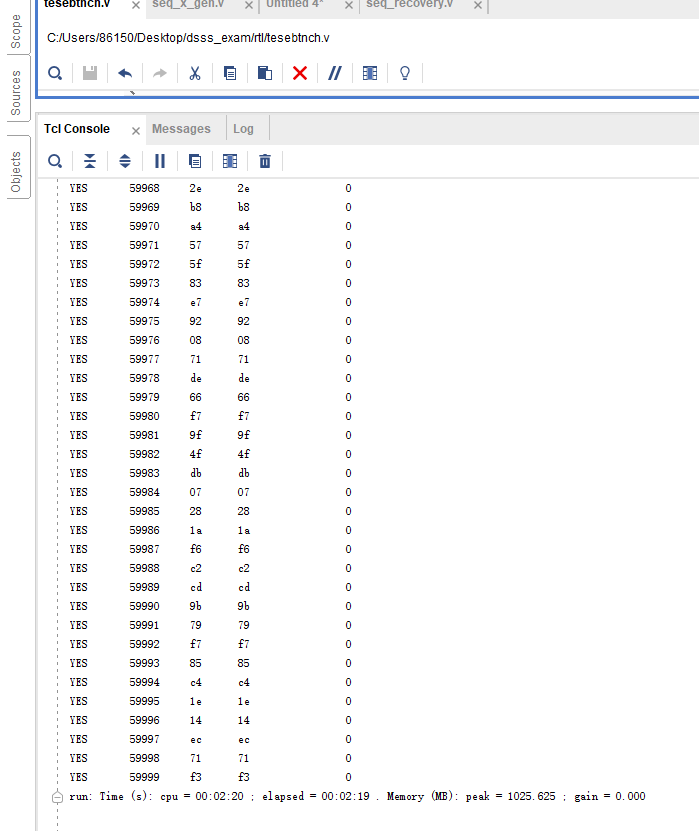
高频时钟，默认它的频率是clk\_d的31倍

抗干扰性验证

我们通过**统计误码率的方式验证抗干扰能力**，当mem\_src[i] == mem\_dec[i]则会说明当前时刻解扩恢复的信息是正确的，否则错误，cnt\_er计数器计数，假如一共测试`TESTNUM个数据，误码率 = ((cnt\_er/`TESTNUM)\*100)%

原则上误码率越高抗干扰性越弱，反之越强。

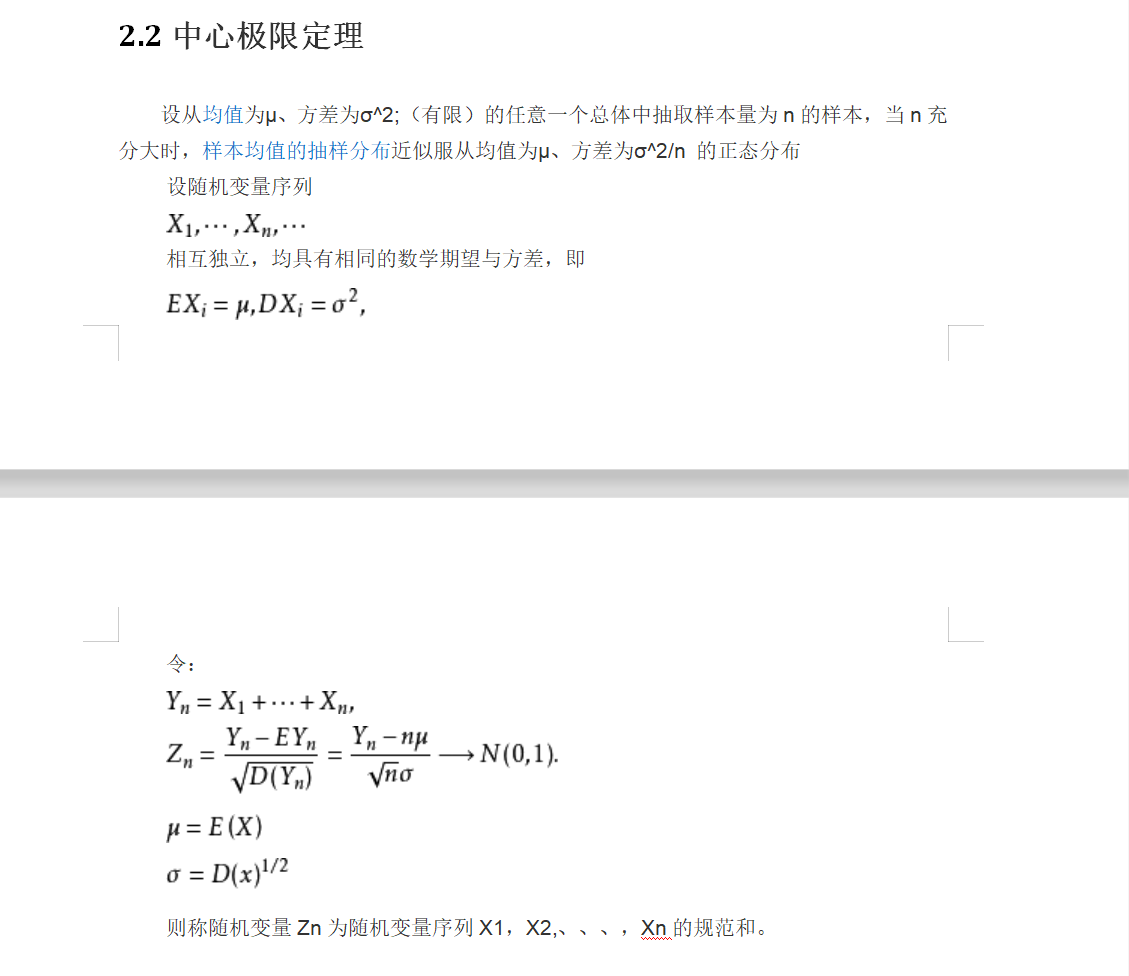




### 2.3.1高斯噪声

高斯噪声是一种常见的噪声，它的概率密度函数服从正态分布，在测试中我们使用的是随机噪声，这里我们讨论高斯噪声的FPGA实现思想。

硬件实现高斯噪声的核心是中心极限定理



Zn = yn -12\*(228/2)/

=

非标准正太分布