# 1. 基带AM调制

## 算法原理

本项目中，为频率为的单音信号，调制深度为，AM信号表达式为

为基带生成部分，为直流0或者1，为调制深度，为0时，为1，为1时，。为载波频率。

## 参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 意义 | 取值范围 | 定点位宽 | 说明 |
|  | 基带过采样率倒数 |  | (64,62) | 有符号数，总位宽64，小数位宽62，符号位1位，上位机通过和FPGA接口速率计算 |
|  | ~~基带过采样率-1~~ |  | ~~(24,0)~~ | ~~该值加1为过采样率，实现方式1可不配置该参数~~ |
|  | AM调制深度 |  | (16,14) | 界面开放参数 |
|  | AM直流偏置 | 0或者1 | (2,0) |  |
|  | 待调制单音信号频率 |  | (28,0) | 界面开放参数 |

## 定点算法描述

方式1：持续相位累加法，在接口速率始终下运行

1. 定义一个内部相位累加寄存器，位宽(64,62)，初始值为，执行(2)；
2. 相位累加寄存器加，再与0x4000000000000000（即十进制1）比较，若大于0x4000000000000000，则相位累加寄存器第63位置0，执行(3)；
3. 将(2)中的相位累加寄存器值四舍五入右移48bit，得到位宽为(16,14)，使用相位累加寄存器低14bit的值查找余弦表，余弦表深度为2^14，真值位宽为(16,14)，执行(4)；
4. 对(3)查得的真值乘以AM调制深度，结果位宽为(32,28)，四舍五入右移14bit，高位饱和2bit，结果位宽为(16,14)，执行(5)；
5. 对(4)的结果加上，结果位宽(17,14)，再四舍五入右移1bit，结果位宽为(16,13)，执行(6)；
6. 将(5)中的结果按照接口速率进行传给DAC，跳转到(2)；

若对过采样率倒数有量化误差，那么多次相位累加会将误差逐渐放大，因此在选取过采样率时，尽量使能够无误差量化过采样率的倒数。

方式2：周期输出法

1. 定义一个内部相位累加寄存器，位宽(64,62)，初始值为，定义一个计数器位宽为(24,0)，执行(2)；
2. 相位累加寄存器加，计数器加1，执行(3)；
3. 使用相位累加寄存器的值查找余弦表，余弦表深度为2^14，真值位宽为(16,14)，执行(4)；
4. 对(3)查得的真值乘以AM调制深度，结果位宽为(32,28)，四舍五入右移14bit，高位饱和2bit，结果位宽为(16,14)，执行(5)；
5. 对(4)的结果加上，结果位宽(17,14)，再四舍五入右移1bit，结果位宽为(16,13)；
6. 若计数器+1，将(5)中的结果~~存入一个循环输出表，并将其~~按照速率进行传给DAC，并转到(2)；否则~~执行(7)~~将相位累加器的初值设置为，将计数器清0，并转到(2)。~~循环输出表深度为+1，真值位宽为(16,13)，+1位宽为(24,0)；~~
7. ~~将循环输出表的真值循环按照速率进行传给DAC。~~

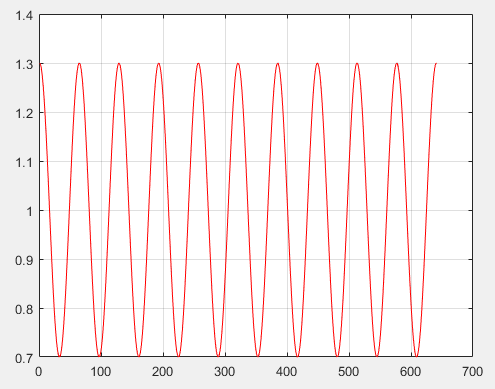
该方法不会因为相位累加次数过多造成的误差放大，适用于周期信号。可以不配置，直接用+1求倒数计算，计算的位宽仍为(64,62)。该方法适用于fs/为整数的情况，如果不满足，周期间可能存在相位跳变，过采样倍数越大，相位跳变越小。

相位累加寄存器做大降低相位量化累积误差，推荐方式1。

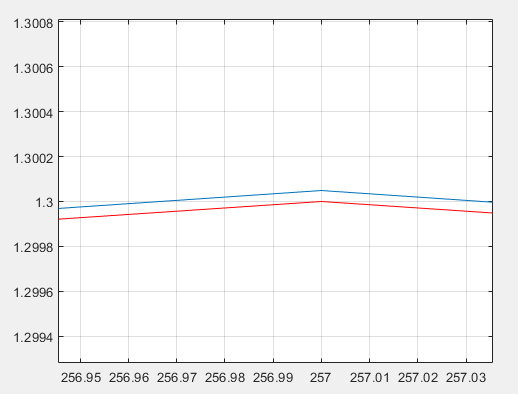
## 仿真验证

### 1.4.1 仿真1

配置=1/64，(64,62)位宽定点化量化，无误差。调制深度0.3，=1，按照上述方式1定点化，仿真1000个周期，定点输出(蓝)vs全浮点输出(红)



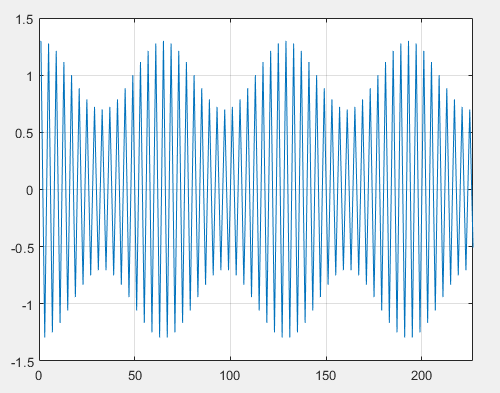
局部放大



统计得到均方误差为5.455\*10-5，定浮点匹配度很高。

### 1.4.2 仿真2

配置，(16,14)位宽定点化量化，无误差。调制深度0.3，=1，按照上述方式1定点化，并且添加混频载波，仿真10个周期，用于观察AM调制的效果。



# 2. 基带FM调制

## 2.1 算法原理

本项目中，待调制信号为频率为的余弦信号，其表达式如下

为载波频率，与两项为基带生成部分，为FM调制系数，其为最大频偏和的比值。

## 2.2 参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 意义 | 取值范围 | 定点位宽 | 说明 |
|  | FM调制指数除以 |  | (48,23) | 界面开放参数 |
|  | 基带过采样率倒数 |  | (64,62) | 上位机计算配置给FPGA |
|  | ~~基带过采样率-1~~ |  | ~~(16,0)~~ | ~~该值加1为过采样率，实现方式1可不配置该参数~~ |
|  | 待调制单音信号频率 |  | (28,0) | 界面开放参数，与AM共用 |

## 2.3 定点算法描述

方式1：持续相位累加法，在接口速率时钟下运行

1. 定义一个内部相位累加寄存器，位宽(64,62)，初始值为，执行(2)；
2. 相位累加寄存器加，再与0x4000000000000000（即浮点数1）比较，若大于0x4000000000000000，则相位累加寄存器第63位置0，执行(3)；
3. 将(2)中的结果四舍五入右移48bit，结果位宽为(16,14)，执行(4)；
4. 使用相位累加寄存器低14bit的值查找正弦表（查正弦表可直接使用余弦表，方法为：相位累加寄存器低14bit+0x3000，再取低14bit查余弦表），余弦表深度为2^14，真值位宽为(16,14)，执行(5)；
5. 对(4)查得的正弦真值乘以FM调制指数，结果位宽为(64,37)，四舍五入右移23bit，取低14bit用于分别查找余弦/正弦真值表（利用余弦表查正弦值的方法参考(4)），余弦/正弦查找真值结果位宽为(16,14)，执行(6)；
6. 将(5)中的余弦/正弦真值结果按照接口速率分别从I/Q路传给DAC，跳转到(2)；

~~若对过采样率倒数有量化误差，那么多次相位累加会将误差逐渐放大，因此在选取过采样率时，尽量使能够无误差量化过采样率的倒数。~~使用较大位宽的相位累加器和，可以大大降低量化误差造成的相位累积误差。

~~方式2：周期输出法~~

1. ~~定义一个内部相位累加寄存器，位宽(32,30)，初始值为，定义一个计数器位宽为(16,0)，执行(2)；~~
2. ~~相位累加寄存器加，计数器加1，执行(3)；~~
3. ~~将(2)中的相位累加器结果四舍五入右移16bit，结果位宽为(16,14)，执行(4)~~
4. ~~使用相位累加寄存器低14bit的值查找正弦表（正弦查找方法同方式1），余弦表深度为2^14，真值位宽为(16,14)，执行(5)；~~
5. ~~对(4)查得的正弦查表真值乘以FM调制指数，结果位宽为(64,37)，分别四舍五入右移23bit，取低14bit用于分别查找余弦/正弦真值表（正弦查找方法同方式1），余弦/正弦结果位宽为(16,14)，执行(6)；~~
6. ~~若计数器+1，将(5)中的余弦/正弦真值结果存入一个循环输出表，并将其按照速率从分I/Q两路传给DAC，并转到(2)；否则转到(7)。循环输出表深度为+1，真值位宽为(16,14)，+1位宽为(16,0)；~~
7. ~~将循环输出表的真值循环按照速率进行传给DAC。~~

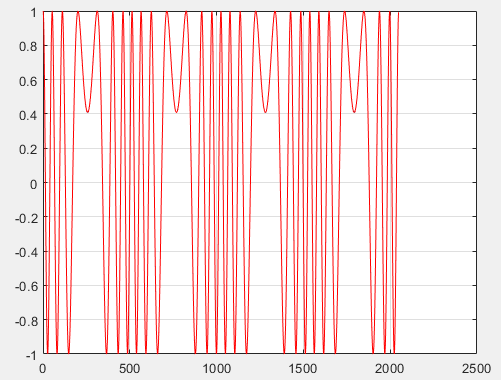
~~该方法不会因为相位累加次数过多，超过一个波形周期造成的误差放大，适用于周期信号。可以不配置，直接用+1求倒数计算，计算的位宽仍为(16,14)。但是如果过采样率+1太大，单个周期的采样点数也比较大，I/Q两路循环输出表的存储深度比较大。~~

## 2.4 仿真验证

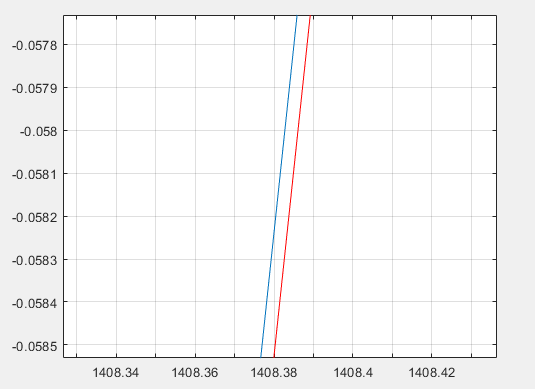
### 2.4.1 仿真1

配置=1/1024，(64,62)位宽定点化量化，无误差。，(48,23)位宽量化，定点输出(蓝)vs全浮点输出(红)。仿真2个周期。

基带输出I路定点vs浮点



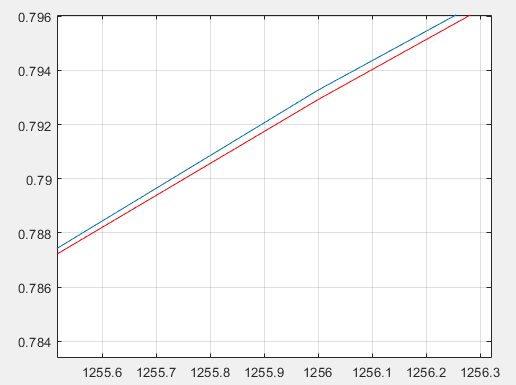
局部放大，测得I路定浮点均方误差为2.9714\*10-4。



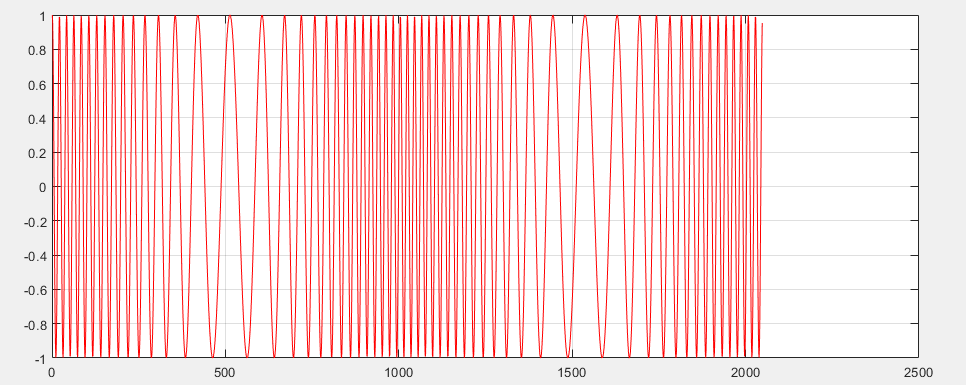
基带输出Q路定点vs浮点



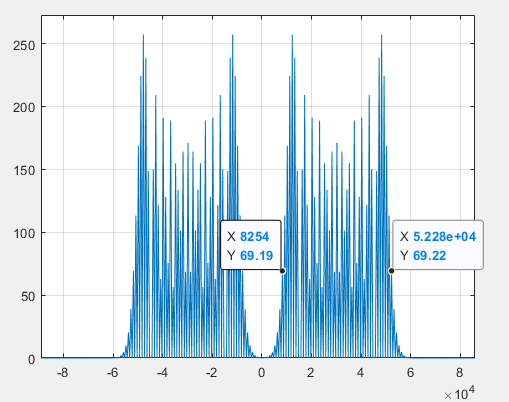
局部放大，测得Q路定浮点均方误差为3.1444\*10-4。



再仿真IQ输出正交调制到载波fc=30\*f0的位置vs全浮点直接调制到载波fc上对比，查看FM效果。



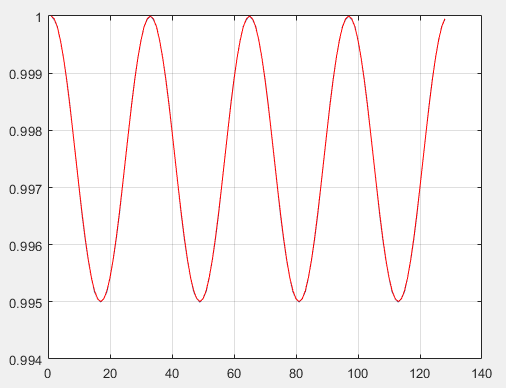
其频谱如下，假设fm=1kHz，调制系数20，fc=20kHz，频谱符合理论预期。



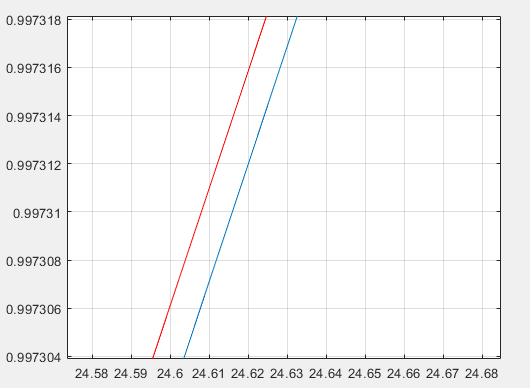
### 2.4.2 仿真2

配置=1/64，(64,62)位宽定点化量化，无误差。，(48,23)位宽量化，定点输出(蓝)vs全浮点输出(红)。仿真2个周期。

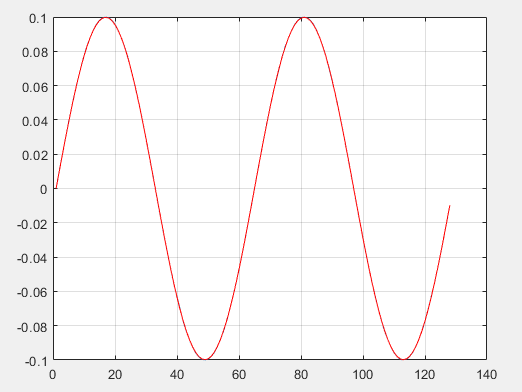
基带输出I路定点vs浮点



局部放大，测得I路定浮点均方误差为9.0851\*10-6。

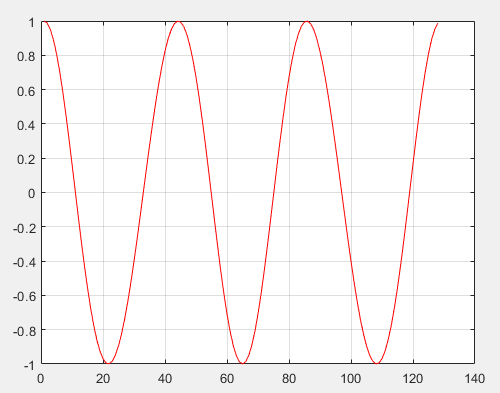


基带输出Q路定点vs浮点

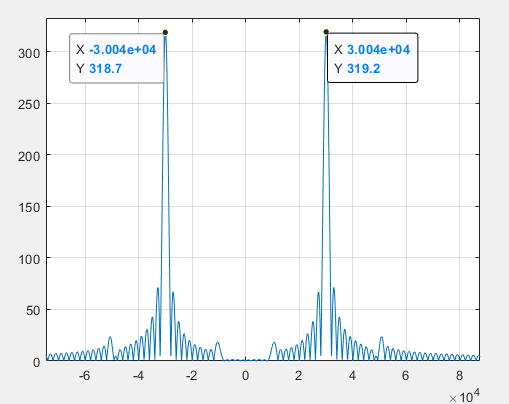


局部放大，测得Q路定浮点均方误差为0.0015。

再仿真IQ输出正交调制到载波fc=30kHz的位置vs全浮点直接调制到载波fc上对比，查看FM效果。



该窄带FM调制频谱如下：



# 3. 基带PM调制

## 3.1 算法原理

本项目中，待调制信号为频率为的余弦信号，其表达式如下

为载波频率，与两项为基带生成部分，为PM调制系数。

## 3.2 参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 意义 | 取值范围 | 定点位宽 | 说明 |
|  | PM调相灵敏度除以2π |  | (48,23) | 界面开放参数，可与FM调制指数共用 |
|  | 基带过采样率倒数 |  | (64,62) |  |
|  | 待调制单音信号频率 |  | (28,0) | 界面开放参数，与AM/FM共用 |

## 3.3 定点算法描述

在FPGA和DAC接口速率时钟频率上运行。

1. 定义一个内部相位累加寄存器，位宽(64,62)，初始值为，执行(2)；
2. 相位累加寄存器加，再与0x4000000000000000（即浮点数1）比较，若大于0x4000000000000000，则相位累加寄存器第63位置0，执行(3)；
3. 将(2)中的结果四舍五入右移48bit，结果位宽为(16,14)，执行(4)；
4. 使用相位累加寄存器低14bit的值查找正弦表（查正弦表可直接使用余弦表，方法为：相位累加寄存器低14bit+0x3000，再取低14bit查余弦表），余弦表深度为2^14，真值位宽为(16,14)，执行(5)；
5. 对(4)查得的正弦真值乘以PM调相灵敏度，结果位宽为(64,37)，四舍五入右移23bit，取低14bit用于分别查找余弦/正弦真值表（利用余弦表查正弦值的方法参考(4)），余弦/正弦查找真值结果位宽为(16,14)，执行(6)；
6. 将(5)中的余弦/正弦真值结果按照接口速率分别从I/Q路传给DAC，跳转到(2)；

## 3.4 仿真验证

由于PM调制本质同FM调制，本节略。

# 4. 基带ASK调制

## 4.1 算法原理

ASK信号可表示为

其中，第n个发送符号电平，是符号成型，是载波，是符号宽度。为基带生成部分，符号成型滤波系数通过软件配置FPGA。

对于2ASK，

当采用升余弦滤波器时，其生成表达式如下：

当采用根升余弦滤波器时，其生成表达式如下：

当采用高斯滤波器时，其生成表达式如下：

B=BTs/Ts， BTs为高斯滤波器配置参数，1/Ts为符号速率：

## 4.2 参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 意义 | 取值范围 | 定点位宽 | 说明 |
|  | 最大129阶fir成型滤波系数 |  | (16,14) | 有软件计算好配置给FPGA，因为滤波器系数对称，这里也可以只配置一半的滤波器系数 |
|  | fir成型滤波系数阶数 |  | (9,0) | 如果滤波器系数只配置一半，阶数也只配置一半 |
|  | 脉冲成型滤波对符号的过采样插值个数 |  | (4,0) | 界面开放参数，界面配置成符号成型过采样率 |
|  | 是否进行成型滤波 |  | (1,0) | 1:进行成型滤波；0：不进行成型滤波 ，界面开放参数 |
|  | 成型滤波器类型 |  |  | 界面开放参数，0：升余弦滤波器，1：根升余弦滤波器，2：高斯滤波器 |
|  | 成型滤波器滚降参数 |  | 浮点 | 界面开放参数，用于软件计算成型滤波器系数， |
|  | 成型滤波器符号截断个数 |  | 浮点 | 界面开放参数，用于软件计算成型滤波器系数 |
|  | 符号率 |  | 浮点 | 用于软件计算插值倍数等控制FPGA成型上采样等流程，和FPGA的时钟速率 |

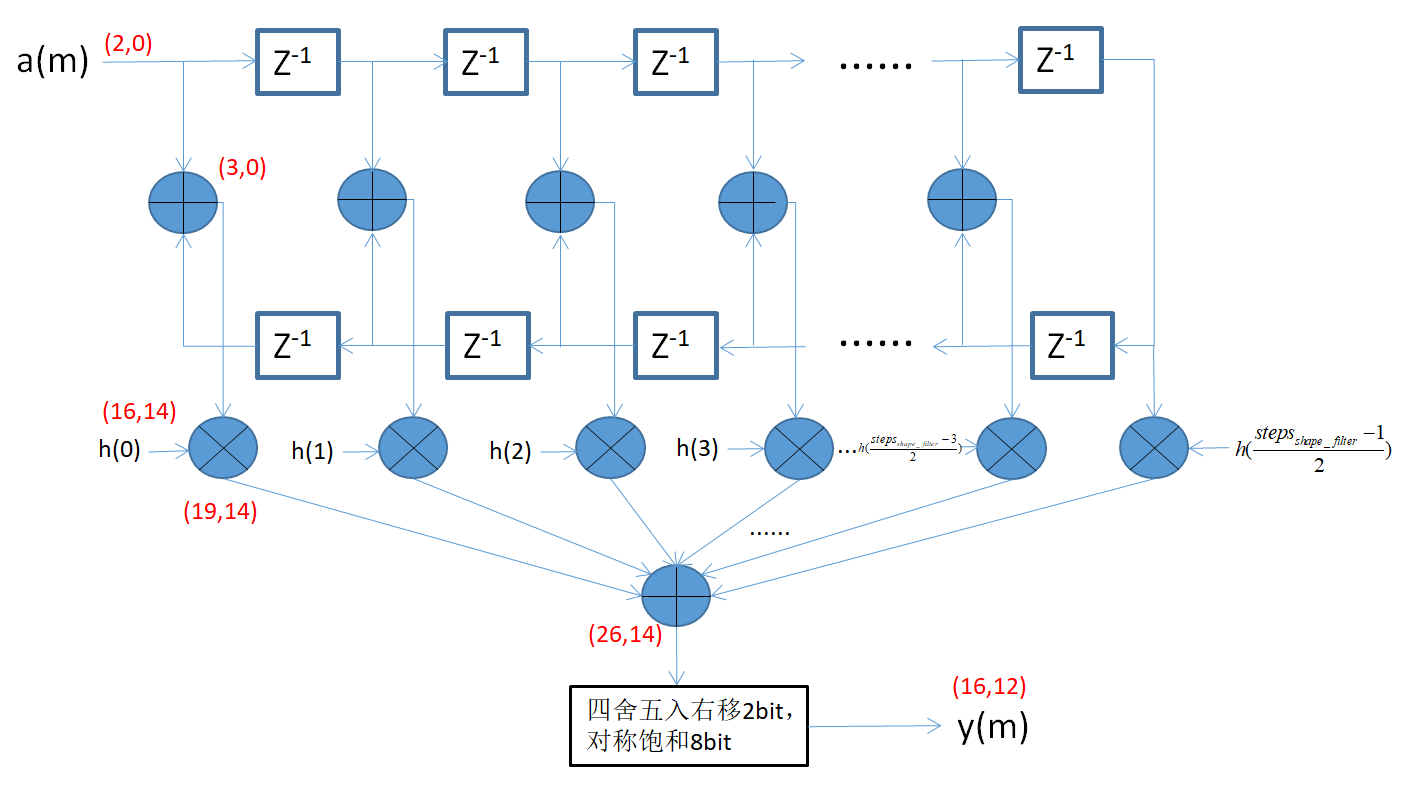
## 4.3 定点算法描述

（1）根据判断是否需要进行成型滤波，若是，执行步骤（2），否则执行步骤（5）；

（2）软件配置129阶fir成型滤波系数，执行步骤（3）；

（3）对符号，每两个符号间插入个0得到，此时样点速率提升为的+1倍，然后执行步骤（4）；

（4）对使用的前个系数进行阶的FIR成型滤波，其实现结构及各节点位宽（红色字体）如下图所示，然后将插值滤波后的符号y(m)送给后级做速率匹配上采样（速率匹配上采样过程见对应章节描述），Q路为0。



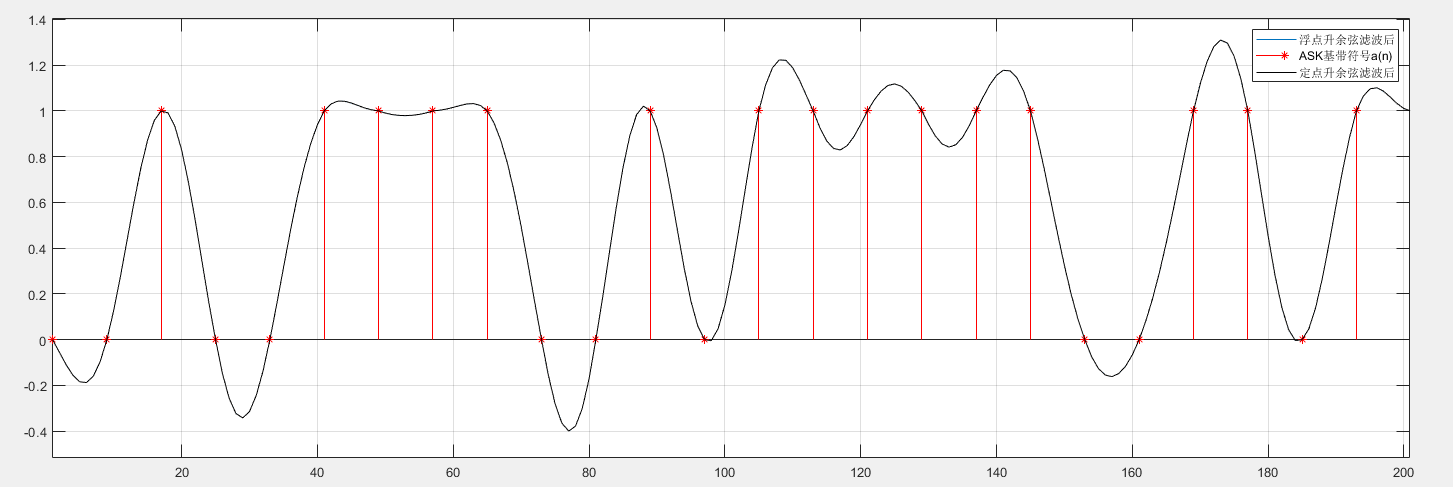
（5）对符号，每两个符号间插入个值得到，直接将a(m)送给后级做速率匹配上采样（速率匹配上采样过程见对应章节描述），Q路为0。

## 4.4 仿真验证

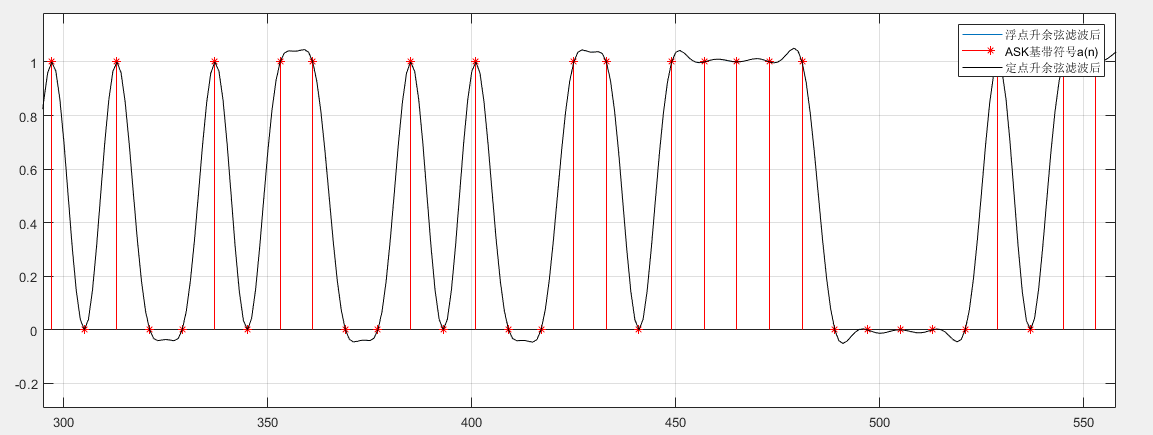
### 4.4.1 仿真1

使用升余弦滤波器。

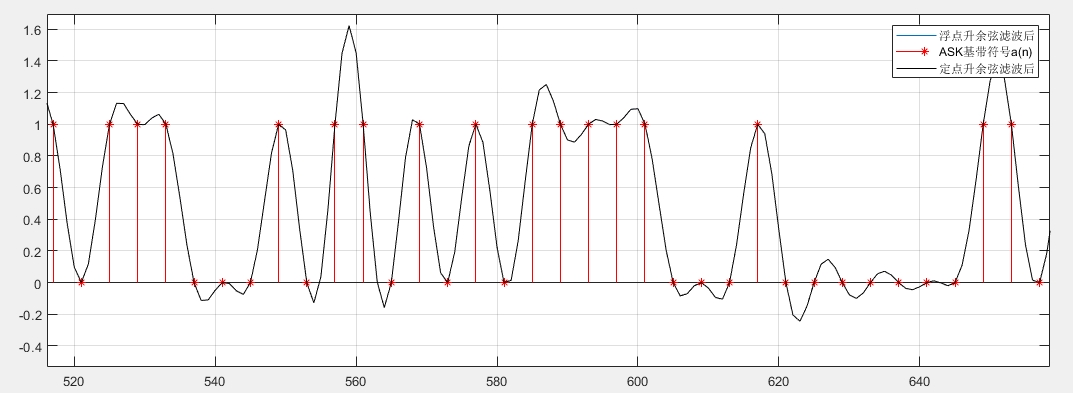
配置129阶滚降系数为0.1，=7的升余弦滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为1.2743\*10-4。



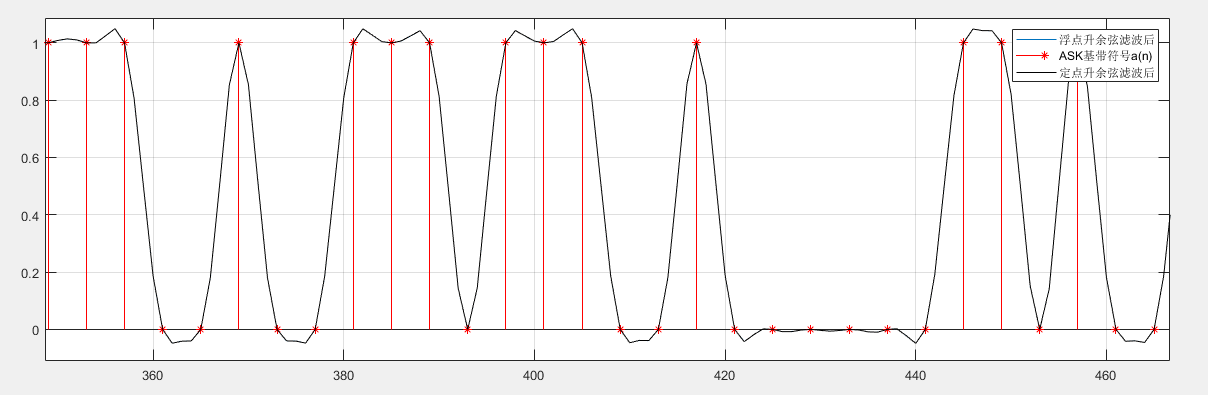
配置129阶滚降系数为0.9，=7的升余弦滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为1.3739\*10-4。



配置129阶滚降系数为0.1，=3的升余弦滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为1.2160\*10-4。



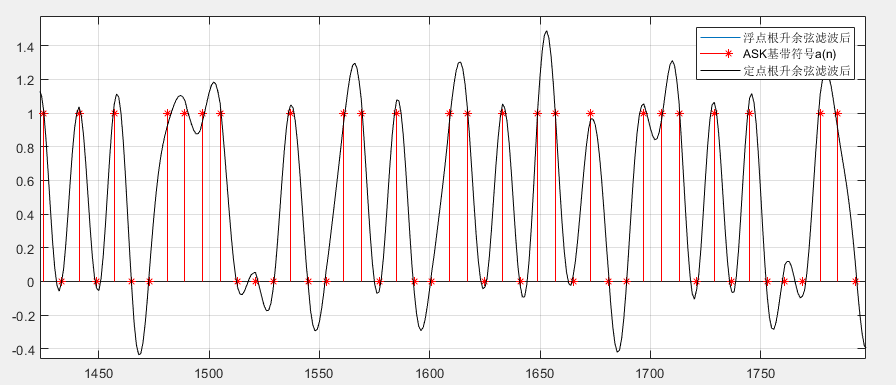
配置129阶滚降系数为0.9，=3的升余弦滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为1.2560\*10-4。



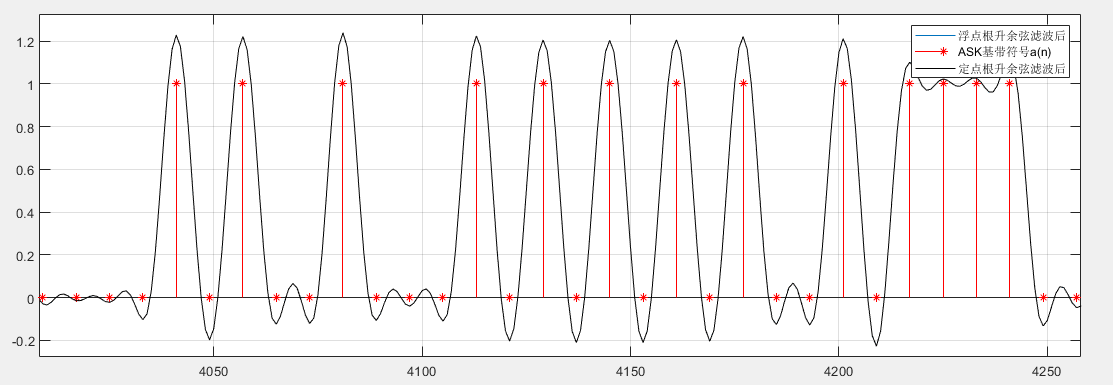
### 4.4.2 仿真2

使用根升余弦滤波器。

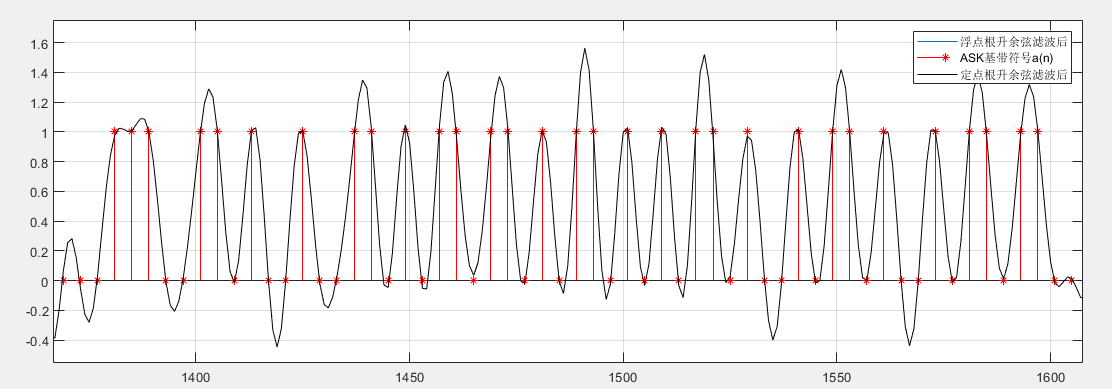
配置129阶滚降系数为0.1，=7的根升余弦滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为1.4864\*10-4。



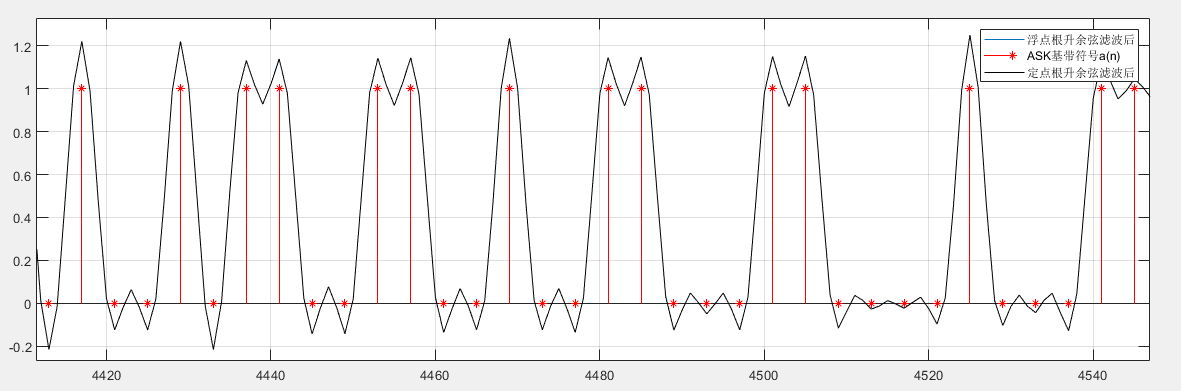
配置129阶滚降系数为0.9，=7的根升余弦滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为1.3568\*10-4。



配置129阶滚降系数为0.1，=3的根升余弦滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为1.2918\*10-4。



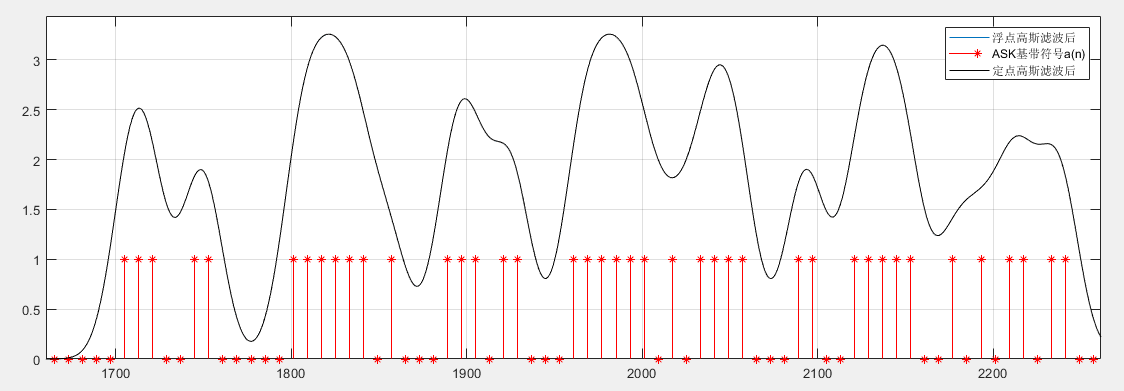
配置129阶滚降系数为0.9，=3的根升余弦滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为2.2432\*10-4。



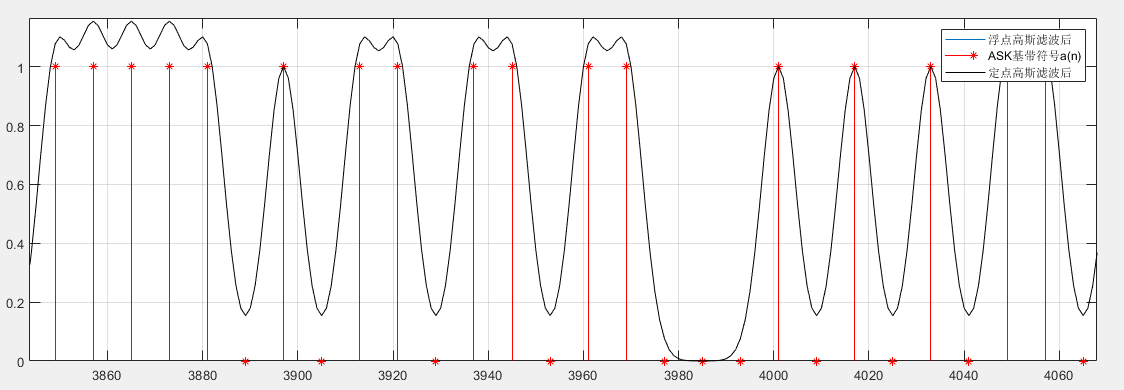
### 4.4.3 仿真3

使用高斯滤波器。

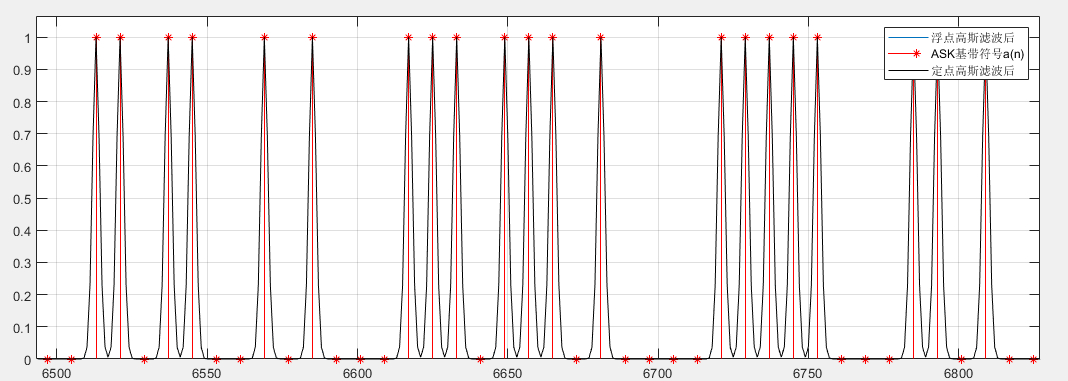
配置129阶高斯滤波器BTs=0.1，=7的高斯滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为4.3899\*10-5。



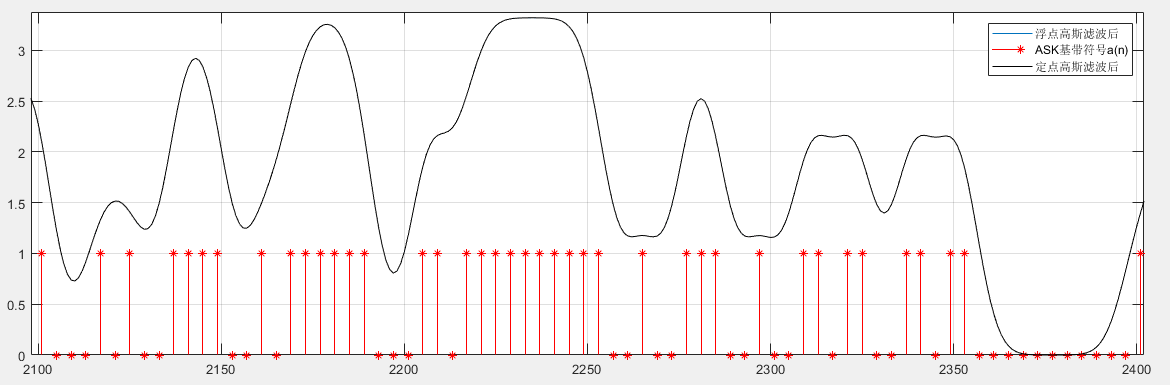
配置129阶高斯滤波器BTs=0.3，=7的高斯滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为1.1459\*10-4。



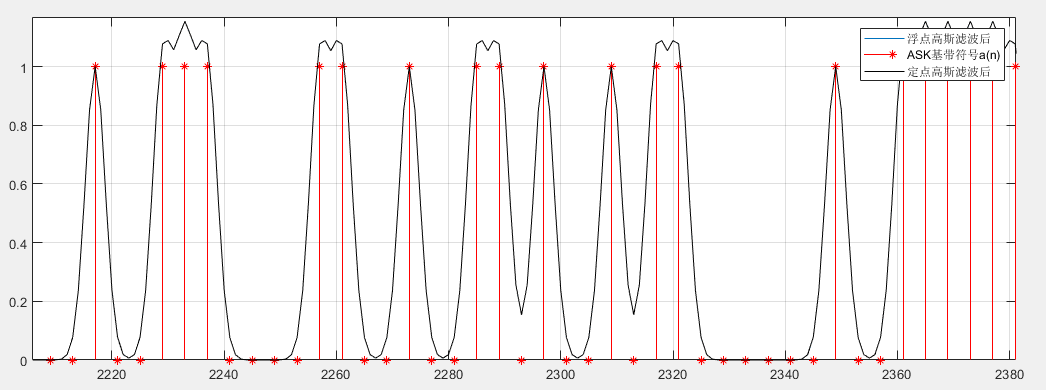
配置129阶高斯滤波器BTs=0.9，=7的高斯滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为2.1933\*10-4。



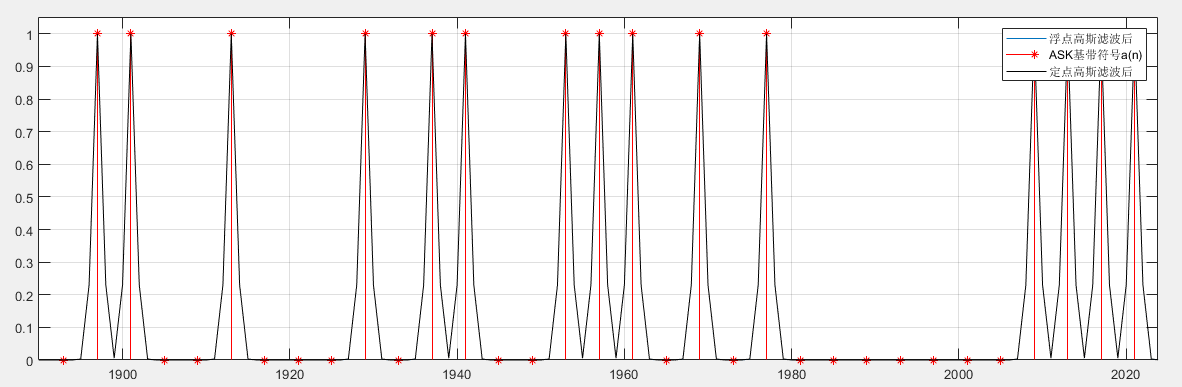
配置129阶高斯滤波器BTs=0.1，=3的高斯滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为5.0603\*10-5。



配置129阶高斯滤波器BTs=0.3，=3的高斯滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为1.1831\*10-4。



配置129阶高斯滤波器BTs=0.9，=3的高斯滤波器，对随机生成的01bit序列进行本章所述处理过程的定浮点仿真对比如下，计算得到定浮点均方误差为1.6271\*10-4。



# 5. 基带FSK调制

## 5.1 算法原理

2FSK信号可以表示为

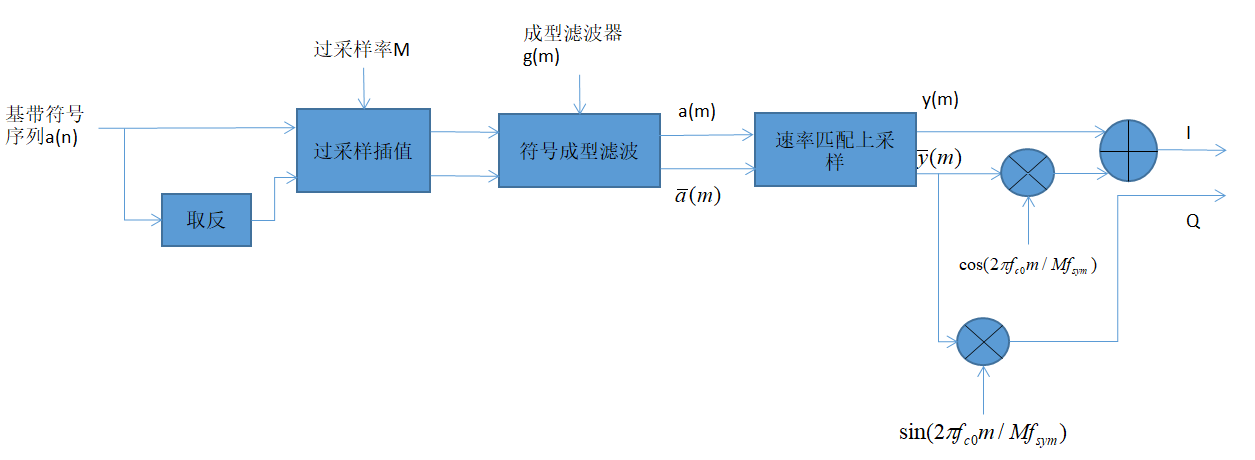
是基带01bit信息序列的按位取反，是符号成型冲击响应，和是2FSK的两个载波，是符号宽度。上述表达式可以变形为：

FPGA完成I路调制和Q路调制，载波为的正交调制由后级正交调制和混频功能完成。

## 5.2 参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 意义 | 取值范围 | 定点位宽 | 说明 |
|  | 129阶fir成型滤波系数 |  | (16,14) | 有软件计算好配置给FPGA，因为滤波器系数对称，这里也可以只配置一半的滤波器系数 |
|  | fir成型滤波系数阶数 |  | (9,0) | 如果滤波器系数只配置一半，阶数也只配置一半 |
|  | 脉冲成型滤波对符号的过采样插值个数 |  | (4,0) | 界面开放参数，界面配置成符号成型过采样率 |
|  | 是否进行成型滤波 |  | (1,0) | 界面开放参数，1:进行成型滤波；0：不进行成型滤波 |
|  | 成型滤波器类型 |  |  | 界面开放参数，0：升余弦滤波器，1：根升余弦滤波器，2：高斯滤波器 |
|  | 基带过采样率倒数 |  | (64,62) | 有符号数，总位宽64，小数位宽62，符号位1位 |
|  | 成型滤波器滚降参数 |  | 浮点 | 界面开放参数，用于软件计算成型滤波器系数， |
|  | 成型滤波器符号截断个数 |  | 浮点 | 界面开放参数，用于软件计算成型滤波器系数 |
|  | 符号率 |  | 浮点 | 界面开放参数，用于软件计算插值倍数等控制FPGA成型上采样等流程，和FPGA的时钟速率 |
|  | 2FSK载波1频率 |  | 浮点 | 界面开放参数 |
|  | 2FSK载波2频率 |  | 浮点 | 界面开放参数，载波1和载波2频率差应做限制小于80M |

## 5.3 定点算法描述



（1）根据判断是否需要进行成型滤波，若是，执行步骤（2），否则执行步骤（5）；

（2）软件配置129阶fir成型滤波系数，执行步骤（3）；

（3）对符号及取反后的符号，每两个符号间插入个0得到和，此时样点速率提升为及的+1倍，然后执行步骤（4）；

（4）对及使用的前个系数进行阶的FIR成型滤波，成型滤波实现结构及各节点位宽同基带ASK调制对应描述，滤波输出位宽为（16,12），然后将插值滤波后的IQ路符号送给后级做速率匹配上采样得到y(m)和（速率匹配上采样过程见对应章节描述），执行步骤（6）。

（5）对符号 及取反后的符号，每两个符号间插入个及值得到和，直接将和送给后级做速率匹配上采样得到y(m)和（速率匹配上采样过程见对应章节描述），速率匹配上采样到接口速率，执行步骤（6）。

（6）定义一个内部相位累加寄存器，位宽(64,62)，初始值为，执行步骤（7）；

（7）相位累加寄存器加，再与0x4000000000000000（即浮点数1）比较，若大于0x4000000000000000，则相位累加寄存器第63位置0，执行(8)；

（8）将(7)中的结果四舍五入右移48bit，结果位宽为(16,14)，执行(9)；

（9）将(8)中的结果分别查找正余弦表（查正弦表可直接使用余弦表，方法为：相位累加寄存器低14bit+0x3000，再取低14bit查余弦表），余弦表深度为2^14，真值位宽为(16,14)，如果的符号为负，对查得的正弦真值取反，执行（10）；

（10）对步骤（4）或者（5）输出的分别乘以步骤（9）中的正余弦真值，位宽为（32,26），再四舍五入右移14bit，对称饱和2bit，得到位宽为（16,12），执行（11）；

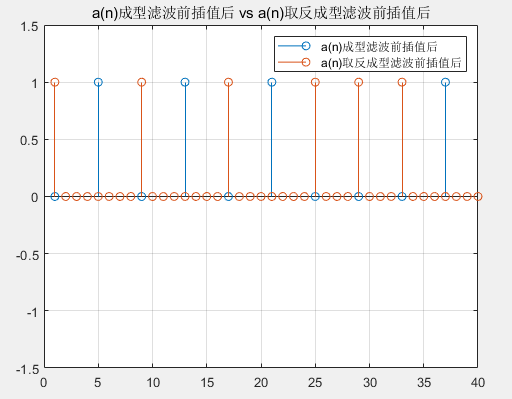
（11）步骤（10）中余弦一路与（4）或者（5）中的y(m)相加，位宽为（17,12），四舍五入1bit，得到（16,11）作为I路输出给DAC，步骤（10）中的正弦一路四舍五入右移1bit视作（16,11）作为Q路输出给DAC。

## 5.4 仿真验证

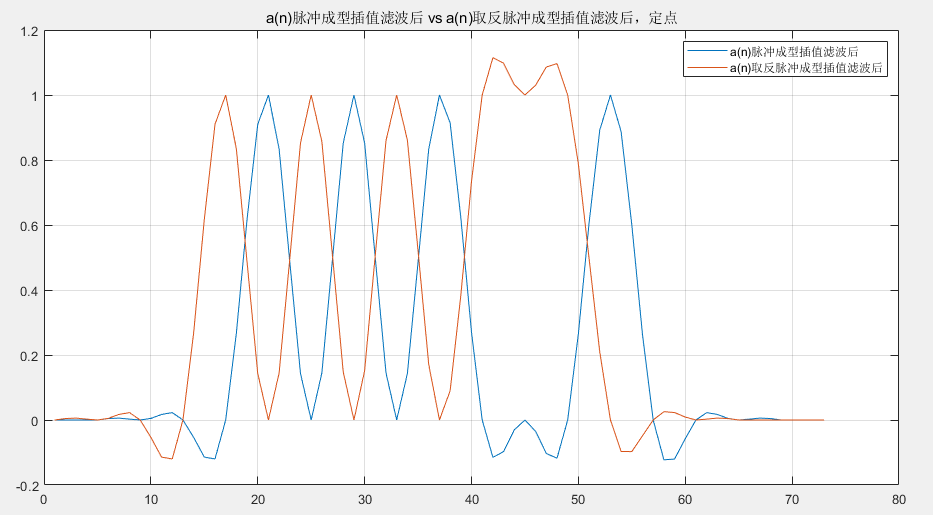
### 5.4.1 仿真1

仿真一个10位随机01信息比特，符号速率为1KHz，经过4倍上采+升余弦符号成型滤波，再加4级2倍上采样加半带滤波，再加16倍上采加cic滤波，总共1024倍上采，载波1的频率为123Hz，载波2频率为37Hz。

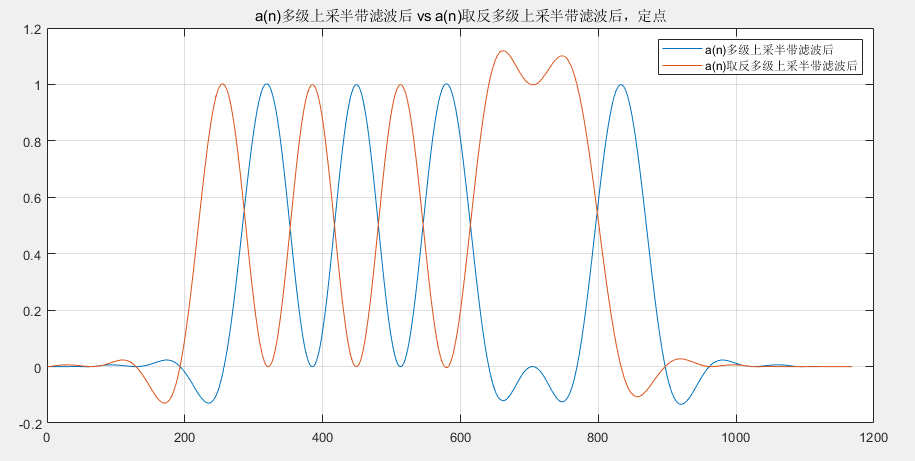
对a(n)和a(n)取反再做4倍插值之后如下，01信息比特为0 1 0 1 0 1 0 0 0 1，数据量变为原来的4倍。



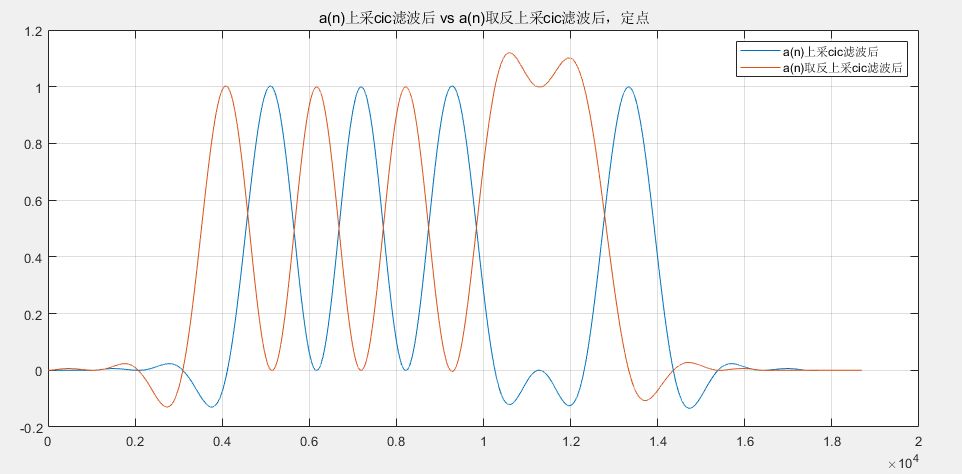
然后进行滚降系数为0.5，符号截断个数为8的定点升余弦滤波。



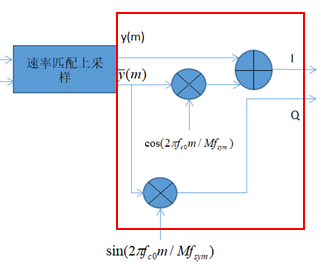
然后再进行定点4级2倍上采半带滤波，数据量变为原始的64倍。

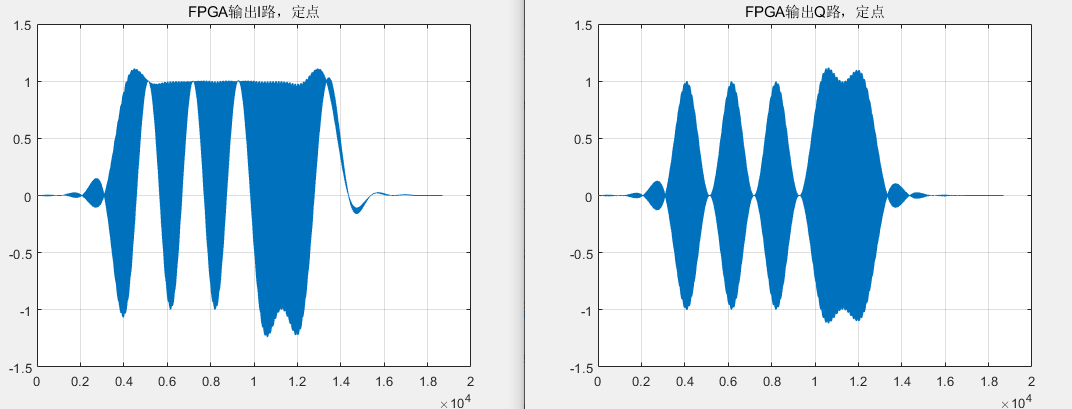


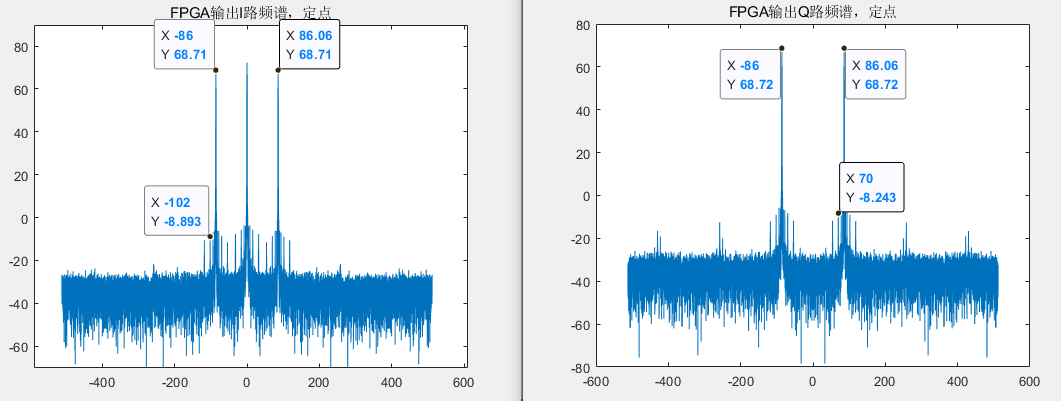
然后再进行定点16倍上采CIC滤波，数据量变为原始的1024倍。



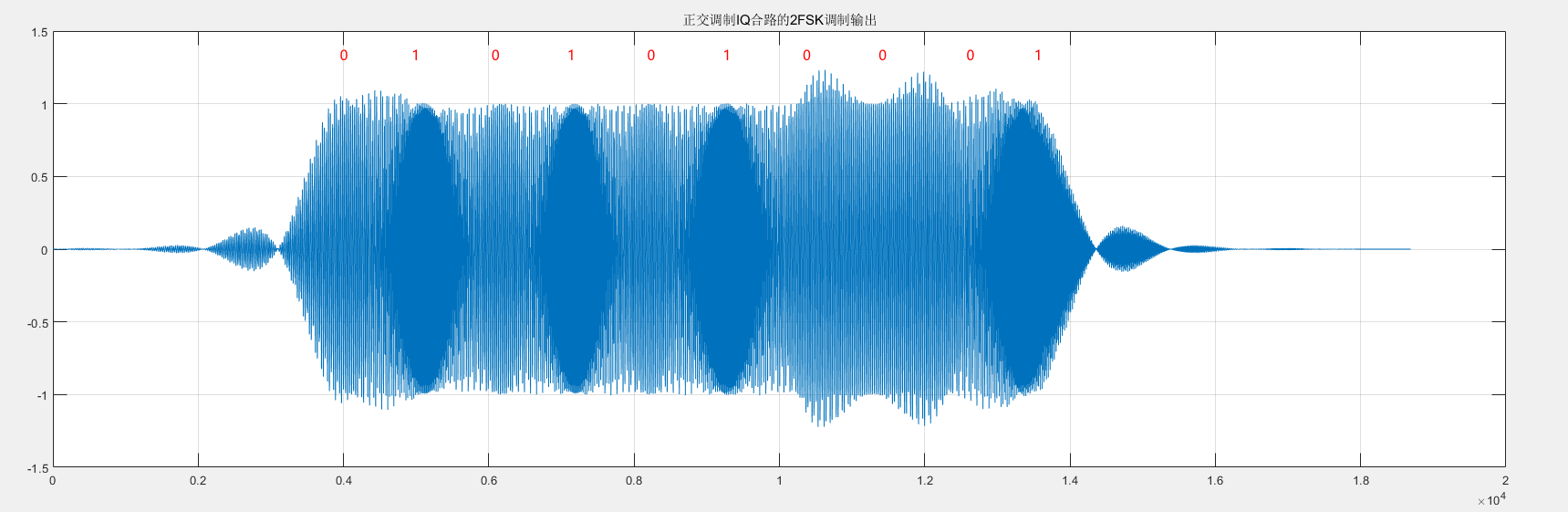
然后对上述CIC滤波输出的两路信号按照如下红框过程进行5.3中对应描述的定点处理，得到IQ路的信号和频谱分别如下：

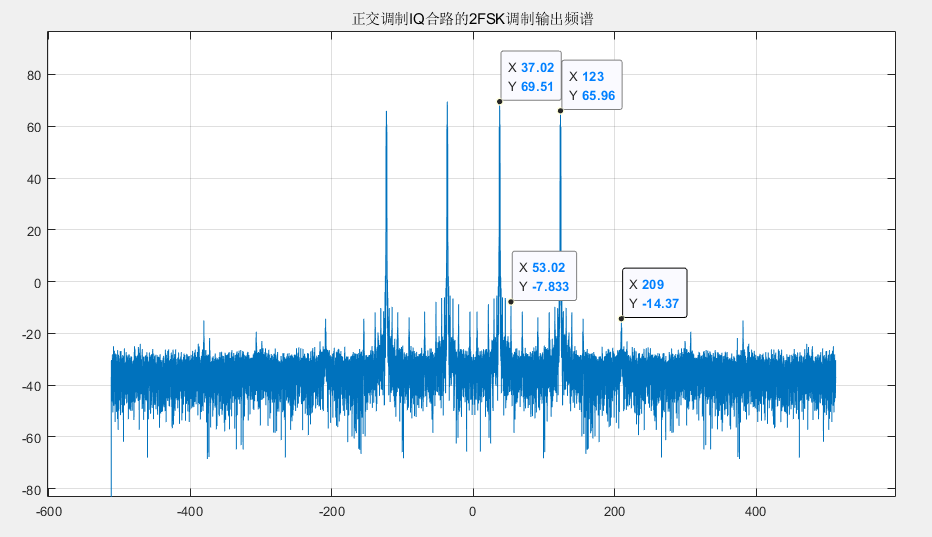




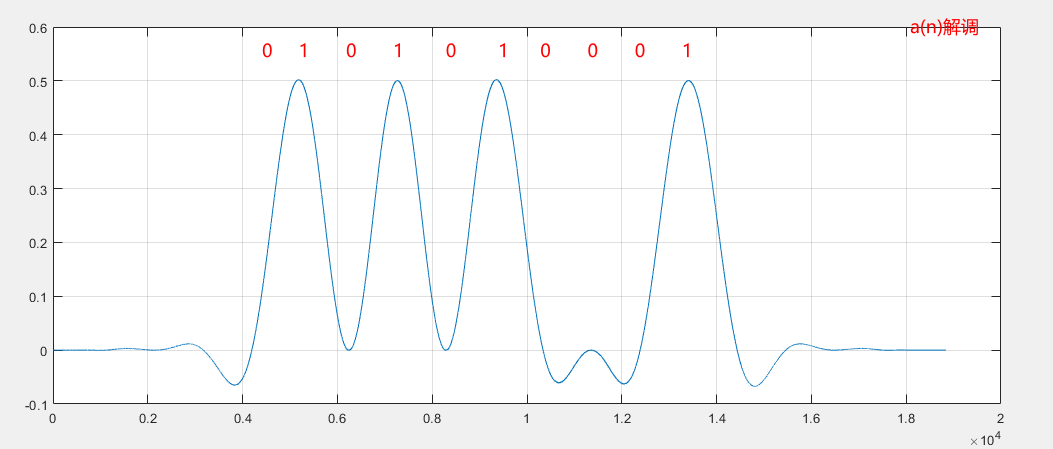


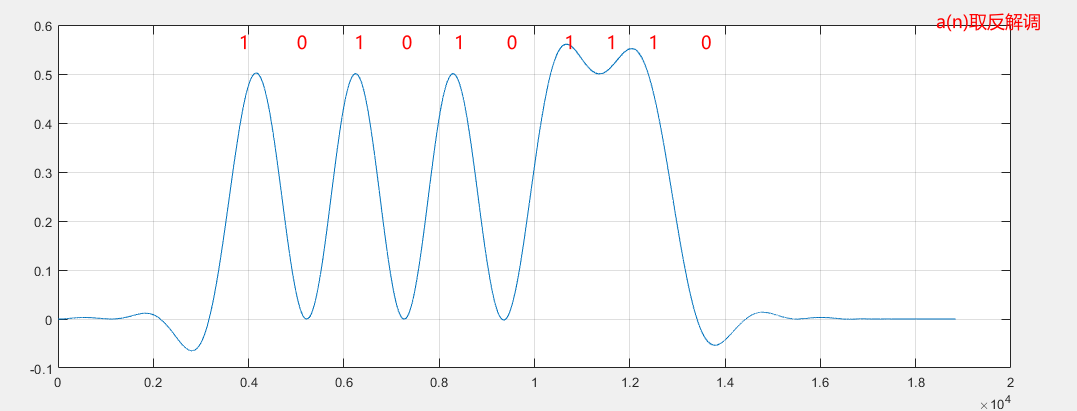
然后进行原理验证，对上述输出的IQ路信号进行载波1的正交调制后将IQ路合路，得到预期的2FSK时域信号和频谱如下，可见对0和1实现了对应载波频率的调制，从频谱上看，再两个载波127和37处出现峰值：





对上述2FSK信号两个载波分别进行相干解调加对应的低通滤波，得到a(n)和的解调波形如下，验证原理。

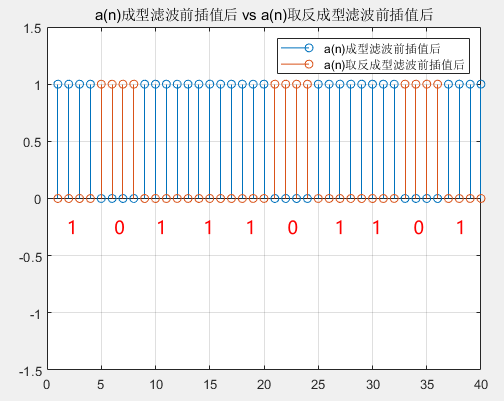




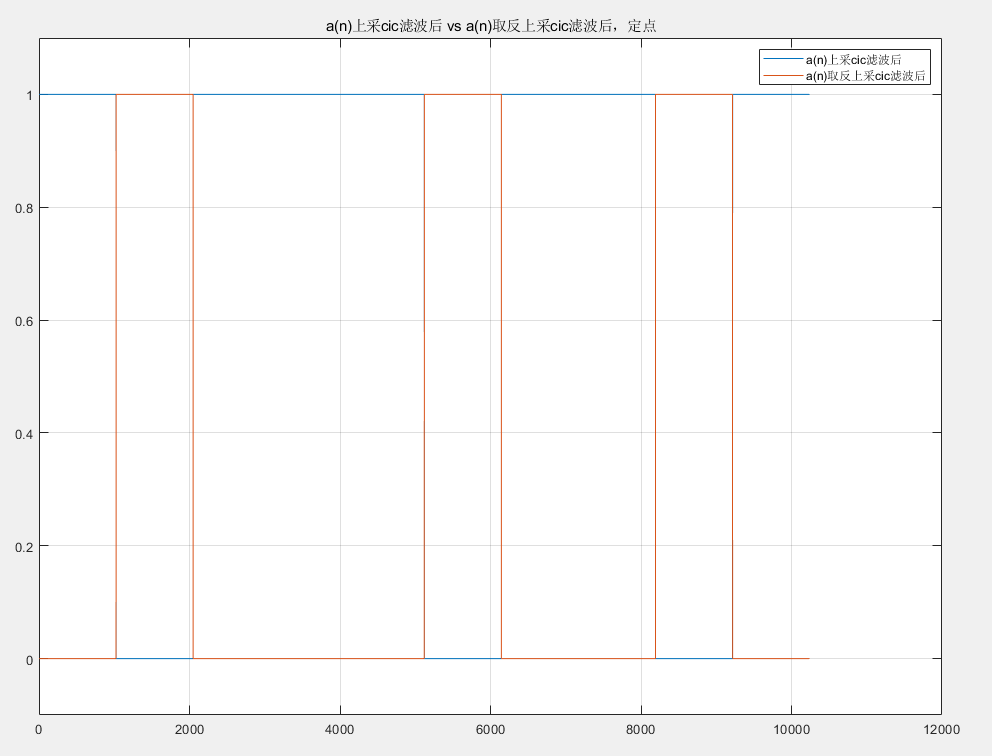
### 5.4.1 仿真2

相对于仿真1，不做成型滤波，半带滤波，cic滤波，只做与仿真1中一致的总倍数为1024的上采（两符号间插入相应个数的与第一个符号相同的值）。

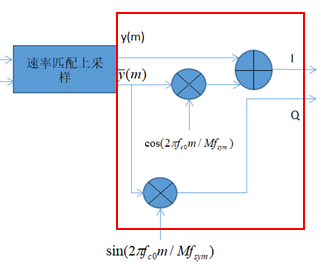
对a(n)和a(n)取反再做4倍非滤波插值之后如下，01信息比特为1 0 1 1 1 0 1 1 0 1，数据量变为原来的4倍。

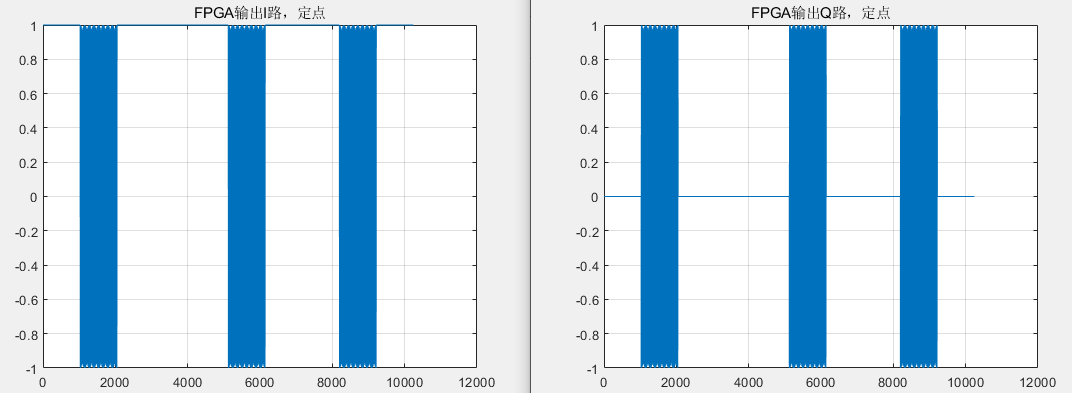


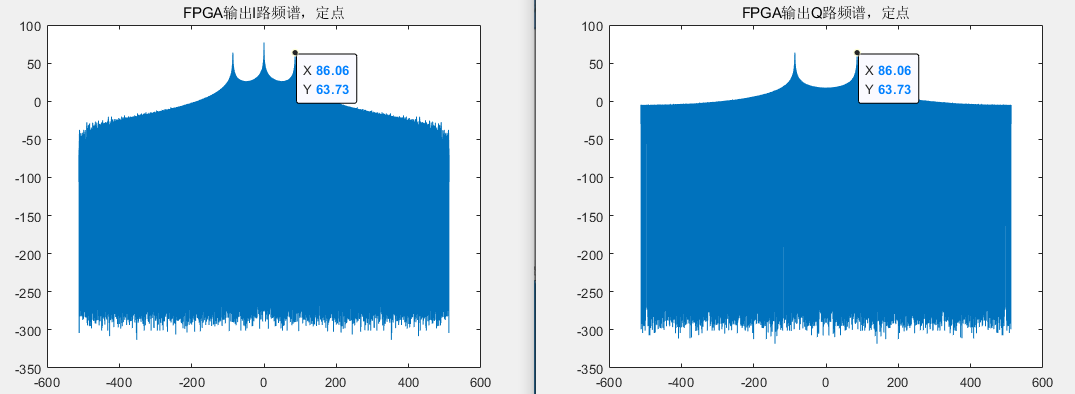
将a(n)和a(n)取反非滤波插值到总过采倍数1024倍之后如下，数据量变为原来的1024倍。



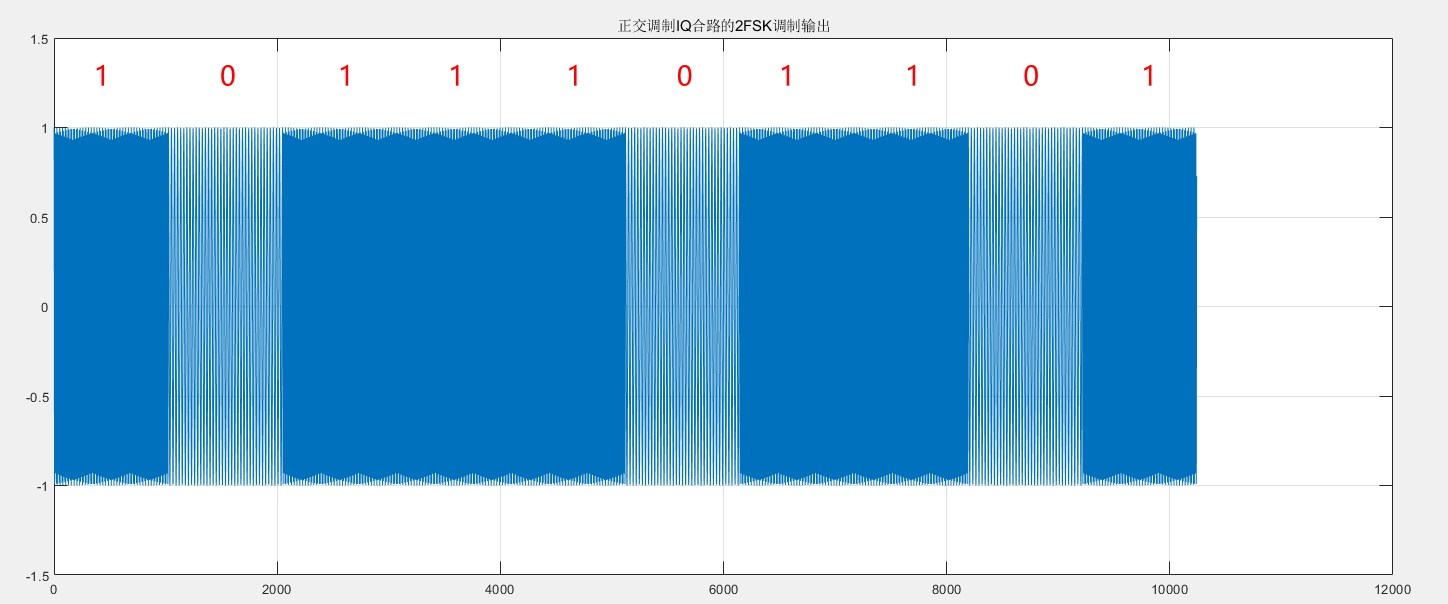
然后对上述1024倍插值输出的两路信号按照如下红框过程进行5.3中对应描述的定点处理，得到IQ路的信号和频谱分别如下：

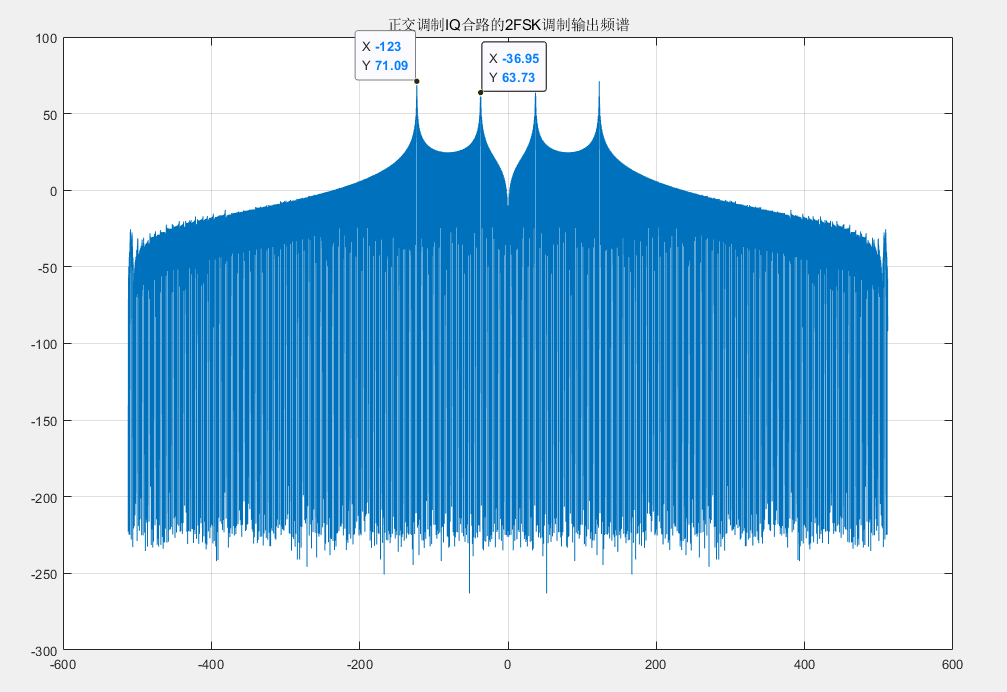




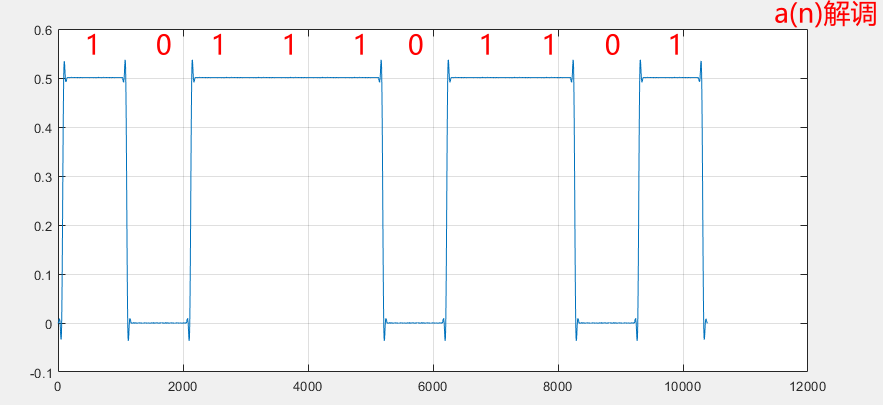


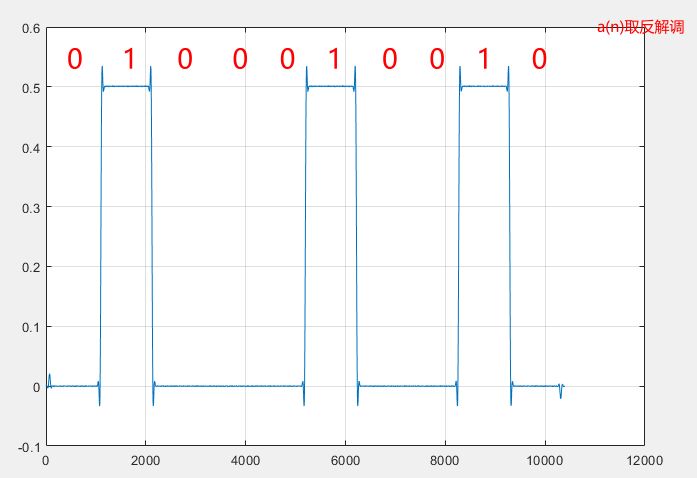
然后进行原理验证，对上述输出的IQ路信号进行载波1的正交调制后将IQ路合路，得到预期的2FSK时域信号和频谱如下，可见对0和1实现了对应载波频率的调制，从频谱上看，再两个载波127和37处出现峰值：





对上述2FSK信号两个载波分别进行相干解调加对应的低通滤波，得到a(n)和的解调波形如下，验证原理。





# 6. 基带BPSK调制

## 6.1 算法原理

BPSK信号的时域表达式为

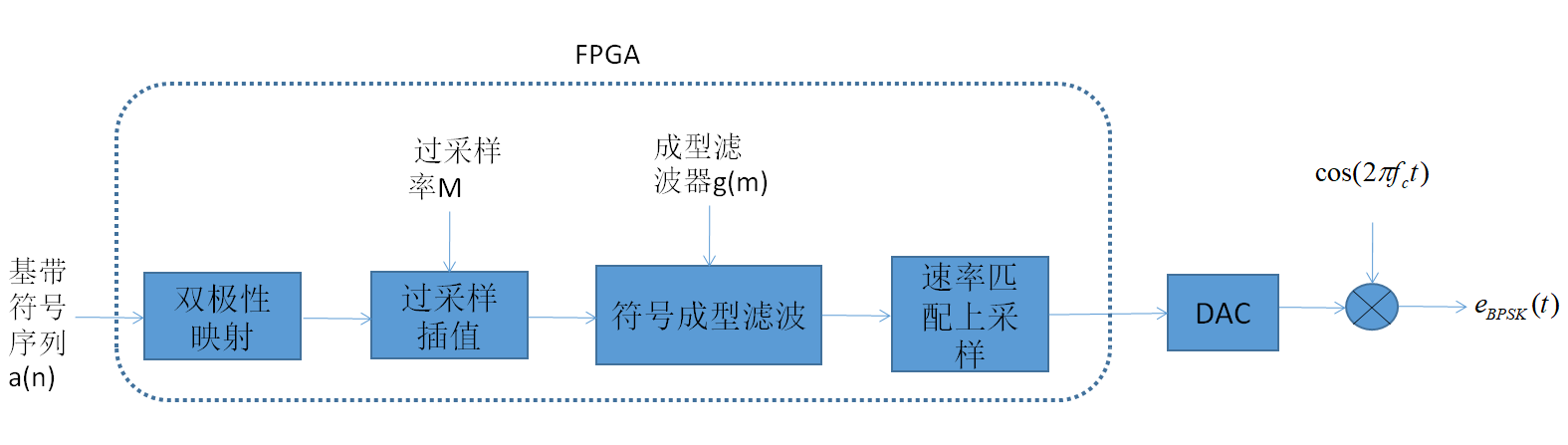
其中为双极性数字信号，是符号成型，定义同4.1，是载波。

FPGA完成部分，调制到载波由后级正交调制和混频完成。

## 6.2 参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 意义 | 取值范围 | 定点位宽 | 说明 |
|  | 129阶fir成型滤波系数 |  | (16,14) | 有软件计算好配置给FPGA，因为滤波器系数对称，这里也可以只配置一半的滤波器系数 |
|  | fir成型滤波系数阶数 |  | (9,0) | 如果滤波器系数只配置一半，阶数也只配置一半 |
|  | 脉冲成型滤波对符号的过采样插值个数 |  | (4,0) | 界面开放参数，界面配置成符号成型过采样率 |
|  | 是否进行成型滤波 |  | (1,0) | 界面开放参数，1:进行成型滤波；0：不进行成型滤波 |
|  | 成型滤波器类型 |  |  | 界面开放参数，0：升余弦滤波器，1：根升余弦滤波器，2：高斯滤波器 |
|  | 成型滤波器滚降参数 |  | 浮点 | 界面开放参数，用于软件计算成型滤波器系数， |
|  | 成型滤波器符号截断个数 |  | 浮点 | 界面开放参数，用于软件计算成型滤波器系数 |
|  | 符号率 |  | 浮点 | 界面开放参数，用于软件计算插值倍数等控制FPGA成型上采样等流程，和FPGA的时钟速率 |

## 6.3 定点算法描述

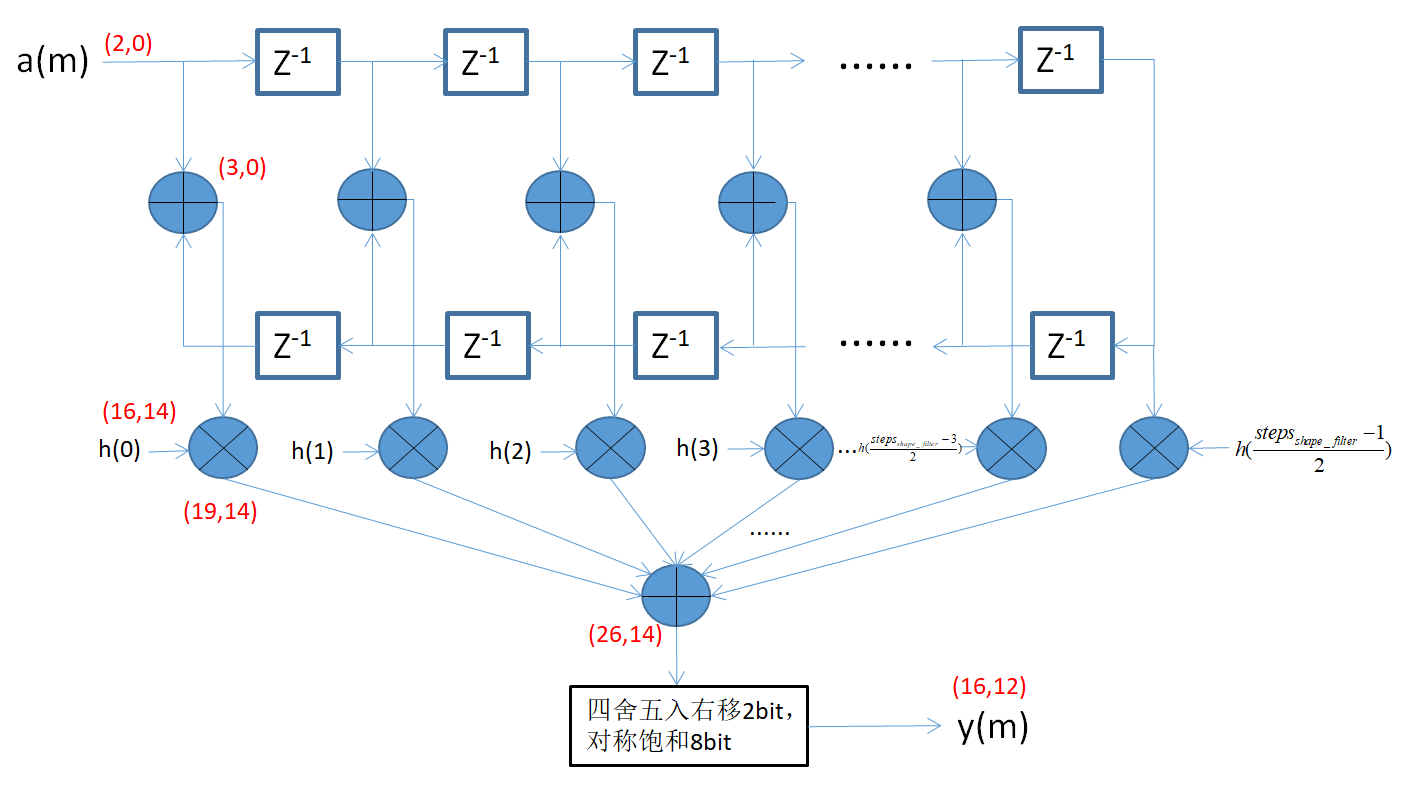


（1）根据判断是否需要进行成型滤波，若是，执行步骤（2），否则执行步骤（5）；

（2）软件配置129阶fir成型滤波系数，执行步骤（3）；

（3）对符号，先将01信息bit序列映射成双极性码，即1映射成1，0映射成-1，然后每两个符号间插入个0得到，此时样点速率提升为的+1倍，然后执行步骤（4）；

（4）对使用的前个系数进行阶的FIR成型滤波，其实现结构及各节点位宽（红色字体）如下图所示（同ASK），然后将插值滤波后的符号y(m)送给后级做速率匹配上采样（速率匹配上采样过程见对应章节描述），最后经I路由DAC进行数模转换，Q路为0。

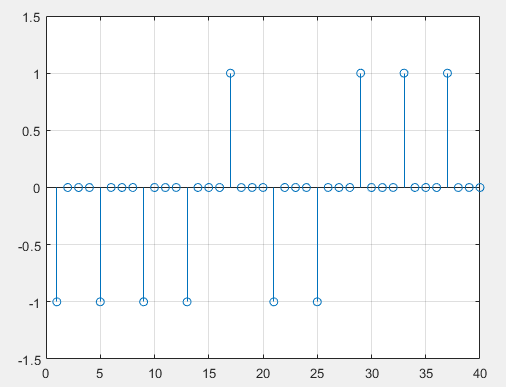


（5）对符号，每两个符号间插入个值得到，直接将a(m)送给后级做速率匹配上采样（速率匹配上采样过程见对应章节描述），Q路为0。

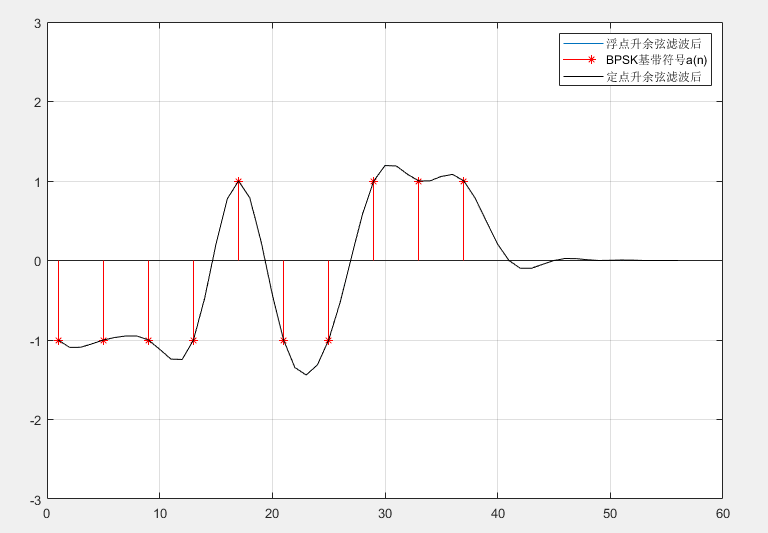
## 6.4 仿真验证

配置33阶滚降系数为0.5，=3的升余弦滤波器，对随机生成的双极性序列[-1 -1 -1 -1 1 -1 -1 1 1 1]进行本章所述处理过程的定浮点仿真对比如下。

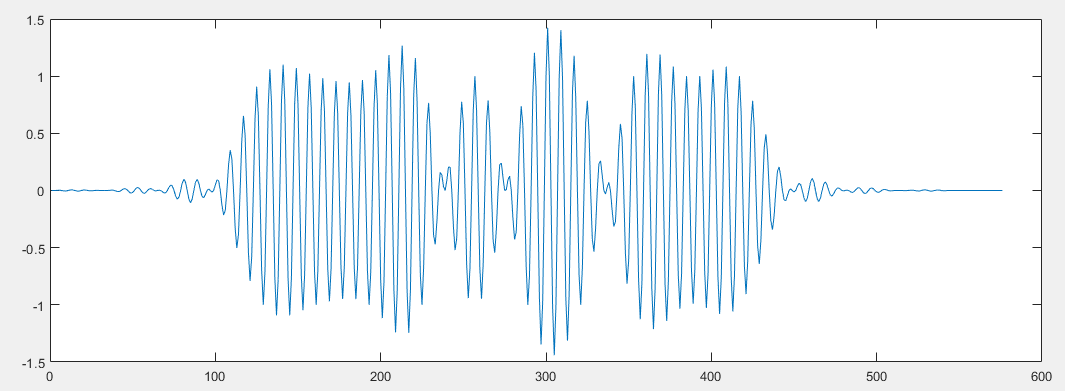
成型滤波前4倍上采插值之后的双极性序列。



成型滤波之后定浮点对比仿真结果如下，定浮点均方误差1.023\*10-4：



载波调制之后的BPSK波形：



# 7. 基带QPSK调制

## 7.1 算法原理

QPSK表达式为

其中一种的配置如下

是符号成型，定义同4.1中对应描述，是载波，是符号周期，A是符号幅度。

将QPSK表达式展开

令

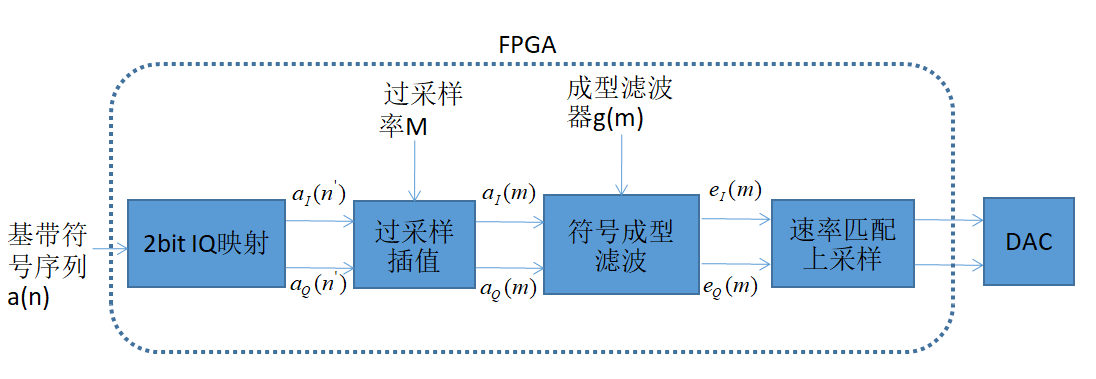
则

和即为FPGA生成的正交调制IQ路基带信号。其中各种二进制码组合对应的和由软件配置。和的载波正交调制由后级IQ调制和混频完成。

## 7.2 参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 意义 | 取值范围 | 定点位宽 | 说明 |
|  | 129阶fir成型滤波系数 |  | (16,14) | 有软件计算好配置给FPGA，因为滤波器系数对称，这里也可以只配置一半的滤波器系数 |
|  | fir成型滤波系数阶数 |  | (9,0) | 如果滤波器系数只配置一半，阶数也只配置一半 |
|  | 脉冲成型滤波对符号的过采样插值个数 |  | (4,0) | 界面开放参数，界面配置成符号成型过采样率 |
|  | 最大支持256QAM的IQ路符号真值表I路 |  | (16,14) | 星座点IQ路符号真值支持任意配置，QPSK软件按最大浮点值1配置 |
|  | 最大支持256QAM的IQ路符号真值表Q路 |  | (16,14) | 星座点IQ路符号真值支持任意配置，QPSK软件按最大浮点值1配置 |
|  | 是否进行成型滤波 |  | (1,0) | 界面开放参数，1:进行成型滤波；0：不进行成型滤波 |
|  | 成型滤波器类型 |  |  | 界面开放参数，0：升余弦滤波器，1：根升余弦滤波器，2：高斯滤波器 |
|  | 成型滤波器滚降参数 |  | 浮点 | 界面开放参数，用于软件计算成型滤波器系数， |
|  | 成型滤波器符号截断个数 |  | 浮点 | 界面开放参数，用于软件计算成型滤波器系数 |
|  | 符号率 |  | 浮点 | 界面开放参数，用于软件计算插值倍数等控制FPGA成型上采样等流程，和FPGA的时钟速率 |

## 7.3 定点算法描述

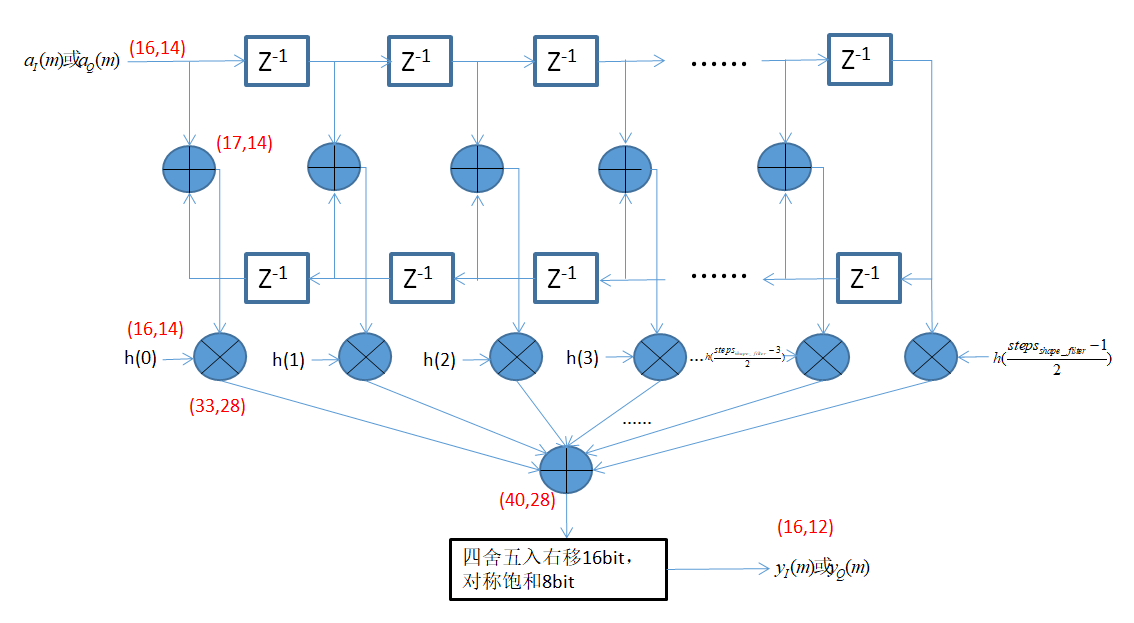


（1）根据判断是否需要进行成型滤波，若是，执行步骤（2），否则执行步骤（5）；

（2）软件配置129阶fir成型滤波系数，执行步骤（3）；

（3）对符号，每2bit序列通过查找和真值表映射成IQ符号和，然后每两个符号间插入个0得到，此时样点速率提升为和的+1倍，然后执行步骤（4）；

（4）对（3）的结果和分别使用的前个系数进行阶的FIR成型滤波，和位宽为(16,14)，其实现结构及各节点位宽（红色字体）如下图所示，然后将插值成型滤波后的IQ路符号和送给后级做速率匹配上采样（速率匹配上采样过程见对应章节描述），最后经IQ路由DAC进行数模转换。



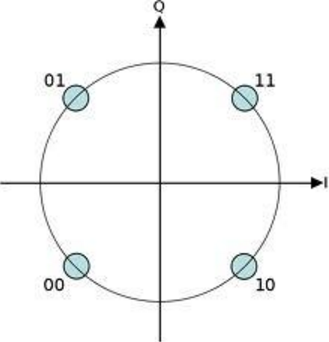
（5）对符号，每2bit序列通过查找和真值表映射成IQ符号和，每两个符号间插入个与和相同的值得到和，直接将和送给后级做速率匹配上采样（速率匹配上采样过程见对应章节描述）。

## 7.4 仿真验证

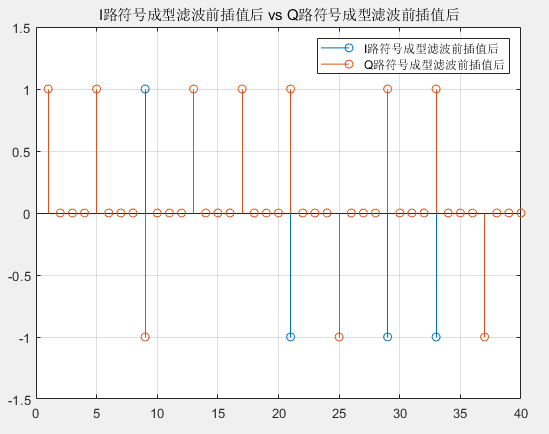
随机生成20位01bit序列，映射成10位QPSK IQ符号，符号速率为1，做4倍插值，利用滚降系数0.5，符号截断个数为8的定点升余弦滤波器进行IQ符号成型滤波，进行速率匹配上采样，这里设置为4级2倍上采半带滤波，然后做1024倍插值CIC滤波，此时采样率为65536。上述过程为定点仿真，模拟FPGA定点处理。

然后继续进行原理性验证，对上述IQ路数据做载波为2048的IQ正交调制后合路，以上为调制过程。

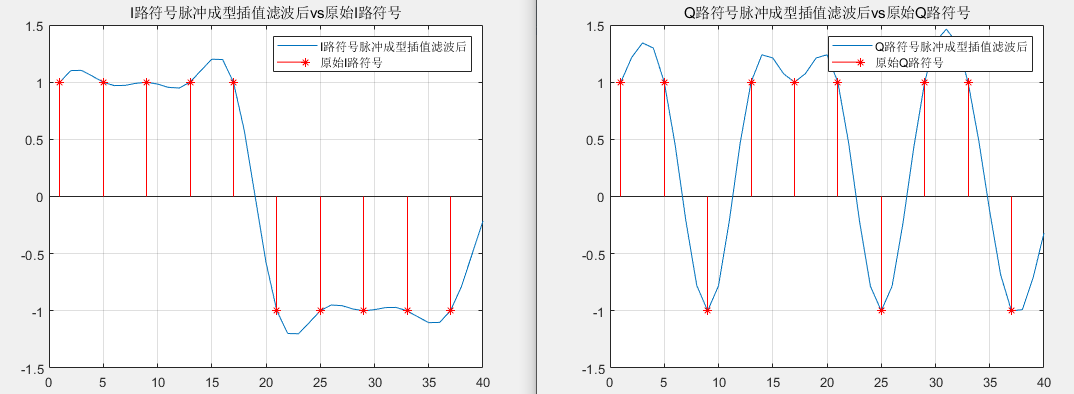
再加入-40dB的噪声，然后是解调过程，使用理想载波同步的IQ本振进行正交下变频变为IQ两路，使用低通滤波，得到IQ解调波形，最后进行理想符号提取，得到解调后的IQ路符号和星座图。



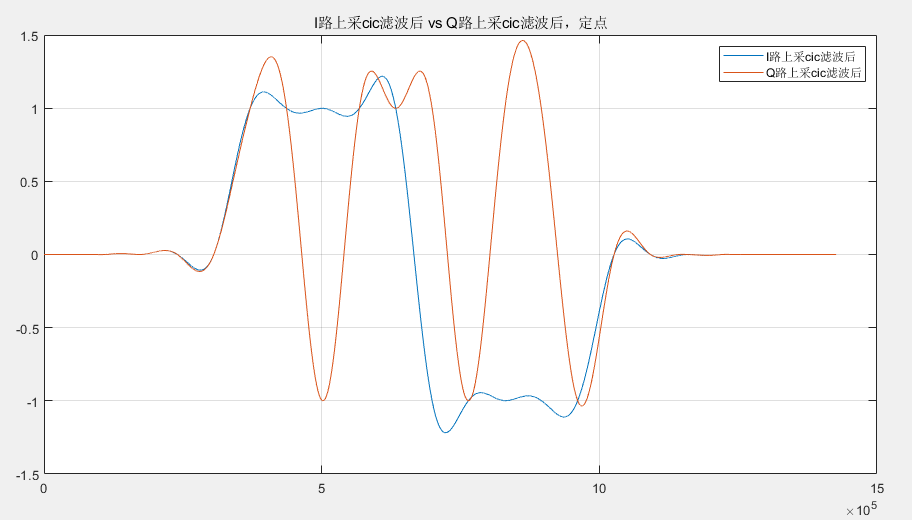
随机生成的20位01bit序列为[1，1，1，1，1 ，0，1，1，1，1，0，1，0，0，0，1，0，1，0，0]，将其映射成为10位IQ路符号，映射关系如上，并进行4倍插值。



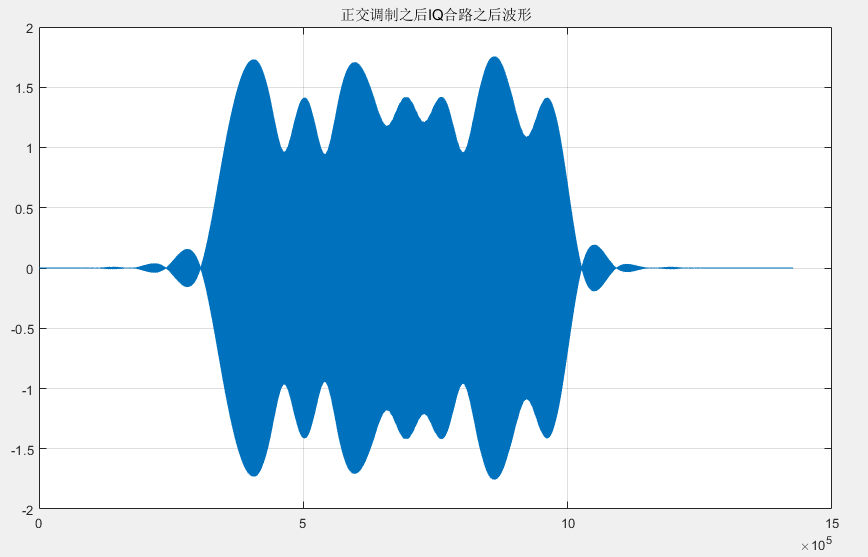
然后进行符号成型滤波。



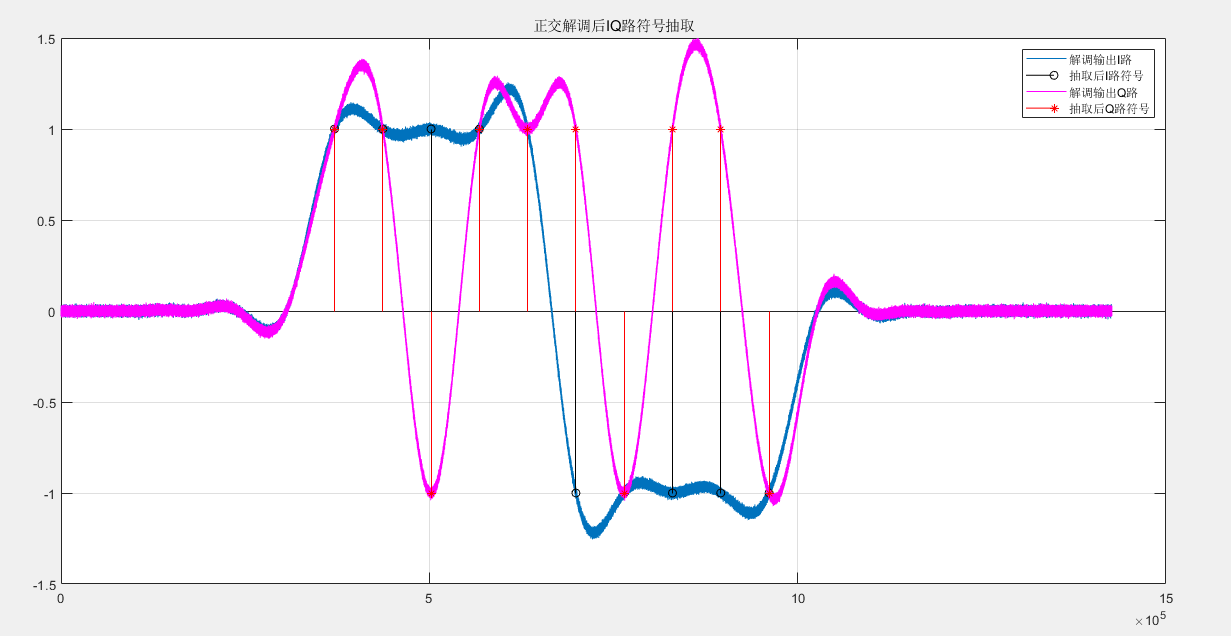
经过速率匹配上采样：4级2倍上采半带滤波及1024倍上采CIC滤波后。

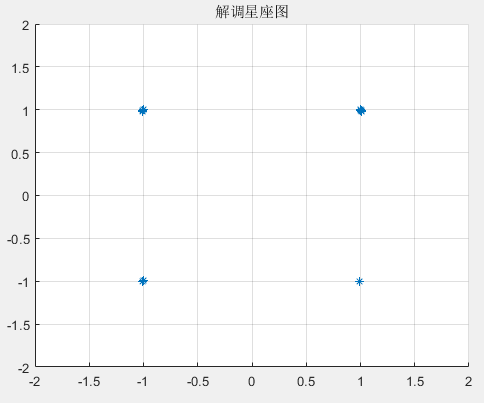


然后仿真QPSK调制解调原理，首先进行载波为2048Hz的正交调制并IQ合路。



然后对加噪的QPSK正交调制信号，利用理想载波同步和符号同步，进行解调，并抽取为IQ符号，如下抽取的QPSK IQ符号为[11,11,1-1,11,11,-11,-1-1,-11,-11,-1-1]，反映射为01bit序列为[11,11,10,11,11,01,00,01,01,00]，与仿真发送的原始01bit序列一致。





# 8. 基带16QAM调制

## 8.1 算法原理

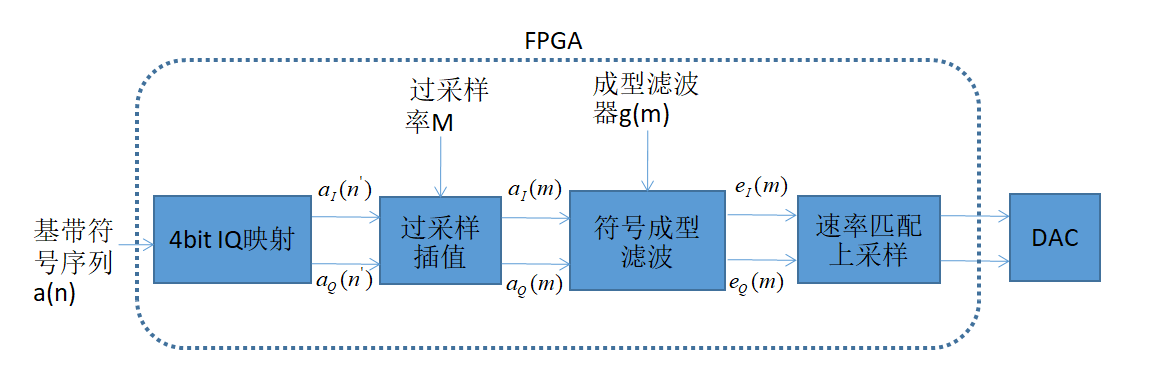
16QAM基带调制过后的每一个符号可以表示0000,0001,0010,0011,0100,0101,0110,0111，1000,1001,1010,1011,1100,1101,1110,1111共16种4bit基带信息，其IQ路正交调制形式表达式为

其中与可根据每4bit 01基带信息bit取值按如下星座图映射成IQ路16QAM符号。为符号成型滤波器，其定义同4.1节描述

## 8.2 参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 意义 | 取值范围 | 定点位宽 | 说明 |
|  | 129阶fir成型滤波系数 |  | (16,14) | 有软件计算好配置给FPGA，因为滤波器系数对称，这里也可以只配置一半的滤波器系数 |
|  | fir成型滤波系数阶数 |  | (9,0) | 如果滤波器系数只配置一半，阶数也只配置一半 |
|  | 脉冲成型滤波对符号的过采样插值个数 |  | (4,0) | 界面开放参数，界面配置成符号成型过采样率 |
|  | 是否进行成型滤波 |  | (1,0) | 界面开放参数，1:进行成型滤波；0：不进行成型滤波 |
|  | 成型滤波器类型 |  |  | 界面开放参数，0：升余弦滤波器，1：根升余弦滤波器，2：高斯滤波器 |
|  | 最大支持256QAM的IQ路符号真值表I路 |  | (16,14) | 星座点IQ路符号真值支持任意配置，16QAM软件按最大浮点值1.5配置 |
|  | 最大支持256QAM的IQ路符号真值表Q路 |  | (16,14) | 星座点IQ路符号真值支持任意配置，16QAM软件按最大浮点值1.5配置 |
|  | 成型滤波器滚降参数 |  | 浮点 | 界面开放参数，用于软件计算成型滤波器系数， |
|  | 成型滤波器符号截断个数 |  | 浮点 | 界面开放参数，用于软件计算成型滤波器系数 |
|  | 符号率 |  | 浮点 | 界面开放参数，用于软件计算插值倍数等控制FPGA成型上采样等流程，和FPGA的时钟速率 |

## 8.3 定点算法描述



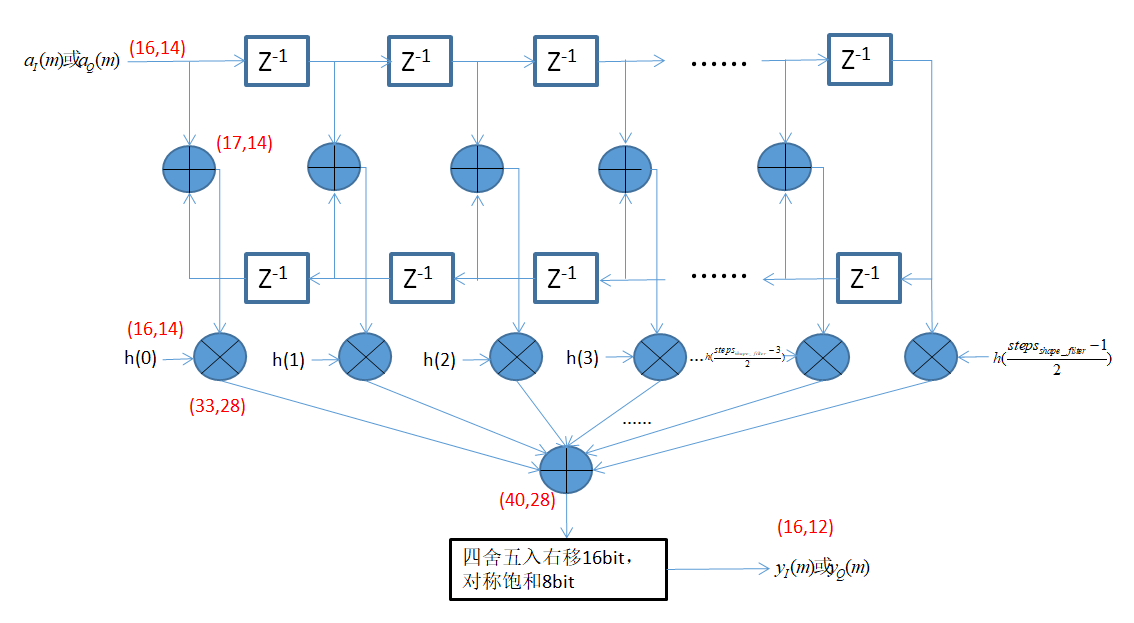
16QAM除原始01信息bit序列映射为IQ路符号与QPSK不同外，其余过程基本和QPSK一致。

（1）根据判断是否需要进行成型滤波，若是，执行步骤（2），否则执行步骤（5）；

（2）软件配置129阶fir成型滤波系数，执行步骤（3）；

（3）对符号，每4bit序列通过查找和真值表映射成IQ符号和，然后每两个符号间插入个0得到，此时样点速率提升为和的+1倍，然后执行步骤（4）；

（4）对（3）的结果和分别使用的前个系数进行阶的FIR成型滤波，和位宽为(16,14)，其实现结构及各节点位宽（红色字体）如下图所示，然后将插值成型滤波后的IQ路符号和送给后级做速率匹配上采样（速率匹配上采样过程见对应章节描述），最后经IQ路由DAC进行数模转换。



（5）对符号，每4bit序列通过查找和真值表映射成IQ符号和，每两个符号间插入个与和相同的值得到和，直接将和送给后级做速率匹配上采样（速率匹配上采样过程见对应章节描述）。

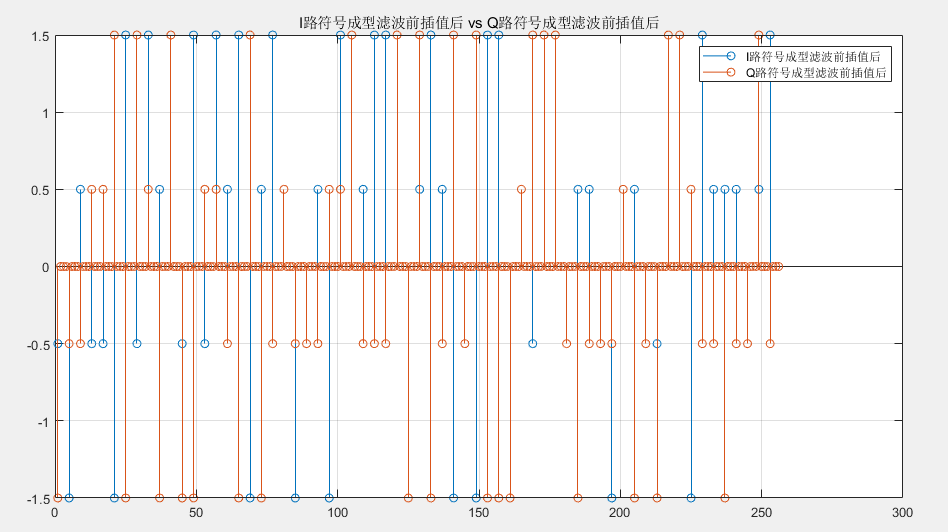
## 8.4 仿真验证

随机生成256位01bit序列，映射成64位16QAM IQ符号，符号速率为1，做4倍插值，利用滚降系数0.5，符号截断个数为8的定点升余弦滤波器进行IQ符号成型滤波，进行速率匹配上采样，这里设置为4级2倍上采半带滤波，然后做1024倍插值CIC滤波，此时采样率为65536。上述过程为定点仿真，模拟FPGA定点处理。

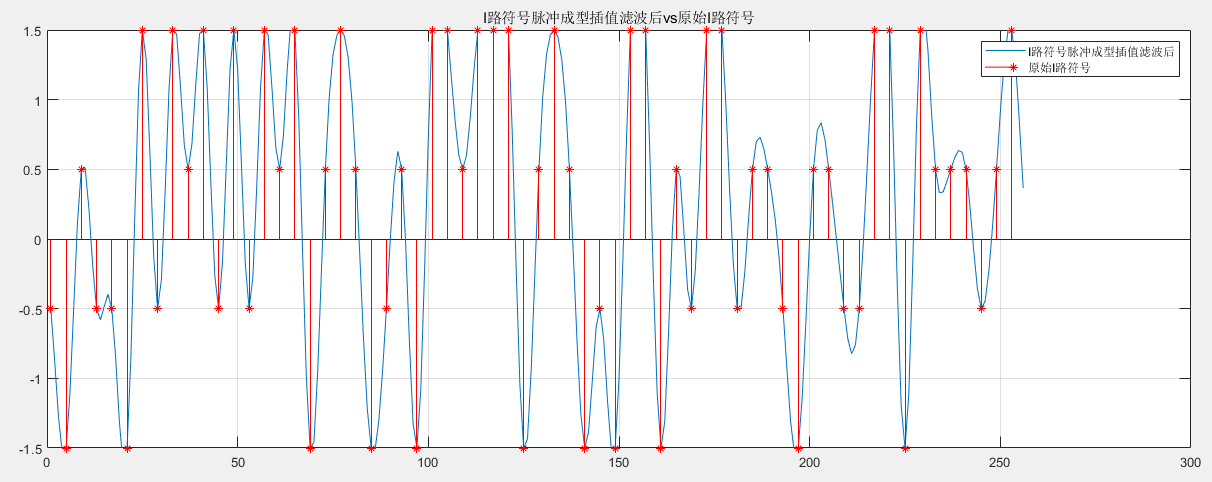
然后继续进行原理性验证，对上述IQ路数据做载波为2048的IQ正交调制后合路，以上为调制过程。

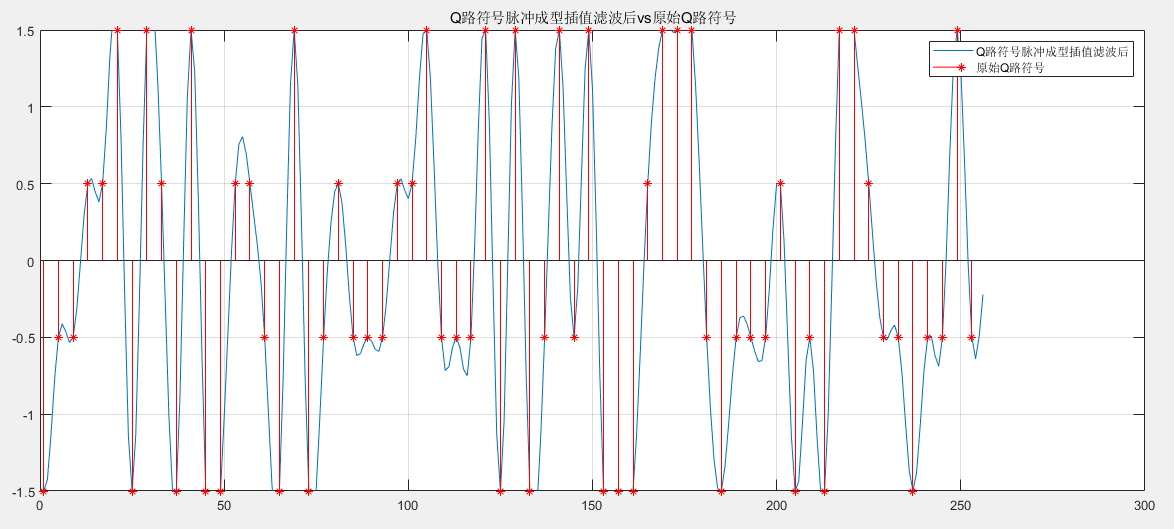
再加入-40dB的噪声，然后是解调过程，使用理想载波同步的IQ本振进行正交下变频变为IQ两路，使用低通滤波，得到IQ解调波形，最后进行理想符号提取，得到解调后的IQ路符号和星座图。

随机生成的256位01bit序列为[1,0,1,1,1,1,1,0,0,0,1,0,1,0,0,0,1,0,0,0,1,1,0,1,0,1,1,1,1,0,0,1,0,1,0,0,0,0,1,1,0,1,0,1,1,0,1,1,0,1,1,1,1,0,0,0,0,1,0,0,0,0,1,0,0,1,1,1,1,1,0,1,0,0,1,1,0,1,1,0,0,0,0,0,1,1,1,0,1,0,1,0,0,0,1,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,1,0,0,1,1,0,0,1,1,0,0,1,0,1,1,1,1,1,0,0,0,1,0,1,1,1,0,0,1,0,1,1,0,1,1,0,1,0,1,1,0,1,0,1,1,1,0,1,1,1,1,1,1,1,0,0,0,0,1,0,0,1,0,1,0,1,0,1,0,1,1,0,1,0,0,0,1,1,0,0,1,0,1,0,1,0,1,1,1,0,0,0,0,0,0,0,1,1,1,0,1,0,1,0,1,1,0,1,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,1,0,1,0,1,0,0,0,0,1,0,1,1,0]，将其映射成为64位IQ路符号，并进行4倍插值。

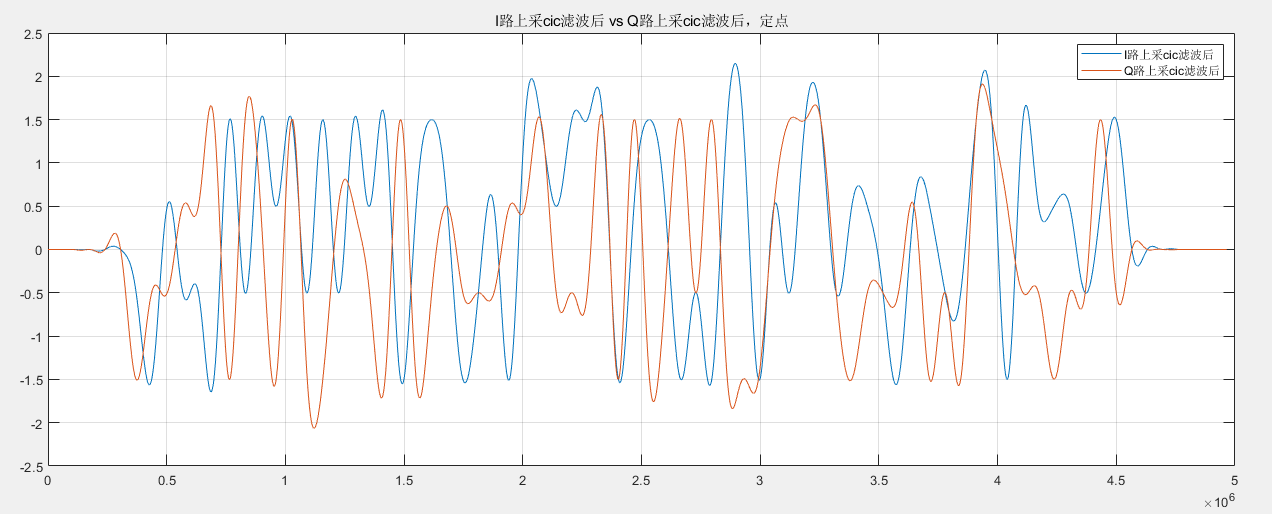


然后进行符号成型滤波，可以看到相对于QPSK，多了幅度为±0.5的符号。

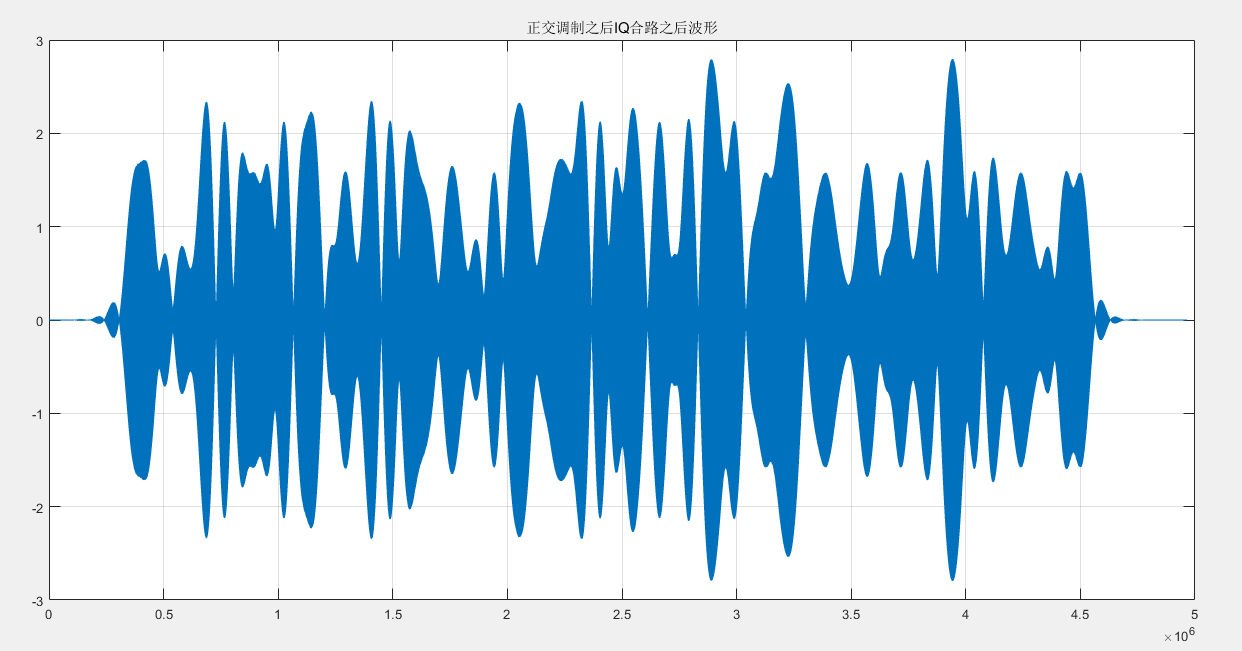




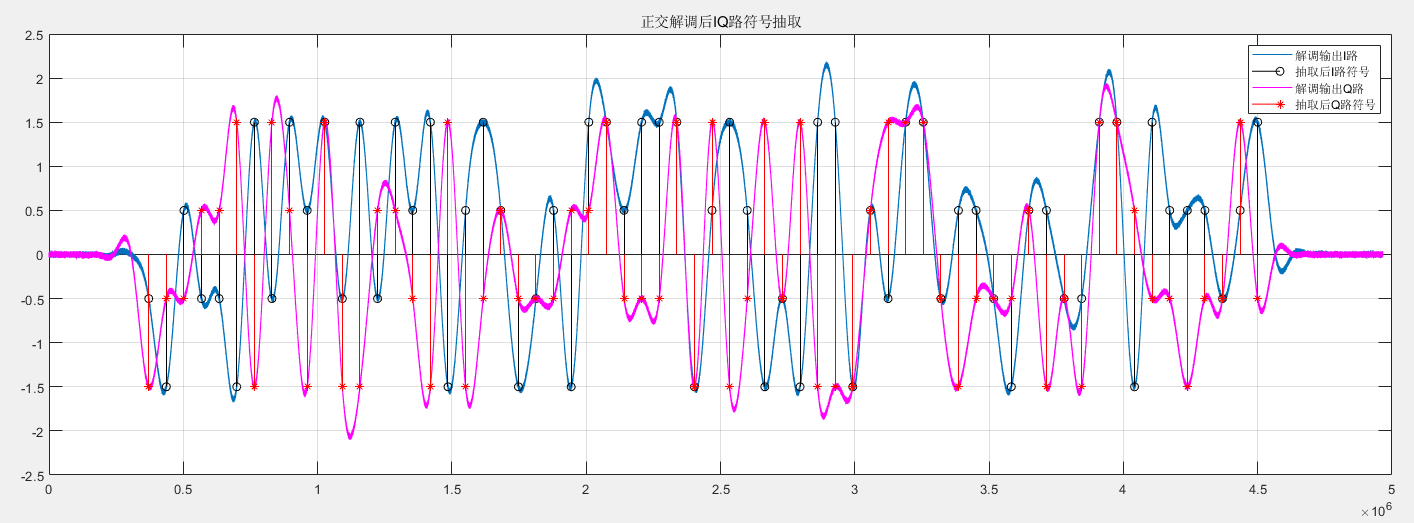
经过速率匹配上采样：4级2倍上采半带滤波及1024倍上采CIC滤波后。

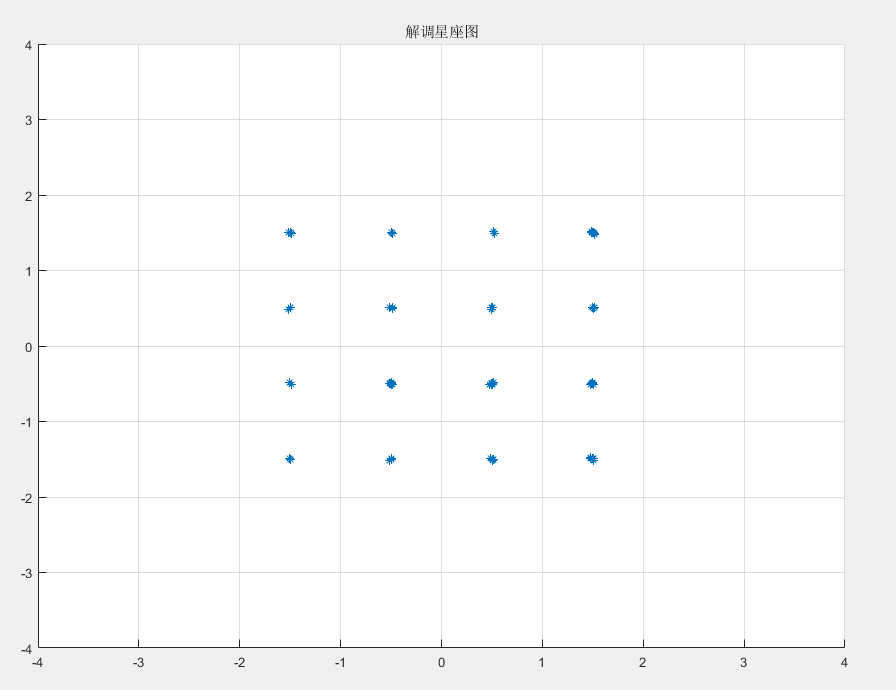


然后仿真QPSK调制解调原理，首先进行载波为2048Hz的正交调制并IQ合路。



然后对加噪的16QAM正交调制信号，利用理想载波同步和符号同步，进行解调，并抽取为IQ符号，如下抽取的16QAM IQ符号（IQIQ排列）为[-0.4945 -1.4869 -1.4861 -0.5059 0.5037 -0.5039 -0.4901 0.5112 -0.5130 0.5103 -1.4831 1.4891 1.5103 -1.5160 -0.4925 1.5033 1.5178 0.5058 0.4904 -1.4867 1.4891 1.5061 -0.5112 -1.5161 1.4992 -1.4926 -0.4888 0.4951 1.5011 0.5114 0.5151 -0.4859 1.4811 -1.4956 -1.4922 1.5041 0.5075 -1.5168 1.4789 -0.5146 0.4920 0.4844 -1.5060 -0.4894 -0.4990 -0.4861 0.4985 -0.5130 -1.4937 0.5119 1.5055 0.5003 1.5045 1.4997 0.4942 -0.4961 1.4979 -0.4870 1.5012 -0.4949 1.4882 1.5111 -1.4920 -1.5000 0.5129 1.5183 1.5076 -1.5140 0.5030 -0.4961 -1.4975 1.5043 -0.5037 -0.4990 -1.5155 1.5056 1.5020 -1.4922 1.4819 -1.4763 -1.5021 -1.4954 0.4969 0.5060 -0.4885 1.4943 1.4991 1.5055 1.5118 1.4816 -0.4914 -0.5109 0.5099 -1.5086 0.4994 -0.4902 -0.4817 -0.5152 -1.4914 -0.5054 0.5031 0.5053 0.5140 -1.5028 -0.4996 -0.4998 -0.4954 -1.5039 1.5124 1.4908 1.4992 1.5069 -1.5177 0.4888 1.5067 -0.5097 0.5060 -0.4978 0.5046 -1.4977 0.4755 -0.5067 -0.5110 -0.4916 0.5210 1.4879 1.4890 -0.4925]，反映射为01bit序列为[1,0,1,1,1,1,1,0,0,0,1,0,1,0,0,0,1,0,0,0,1,1,0,1,0,1,1,1,1,0,0,1,0,1,0,0,0,0,1,1,0,1,0,1,1,0,1,1,0,1,1,1,1,0,0,0,0,1,0,0,0,0,1,0,0,1,1,1,1,1,0,1,0,0,1,1,0,1,1,0,0,0,0,0,1,1,1,0,1,0,1,0,0,0,1,0,1,1,0,0,0,1,0,0,0,1,0,1,0,0,1,0,0,1,1,0,0,1,1,0,0,1,0,1,1,1,1,1,0,0,0,1,0,1,1,1,0,0,1,0,1,1,0,1,1,0,1,0,1,1,0,1,0,1,1,1,0,1,1,1,1,1,1,1,0,0,0,0,1,0,0,1,0,1,0,1,0,1,0,1,1,0,1,0,0,0,1,1,0,0,1,0,1,0,1,0,1,1,1,0,0,0,0,0,0,0,1,1,1,0,1,0,1,0,1,1,0,1,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,1,0,1,0,1,0,0,0,0,1,0,1,1,0]，与仿真发送的原始01bit序列一致。





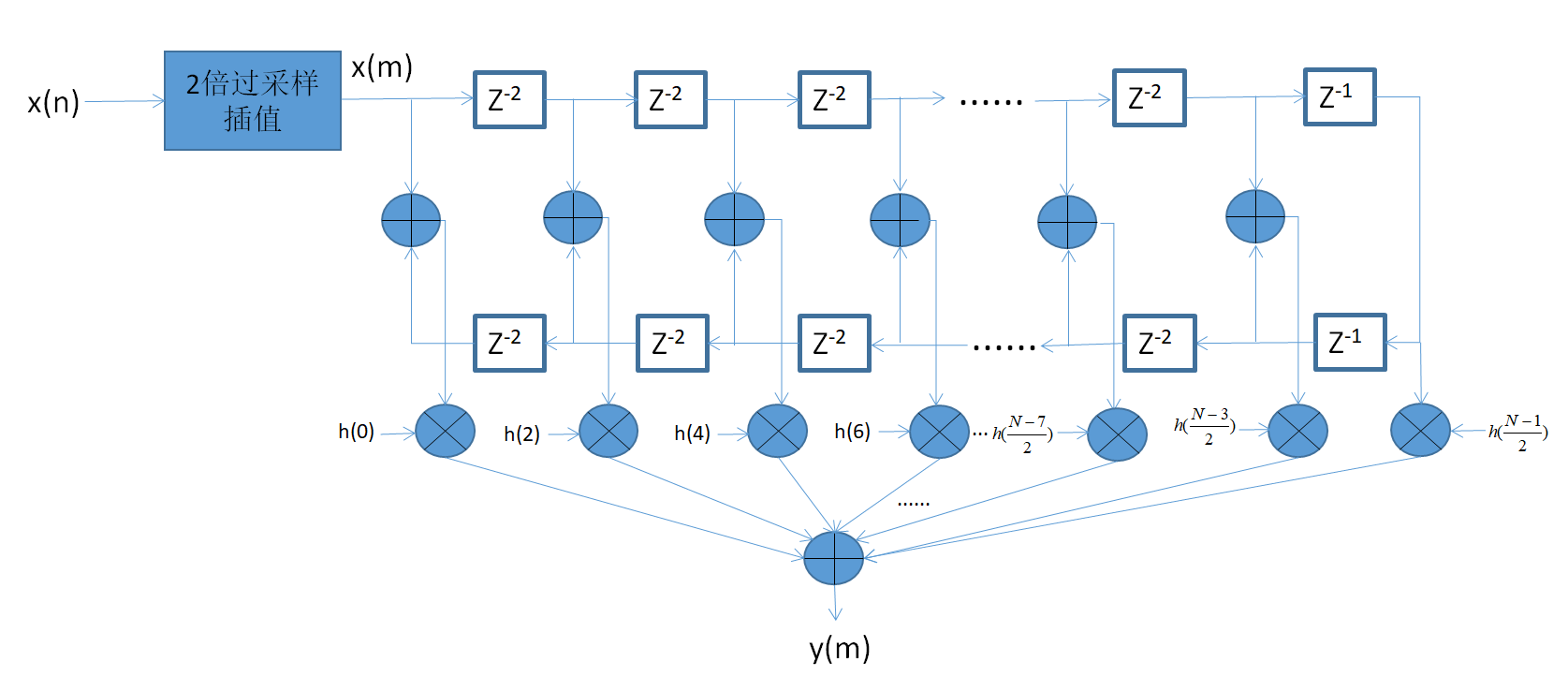
# 9. 速率匹配上采样

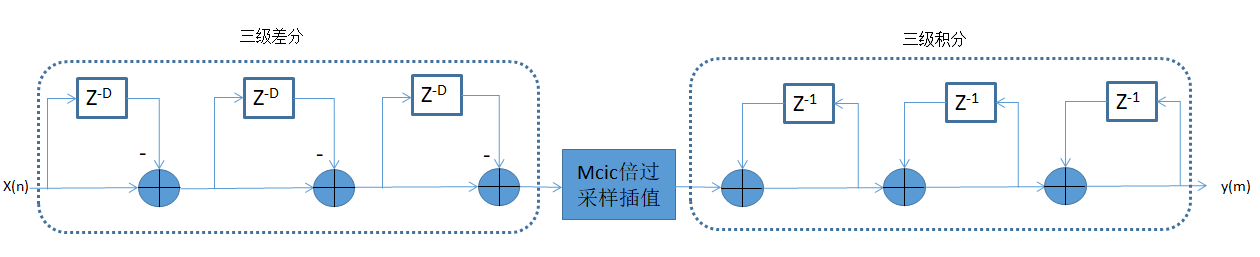
## 9.1 算法原理

速率匹配上采样是将低采样率的I/Q路信号通过不同级数的2倍上采+半带滤波，及不同插值倍数的上采+CIC滤波组合，使I/Q速率提升到需要的输出速率上，并实现对插值带来的频谱镜像的抑制。

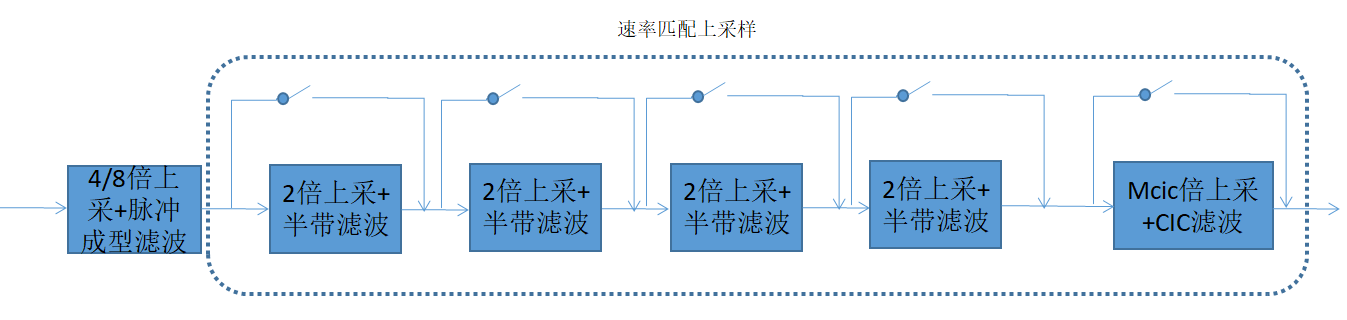
假设2倍上采+半带滤波级数为，CIC滤波的上采倍数为，则速率匹配总上采样倍数为。

半带滤波器是一种特殊的FIR滤波器，假设奇数阶半带滤波器系数为h(n)，n=0~N-1，其系数除了中心（n=）和距离中心的奇数阶系数以外，其余系数均为0。其结构如下图：

 倍上采样CIC滤波器包含一个M级的差分器，插值器，及一个M级的积分器。一个带3级差分/积分器的上采CIC滤波器结构如下。



在本项目中，速率匹配上采样采用4级2倍上采半带滤波+一级任意倍上采CIC滤波，且各级滤波器开关可控的方案，半带滤波器阶数支持19阶，CIC滤波差分/积分器级数3级。如下。



## 9.2 参数

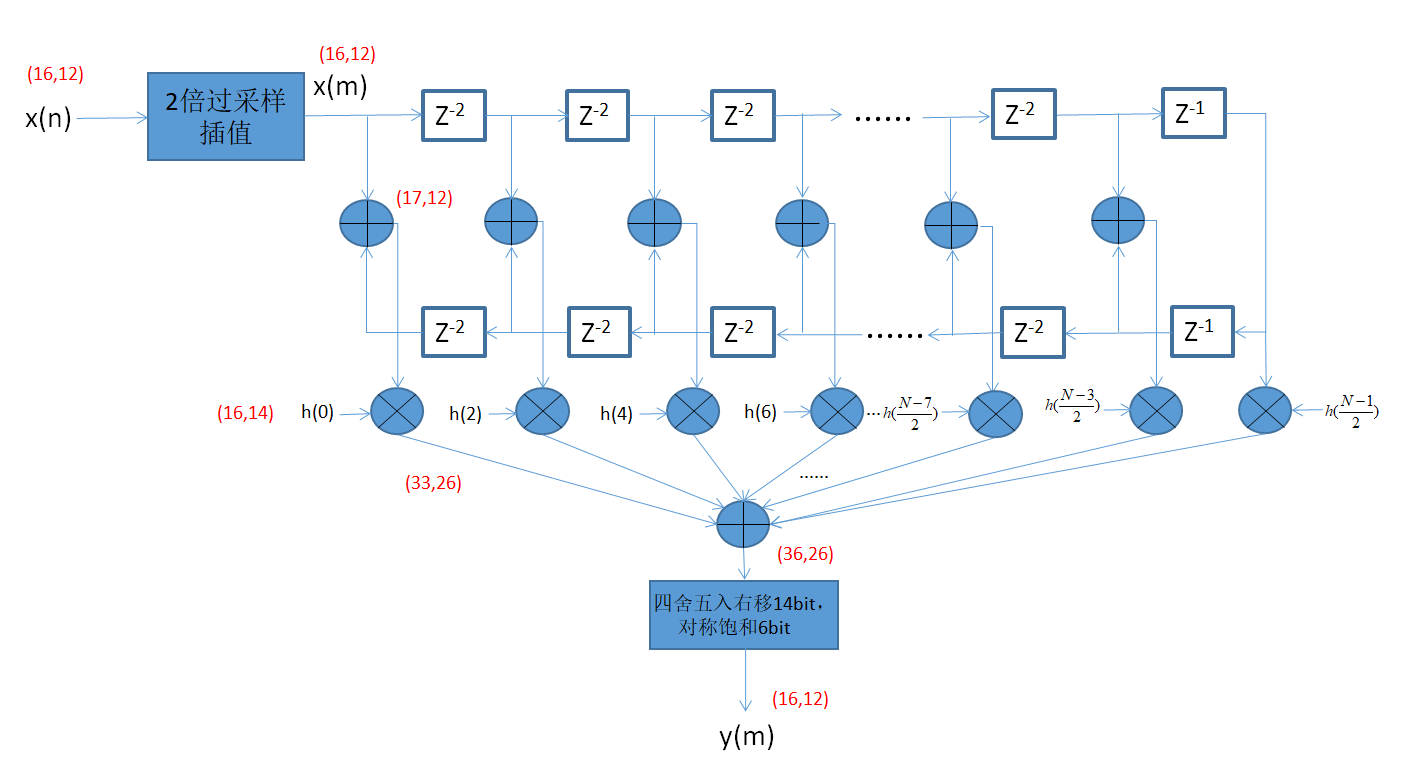
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 意义 | 取值范围 | 定点位宽 | 说明 |
|  | 19阶半带滤波器6阶配置系数 |  | (16,14) | 软件计算好配置给FPGA，因为滤波器系数对称，且具有间隔为0的特点，只需配置6个系数 |
|  | 4级半带滤波功能开关 |  | (1,0) |  |
|  | 4级半带滤波2倍插值开关 |  | (1,0) |  |
|  | CIC滤波器插值倍数 |  | (16,0) |  |
|  | CIC滤波功能开关 |  | (1,0) |  |
|  | CIC插值功能功能开关 |  | (1,0) |  |
|  | CIC插值滤波输出四舍五入右移bit数 |  | (8,0) | 微分积分固定为3级之后，软件计算方法为ceil(log2()) |

## 9.3 定点算法描述

（1）根据，，判断是否有半带滤波2倍上采样使能，如果有，转到步骤（2），否则转到步骤（5）；

（2）判断（1）中n对应的，如果对应的第n级半带滤波器使能，则转到步骤（3），否则转到步骤（4）；

（3）每一级2倍上采半带滤波，输入数据位宽为(16,12)，对其每两个采样点之间插一个0，速率提升为输入采样率的2倍，然后根据配置的半带滤波器系数，进行如下过程和位宽的半带滤波，输出位宽仍为(16,12)，如果n==3转到步骤（5），否则转到步骤（1）；

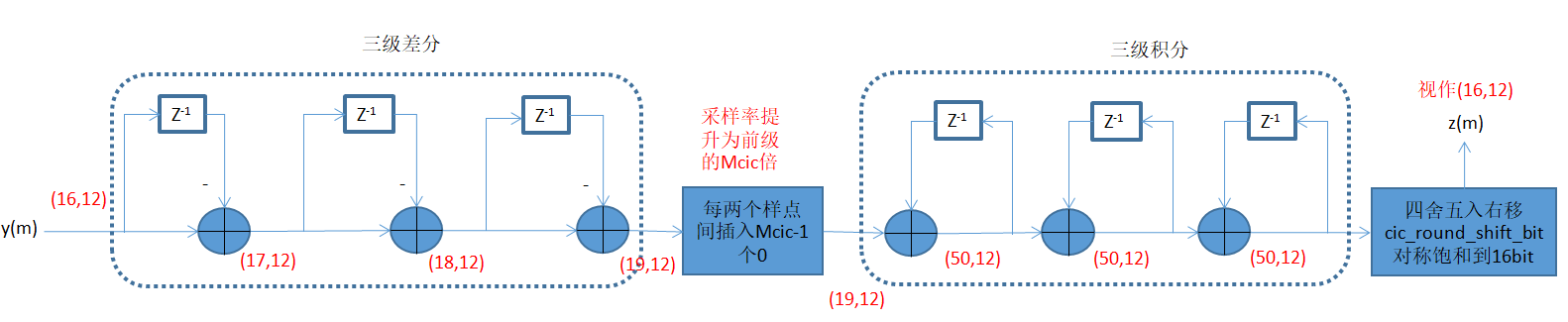


（4）每一级2倍上采，输入数据位宽为(16,12)，对其每两个采样点之间插一个与第一个符号相同的值，输入位宽仍为(16,12)，如果n==3转到步骤（5），否则转到步骤（1）；

（5）判断，如果CIC插值使能，转到步骤（6），否则转到步骤（9）；

（6）判断，如果CIC滤波使能，转到步骤（7），否则转到步骤（8）；

（7）根据配置的参数及，按照下图的CIC插值滤波过程和位宽处理，输出采样率为输入的倍，输出位宽为(16,12)，跳转到步骤（9）；



（8）输入数据位宽为(16,12)，根据参数，进行倍上采，对其每两个采样点之间插个与第一个样点相同的值，采样率提升为输入的倍，输出位宽仍为（16,12），跳转到步骤（9）；

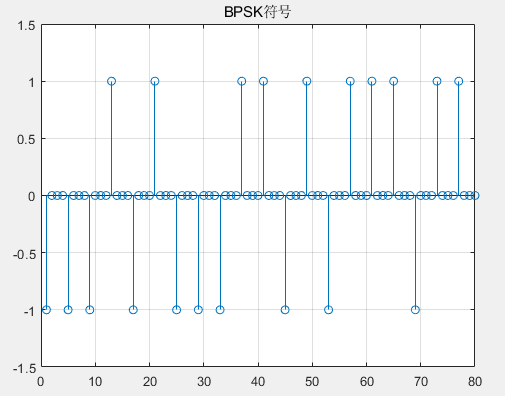
（9）将（8）的数据（I/Q路或者只有其中一路）送入后级IQ幅相不一致校正。

## 9.4 仿真验证

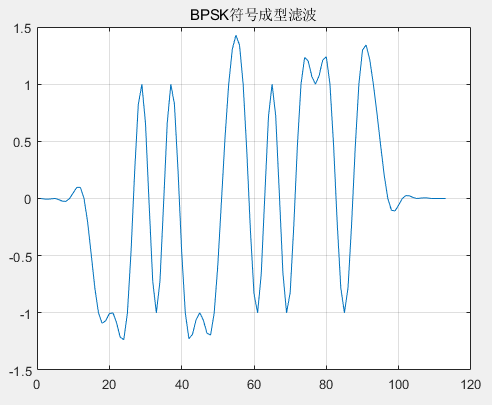
### 9.4.1 仿真1

随机生成20位01bit信息序列，BPSK符号映射，4倍过采样0.5滚降系数升余弦滤波器成型，成型滤波器符号截断个数8；4级2倍上采半带滤波，原始15阶浮点半带滤波器系数[-0.00821845834704578,0,0.0429455611270786,0,-0.146779725449167,0,0.611712579457848,1,0.611712579457848,0,-0.146779725449167,0,0.0429455611270786,0,-0.00821845834704578]，CIC滤波上采倍数为8192，根据计算得到配置为26，各级插值和滤波功能全部打开，总符号上采倍数为524288倍，仿真对比4级2倍上采半带滤波输出，及CIC上采滤波输出，过程全定点与全浮点对比。

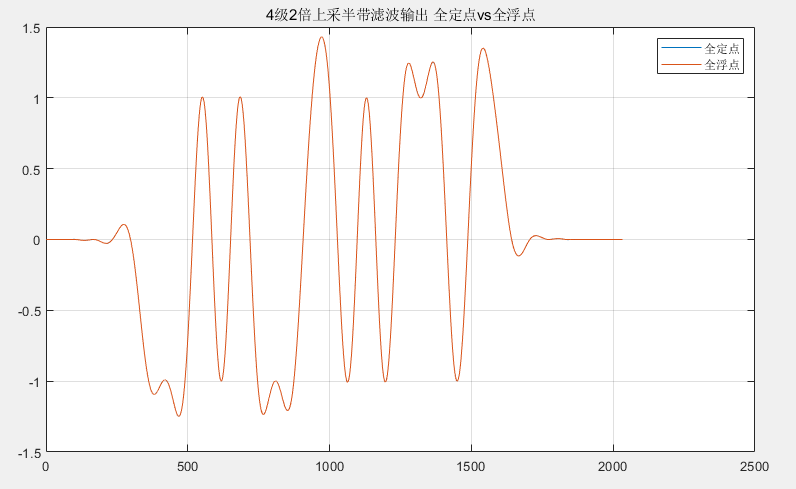
生成的20位BPSK符号为[-1,-1,-1,1,-1,1,-1,-1,-1,1,1,-1,1,-1,1,1,1,-1,1,1]。



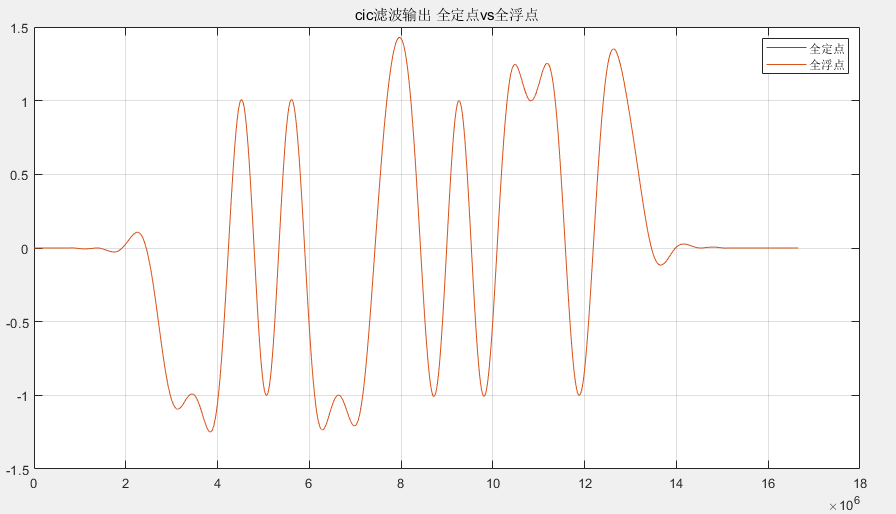
进行插值符号成型滤波后波形。



全定点vs全浮点4级2倍插值半带滤波，定浮点波形基本重合，定浮点均方误差为1.61\*10-4。



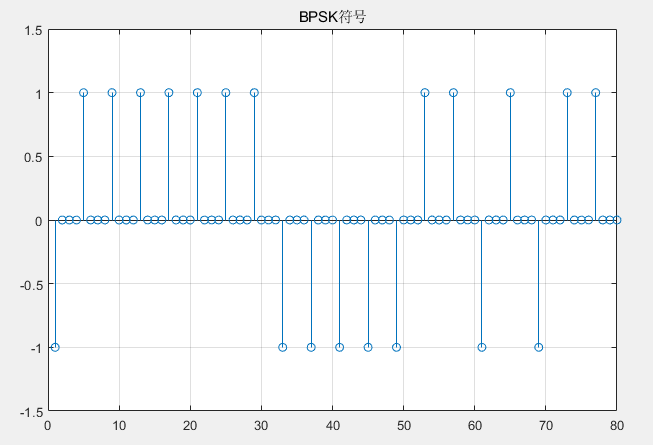
全定点vs全浮点8192倍插值CIC滤波，定浮点波形也基本重合，定浮点均方误1.76\*10-4。



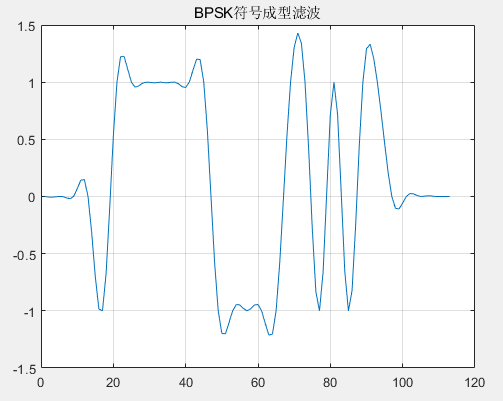
### 9.4.2 仿真2

CIC插值倍数变为7000，根据计算仍然为26，总上采倍数为448000，其余仿真条件与仿真1一致。

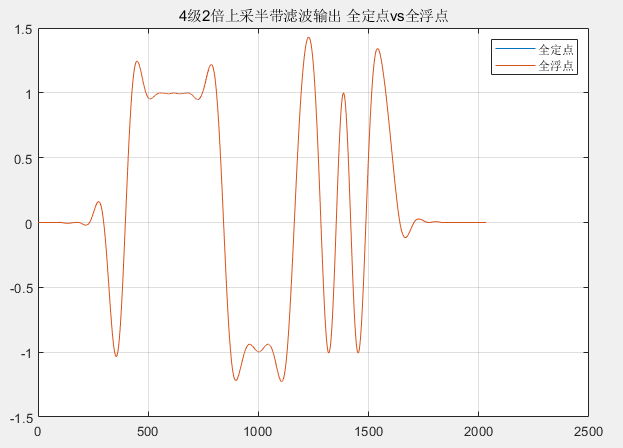
生成的20位BPSK符号为[-1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1,1,1,-1,1,-1,1,1]。



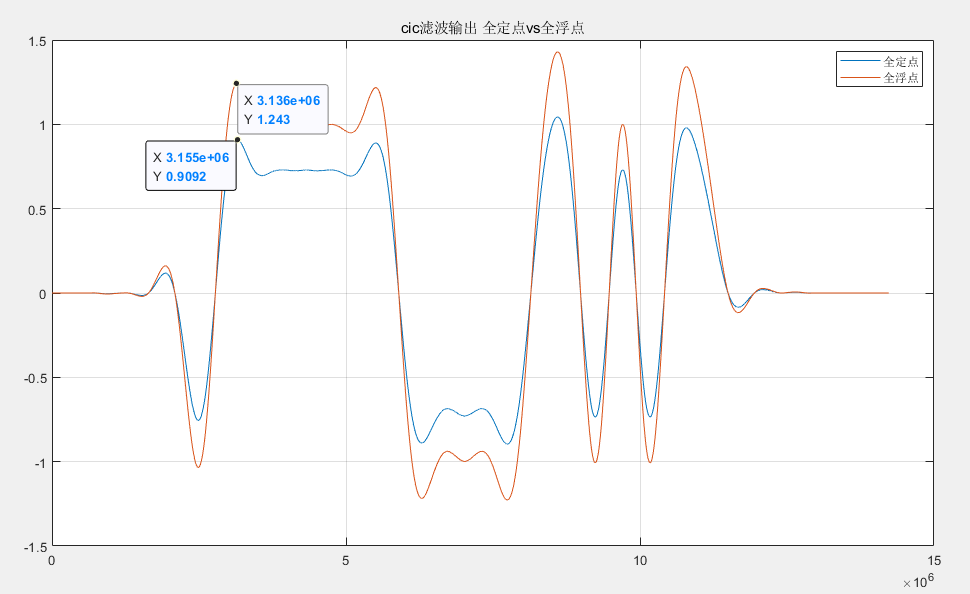
进行插值符号成型滤波后波形。



全定点vs全浮点4级2倍插值半带滤波，定浮点波形基本重合，定浮点均方误差为1.59\*10-4。



全定点vs全浮点7000倍插值CIC滤波，定浮点出现了差异，定点输出幅度约为浮点幅度的0.7315倍。原因在7000倍插值对应的准确值应为log2(7000^2)=25.5463 bit，7000^2即为整个CIC滤波的增益。实际上CIC滤波输出四舍五入右移bit数只能是取整数，为了防止饱和，对准确值向上取整为26bit，因此在定点化过程中相当于数据多右移了0.4537bit，对应幅度衰减为2^(-0.4537)=0.7302，与仿真结果看到的定浮点幅度差异吻合。取整误差最大不会超过1bit，因此定浮点幅度衰减不会低于0.5。对于的配置值，也可以根据成型滤波器类型，滚降系数等参数，以及调制方式和CIC滤波插值倍数这些因素综合考虑，以达到位宽的充分利用且不至于高位饱和的效果。

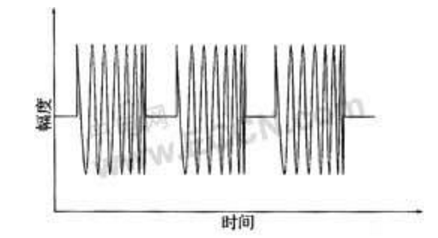


# 10. 脉冲内调制

## 10.1 算法原理

脉冲信号是指在脉冲宽度内部具有电平或者调制，在非脉冲宽度期间输出关断的信号。第个脉冲周期的时域表达式为

其中为载波，A为幅度，为第一个脉冲的起始时刻，表示第m个脉冲周期，表示第n个脉冲宽度。FPGA只需根据配置的正脉宽样点数和负脉宽样点数按照FPGA出口速率生成脉冲信号的包络信号即可，的载波调制由后级IQ调制器和混频完成。



## 10.2 参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 意义 | 取值范围 | 定点位宽 | 说明 |
|  | 按照FPGA输出采样率的正脉冲宽度样点数 |  | (32,0) |  |
|  | 按照FPGA输出采样率的负脉冲宽度样点数 |  | (32,0) |  |
|  | 正脉冲输出电平 |  | (16,12) |  |
|  | 负脉冲输出电平 |  | (16,12) |  |
|  | 脉冲重复频率 |  | 浮点 | 界面开放参数，用以软件计算正负脉宽的采样点数 |
|  | 正脉冲占空比 |  | 浮点 | 界面开放参数，用以软件计算正负脉宽的采样点数 |

## 10.3 定点算法描述

（1）定义一个32bit负脉宽计数器，和一个32bit正脉宽计数器，其初始值均为0，跳转到步骤（2）；

（2）判断负脉宽计数器值，若等于，负脉宽计数器清0，跳转到步骤（5），否则跳转到步骤（3）；

（3）FPGA输出I路按照接口速率输出一个，跳转到步骤（4）；

（4）负脉宽计数器加1，跳转到步骤（2）；

（5）判断正脉宽计数器值，若等于，正脉宽计数器清0，跳转到步骤（2），否则跳转到步骤（6）

（6）FPGA输出I路按照接口速率输出一个，跳转到步骤（7）；

（7）正脉宽计数器加1，跳转到步骤（5）；

（8）Q路及I路没有指定输出信号时，按照接口速率输出0；

## 10.4 仿真验证

略。