FPGA 综合实验报告 3

周子涵 2018011218014

一、题目需求分析

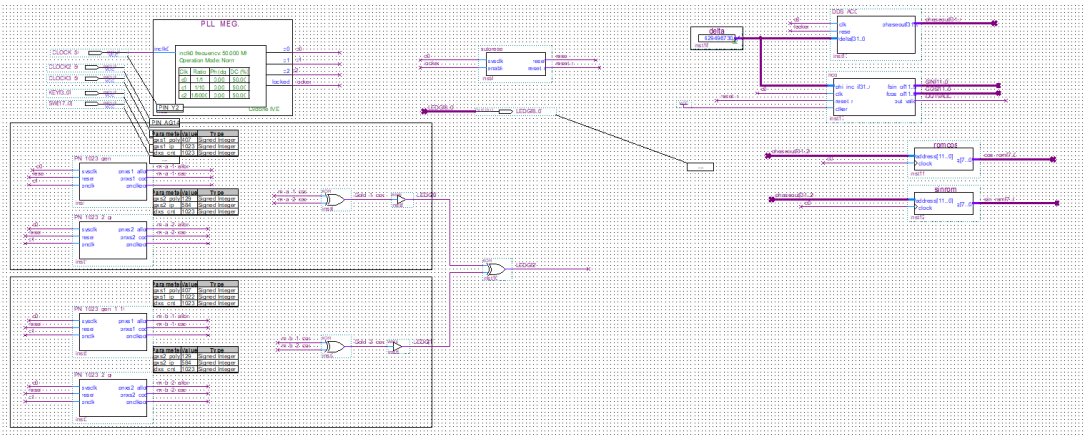
利用 Verilog 语言编程，在 FPGA 中产生两路伪随机码，通过改变其中一路伪 随机码的初相，验证 PN 码的特性，体会直接序列扩频通信系统的基本特点。

利用 Verilog 语言、在 Quartus 软件中，使用 LPM 模块、IPcore 等模块，在 FPGA 中设计一种数字频率直接合成器。可以通过改变 DDS 的初值，改变系统时钟得到 工程需要的不同频率信号源。利用 SignalTap 验证 DDS 生成的信号频率，并将之与 sin 表的查找表文件比较。

调整 PLL 分倍频系数，生成合适的时钟频率，分别用 IPcore 和自编模块设计 DDS，产生 25MHz 的单频信号。设计 PN 码模块，产生 Gold 码序列（Gold 码序列 的生成多项式、初相参考实验五内容）。设计模拟串行信号产生模块，将此串行 信号用 PN 码扩频。应用 BPSK 调制原理，将扩频后的串行信号调制在 25MHz 的载 波上。

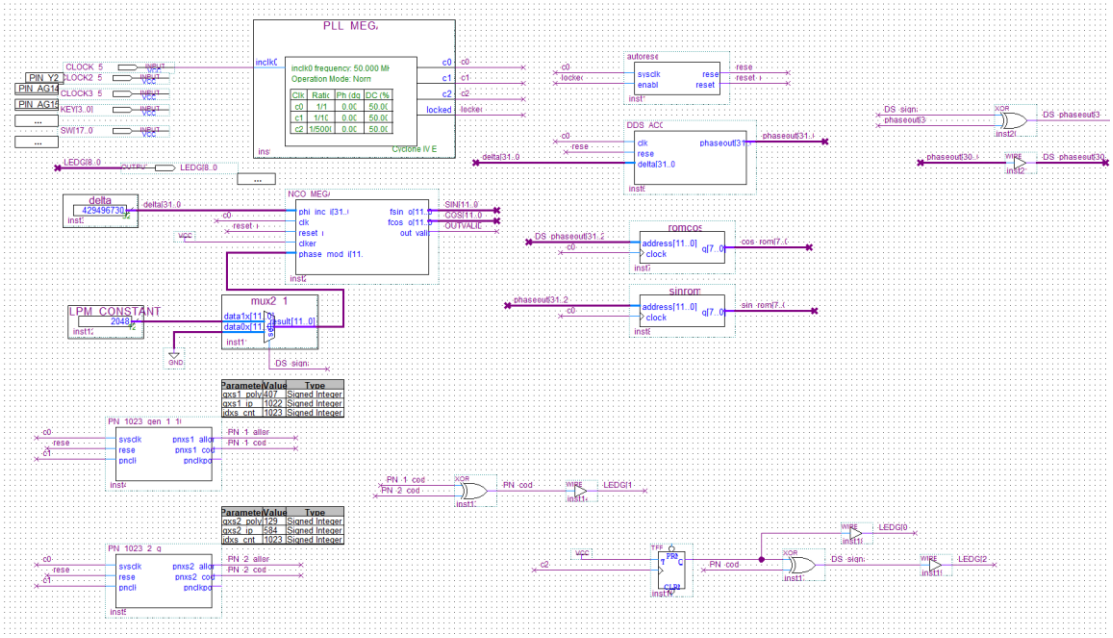
二、顶层设计

PN 码及 DDS 的设计如下图：



首先生成时钟和上电复位模块，生成一个 PN 码序列，然后构建一个新的 PN 码序列，将两者进行模 2 加，得到 gold 码序列，复制一个 gold 码序列，修改初相，将两者进行异或，生成相位累加器和波型存储器，截取相位累加器的高 12 位作为波形存储器的查表地址位。

BPSK 的设计如下图：



首先生成时钟和上电复位模块，在 PN 码和 DDS 的基础上，用 T 触发器产生 0,1 序列作为调制信息，产生任意串行时钟信号，将 gold 码与调制信息模二加，在相位累加器、sin 表、cos 表模块中，利用调制信号改变查找 表的地址信号最高位，实现相位调整，也可以实现 BPSK 信号调制。

三、实验原理

PN 码：

2.1 m序列

m序列又称为最长线性反馈移位寄存器序列。由于m序列的产生结构比其它的伪随机序列简单，再加上学者们对它的研究也比较早，理论知识比较成熟，因此它是使用最多的一种伪随机序列。如图2是一个级线性反馈移位寄存器。其中a_0，a_1，……，a_{n-1}是移位寄存器的状态，也叫做码产生器的**初相**，每个寄存器的取值为0或者1。在系统时钟的驱动下，每级移位寄存器的状态从左向右移动，成为下一级状态。

c_0，c_1，……，c_{n-1}为移位寄存器的反馈系数，也叫做码产生器的**生成多项式**，用来控制各级寄存器的接通与断开，当c为1时表示连通，为0时表示断开，加法器采用模2加法。反馈移位寄存器产生的序列取决于反馈系数，产生的二值序列的值为：

$$\{a_n\}=c_1a_{n-1}+c_2a_{n-2}+\cdots\cdots+c_na_0$$

反馈系数c_i一旦确定，产生的序列就确定了，其最大周期为2^n-1。

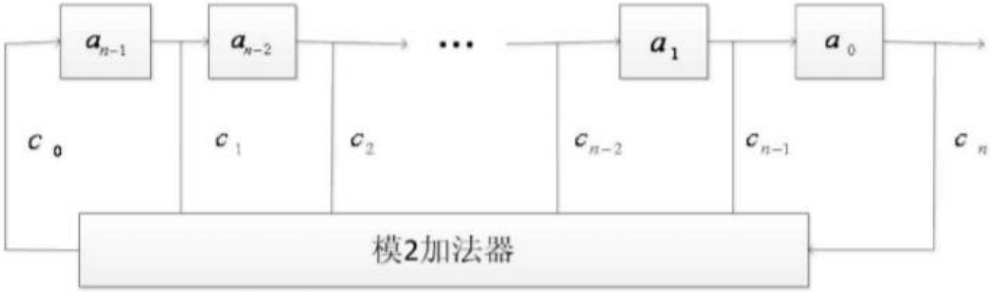


图2. n位线性反馈移位寄存器

2.2 Gold序列

m序列自相关特性良好，但是其互相关特性不好，且可用于产生m序列的本原多项式较少，在一些要求地址码多的系统中m序列就很难满足要求。Gold序列是由两个可以组成优选对的m序列构成。Gold序列的自相关特性优良，互相关特性也很好，同时它所能产生的地址码的数目比m序列要多很多。因此，Gold序列成为了一种常用的伪随机序列。

产生 Gold 码序列可以将两个m序列的移位寄存器并联再进行模 2相加。图为并联 Gold 码产生器的逻辑图。我们在本次实验中需要设计验证的就是Gold序列。

2.3 M序列

M序列又称为最长非线性移位寄存器序列，与m序列相比，M序列的发生器也是由多级移位寄存器构成的，但是在m序列中这些寄存器是线性的，而在M序列中这些寄存器是非线性的。M序列的产生可以根据m序列的结构来实现。因为m序列已经包含了全部的非0状态，只是少了一个全0的状态，所以，通过m序列来产生M序列时，只需要在合适的位置上插入一个0，让它变成全0状态就可以了。这样就可以将长度为2n-1的m序列变成长度为2n的M序列。与同阶数的序列相比，M序列的数目要多得多，甚至和Gold序列相比，当阶数较高时，M序列的数目也占有优势，M序列是作为多址接入码的良好选择。M序列的自相关函数不具有双值特性，和m序列相比，其相关函数旁瓣振荡较大。

DDS:

DDS 的原理

直接数字频率合成技术是直接从相位角度出发的一种新颖频率合成方法，其理论依据是时域抽样定理。下面以正弦信号的产生为例，说明 DDS 的基本原理。

任意频率为的正弦信号的时域表示式

$$\varphi(t)=\sin(2\pi f_0 t)=\sin\omega_0 t=\sin\theta(t)$$

如果对该正弦信号以周期$T_c=1/f_c$（f_c为采样频率）进行采样，得到离散的波形序列

$$\varphi(n)=\sin(2\pi f_0 nT_c)=\sin\theta(n)=\sin n\Delta\theta \quad (n=0,1,2...)$$

式中：

$$\Delta\theta=2\pi f_0 T_c=2\pi f_0/f_c$$

为两次采样之间的相位变化量。显然，离散波形序列的相位序列具有显著的线性，即相邻之间相位的增量是一个固定值，且这个固定值只与正弦信号的频率和采样频率的比值有关。

DDS 的基本结构

通常所说 DDS 的一般指的查找表方法的 DDS。原理框图如图 1 所示，它主要是由系统时钟、相位累加器、波形存储器、数模转换器及低通滤波器组成。

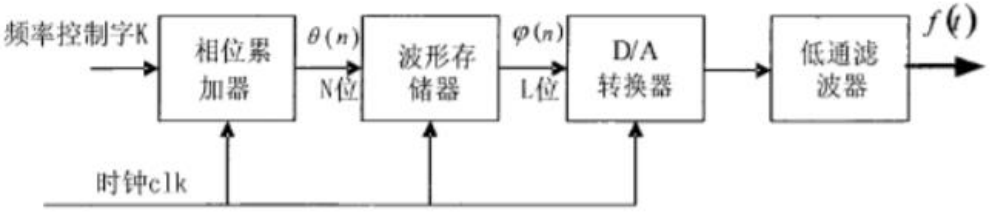


图 1. DDS 原理框图

其中 K 为频率控制字，clk 为系统时钟，N 为相位累加器的字长，L 为 ROM 幅度量化位数及 D/A 转换器的量化位数。相位累加器在时钟 clk 的控制下以步进 K 作累加，输出的 N 位 $\theta(n)$ 为二进制码，作为波形存储器的地址，对波形存储器进行寻址，波形存储器输出 L 位的幅度码 $\varphi(n)$ 经 D/A 转换器转化为阶梯波，再经过低通滤波器滤波平滑后就得到合成的信号波形 $f(t)$。

1） 相位累加器

相位累加器由一个 N 位的加法器和一个 N 位的寄存器构成，是典型的反馈电路，其结构如图 3 所示。在系统时钟 clk 的控制下，加法器先将频率控制字 K 与寄存器输入的相位数值相加，然后把相加得到的相位数值再次送到寄存器中，寄存器再将新的相位数值反馈至加法器的输入端，以便在下一个 clk 时钟时继续与频率控制字 K 相加。当寄存器内的数值达到满量时，便会产生一次溢出，完成一个周期。通常情况下，累加器采用二进制，位数为 N 的累加器，其加满溢出值为 2^N。

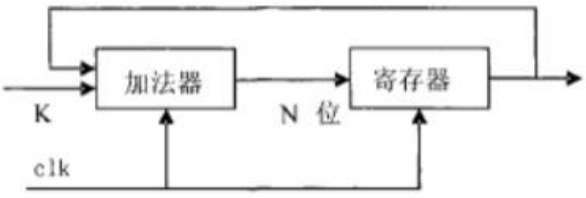


图 3. 相位累加器原理框图

2） 波形存储器

波形存储器用于存储所产生波形一个周期的数据，可设计成 ROM，也可根据产生的波形利用不同的存储器。当设计为 ROM 时，以正弦波形存储器为例，其原理结构如图 4 所示，ROM 的输入为相位累加器产生的相位量化序列，作为 ROM 的地址进行寻址。若 ROM 有 n 位地址线，则有 2^N 个存储单元，共存储正弦波形的一个周期数据。如果重复地从 0～2^N 个单元读出波形 ROM 的数据，在波形数据存储器的输出端就会得到周期的正弦波。

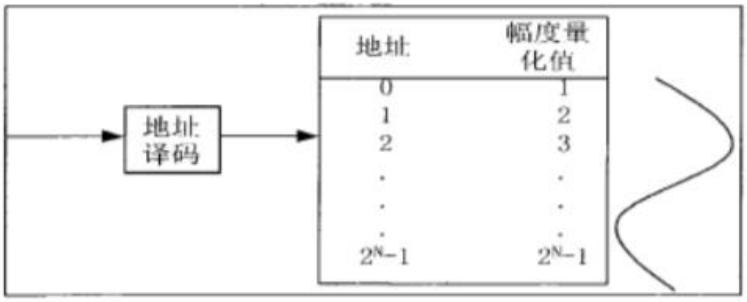


图 4. 波形存储器的结构

BPSK:

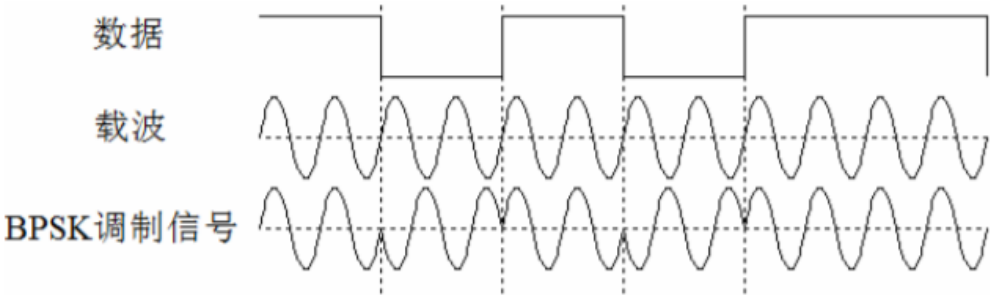


图1. BPSK的基本原理

本次实验在实现硬件调制电路时，通过设计两个相位寄存器来寄存调制信号、值所对应的相位分量，并通数据选择器和键控频率模块的配合来实现信号的调制输出，其电路的基本结构如下图所示。

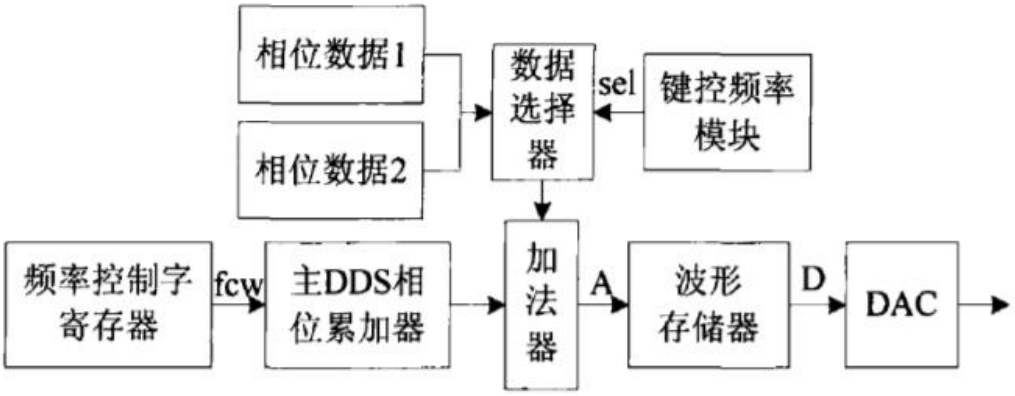


图 4. BPSK 的 FPGA 实现方式

在 DDS 系统中实现 BPSK 时，需要在相位累加器的输出端添加一个加法器，加法器通过将数字调制信号所对应的相位值加入 DDS 电路的输出相位中，然后用于波形存储器的寻址，从而改变了输出信号的相位值。从 BPSK 硬件电路输出信号的时序仿真图 5 中可以看出，在调制信号分别为 0、1 时，其调制输出的信号相位刚好相差了 180° 。

三、单元设计

PN 码的实验步骤：

1. 启动系统生成器，生成 project，创建 PLL 使用 50MHz 作为系统钟，5MHz 作为码钟；
2. 构建 PN_1023_gen_1 模块，验证生成的序列码型，此码序列参数如下；
   - 码生成多项式码：407；
   - 初相：1023
   - 截短长度：1023

   Verilog代码文件为：PN_1023_gen_1.v.

   根据图2所示的移位寄存器原理图，我们可以用同样的方式描述PN_1023_gen_1.v的PN码产生器原理框图，如图3所示：

生成多项式：$(407)_{10} = (0110010111)_2$

初相： 1111111111

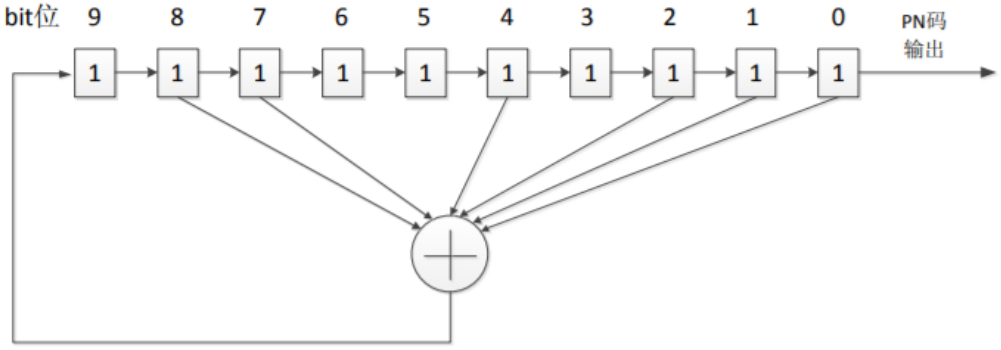


图4. 1023位m序列产生器原理框图

如果利用原理图的方式，可以选择移位寄存器设计码产生器如下图所示：

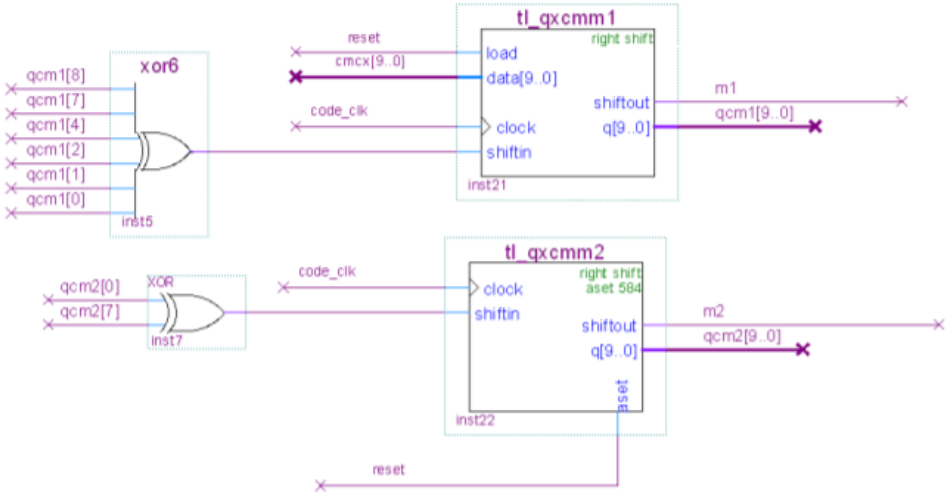


图5. 1023位m序列产生器原理图方式输入顶层文件

3. 在理解 PN_1023_gen_1 模块的基础上构建新的 PN 码序列产生模块，要求如下，并验证生成的序列码型；
   - 码生成多项式码：129；
   - 初相：584
   - 截短长度：1023
4. 按照图 4 的方法，将两个模块输出的 m 码序列进行模二加，得到 Gold 码序列，并验证码型；
5. 复制一个相同的Gold码产生器，仅将第一个m序列产生器的初相改为1022，试与前一个 Gold 码产生器生成的码型做比较，可将二者进行异或。体会 PN 码的多址特性和保密传输特性。结构框图如下图所示。

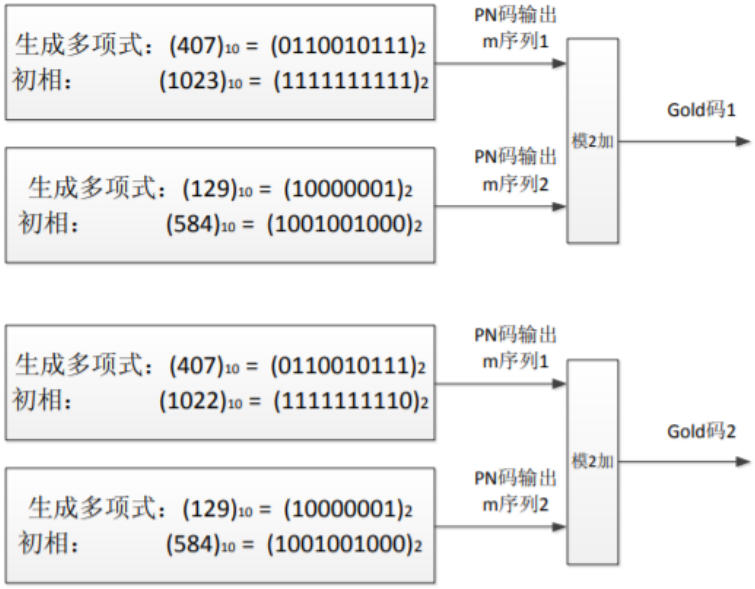


图 6. 更改初相观察 PN 码生成的变化

6. 利用 signalTap 调试、验证关键功能。

DDS 的实验步骤:

1. 启动系统生成器，生成 project，创建 PLL 使用 50MHz 作为系统钟；
2. 利用 Verilog 语言编写上电复位模块，与 PLL 的 locked 输出信号配合，对整个 project 进行上电复位，不建议使用 SW 拨段开关作为复位信号，可以使用 KEY 按键开关作为手动复位按钮。(上电复位模块参考文件 autoreset.v)
3. 利用 Verilog 语言编写图 3 所示的相位累加器，相位累加器累加位数 N=32。根据前述公式 $K=\frac{f_0}{f_c}2^N$，按照 $f_0=5MHz, f_c=50MHz$ 计算 K 值，将 K 值作为相位累加器“delta[31..0]”的输入。(相位累加器模块参考文件 DDS_ACC.v)
4. 直接利用 LPM_ROM 构建图 1 所示的波形存储器。波形存储器地址位：12 位；幅度量化位数：8 位。

5. 截取相位累加器的高 12 位作为波形存储器的查表地址位，查找表中对应波形幅度值。
6. 利用 SignalTap 观测相位累加器、波形存储器的输出，并与 mif 文件比对。观察 SignalTap 时，分别设置总线格式为“Signed Decimal in Two’s Complement”、“Signed Line Chart”。设置方式及观测结果如下图所示

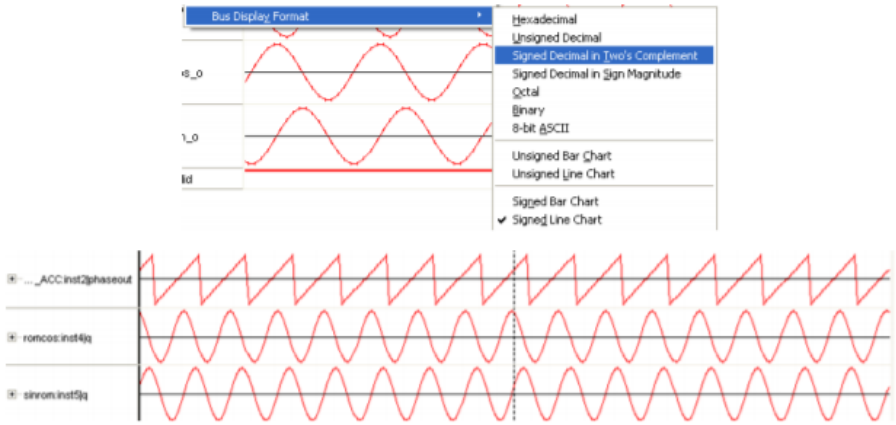
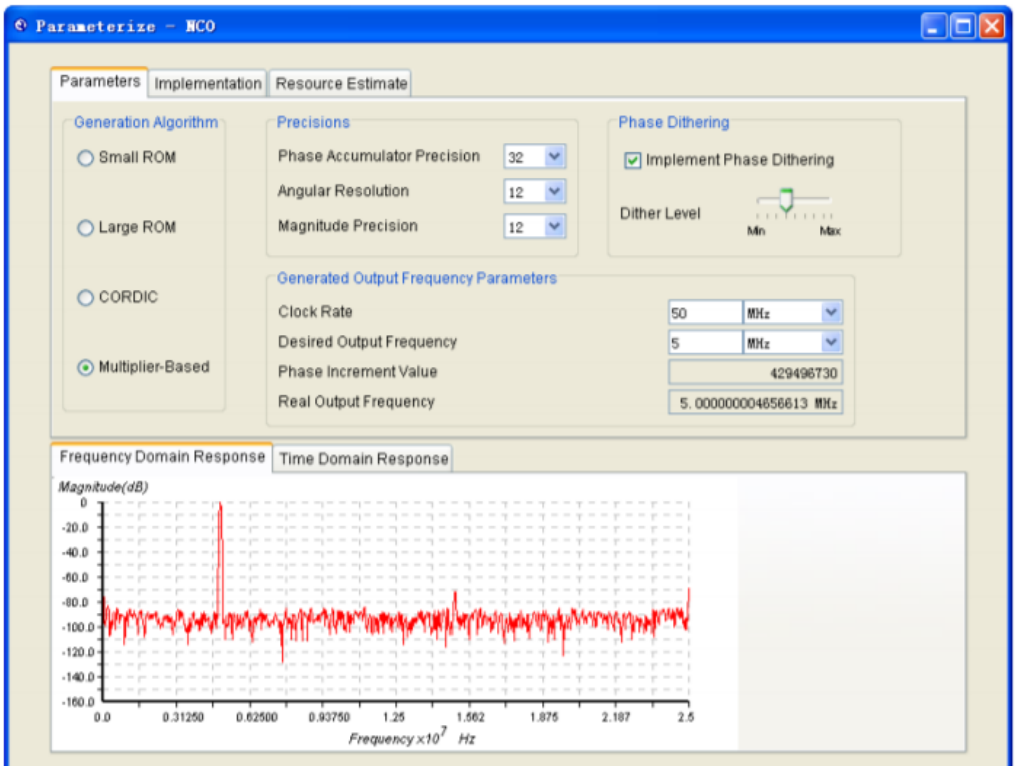
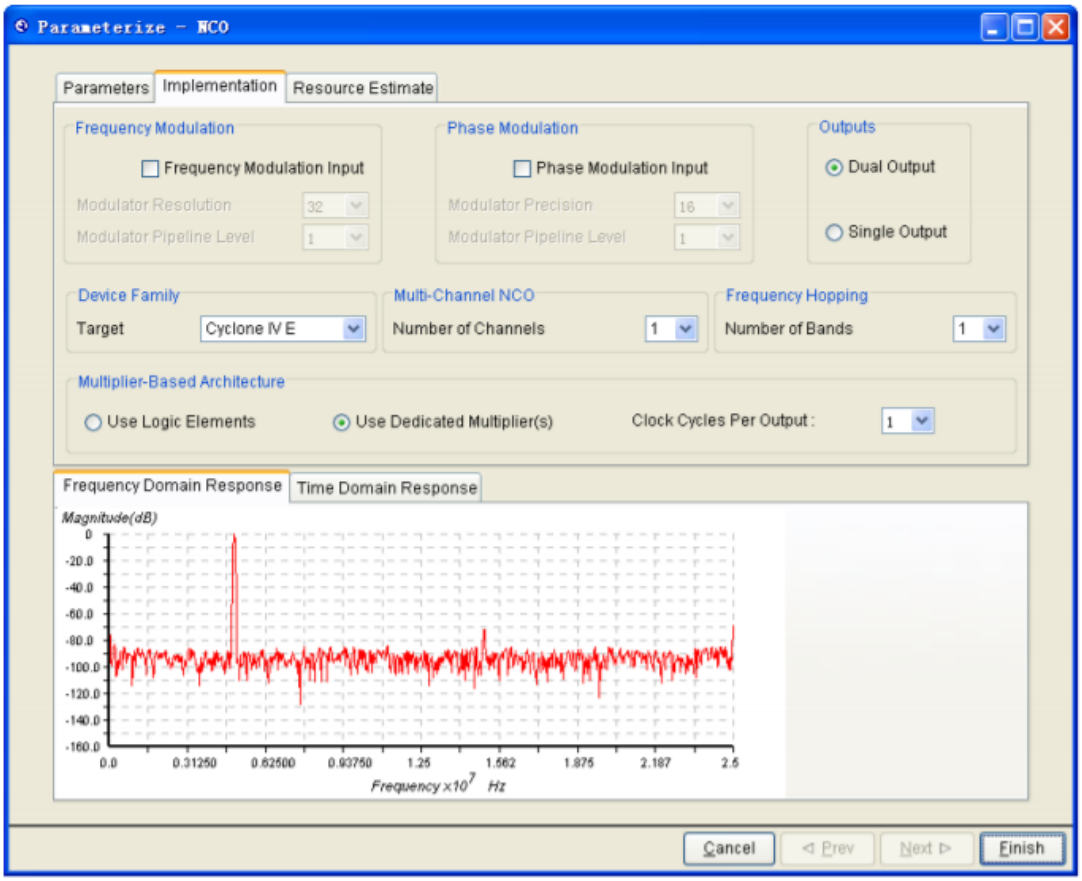
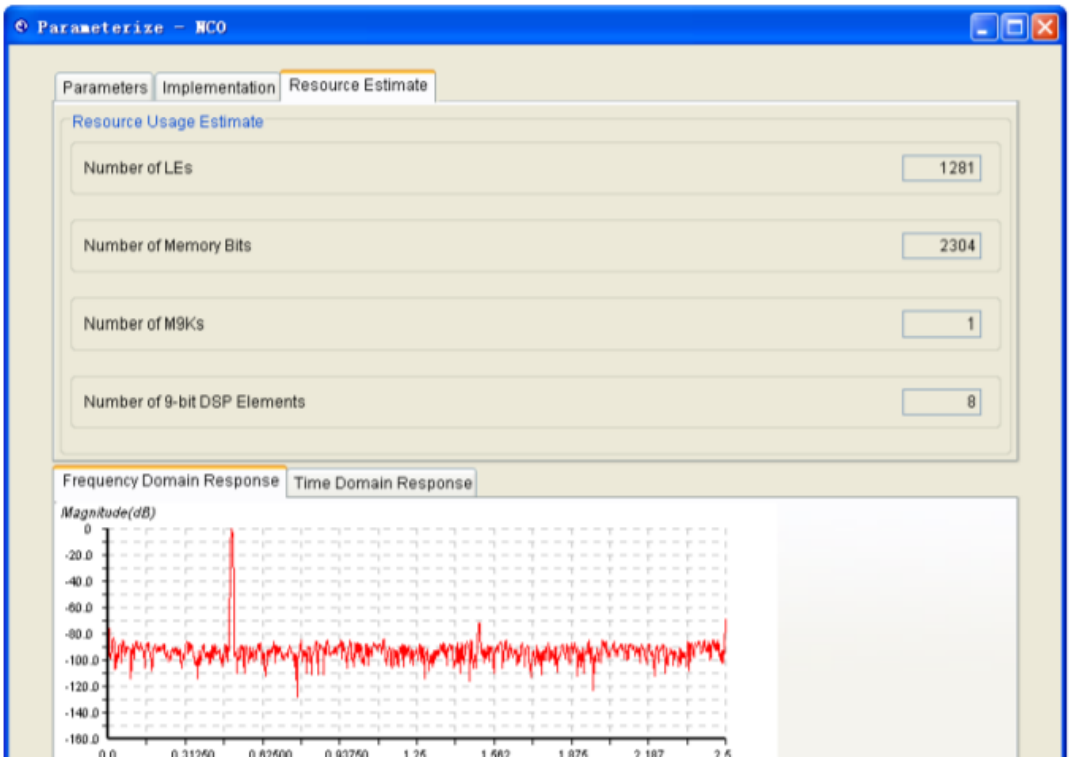


图 7. 波形观测结果

7. 直接调用 IPcore 中的 NCO 模块，直接生成 DDS，NCO 参数按照下图设置。




Parameterize - NCO
Parameters
Implementation
Resource Estimate
Frequency Modulation
Frequency Modulation Input
Modulator Resolution
32
Modulator Pipeline Level
1
Phase Modulation
Phase Modulation Input
Modulator Precision
16
Modulator Pipeline Level
1
Outputs
Dual Output
Single Output
Device Family
Target
Cyclone IV E
Multi-Channel NCO
Number of Channels
1
Frequency Hopping
Number of Bands
1
Multiplier-Based Architecture
Use Logic Elements
Use Dedicated Multiplier(s)
Clock Cycles Per Output :
1
Frequency Domain Response
Time Domain Response
Magnitude(dB)
0
-20.0
-40.0
-60.0
-80.0
-100.0
-120.0
-140.0
-160.0
0.0
0.31250
0.62500
0.93750
1.25
1.562
1.875
2.187
2.5
Frequency x10^7 Hz
Cancel
Prev
Next
Finish


Parameterize - NCO
Parameters
Implementation
Resource Estimate
Resource Usage Estimate
Number of LEs
1281
Number of Memory Bits
2304
Number of M9Ks
1
Number of 9-bit DSP Elements
8
Frequency Domain Response
Time Domain Response
Magnitude(dB)
0
-20.0
-40.0
-60.0
-80.0
-100.0
-120.0
-140.0
-160.0

8. 利用 signalTap 观察 NCO 的输出。
9. 改变 K 值，产生 25MHz 的信号，观察输出信号波形。
10. 改变 PLL 的分频比，调整系统钟至 250MHz，计算 K 值，重新产生 25MHz 信号。观察信号波形。

BPSK 的实验步骤:

1. 启动系统生成器，生成 project，创建 PLL 与上电复位模块；
2. 合理设置 PLL 的分频比，分别产生 250MHz、5MHz、10KHz 的时钟；
3. 250MHz 作为 DDS 系统钟，根据 DDS 公式，计算初值，分别用 IPcore 和自编相位累加器、Sin 表的方式产生 25MHz 载波信号；
4. 5MHz 作为 PN 码产生器的码钟，产生 5MHzGold 码，Gold 码的初相和生成多项如下：

   m1：生成多项式407；初相1023

   m2： 生成多项式129；初相584
5. 10KHz 作为产生调制信息的时钟，产生任意串行时钟信号。可以使用 T 触发器直接产生 0、1 序列作为调制信息，T 触发器的使用方法如图 6 所示。

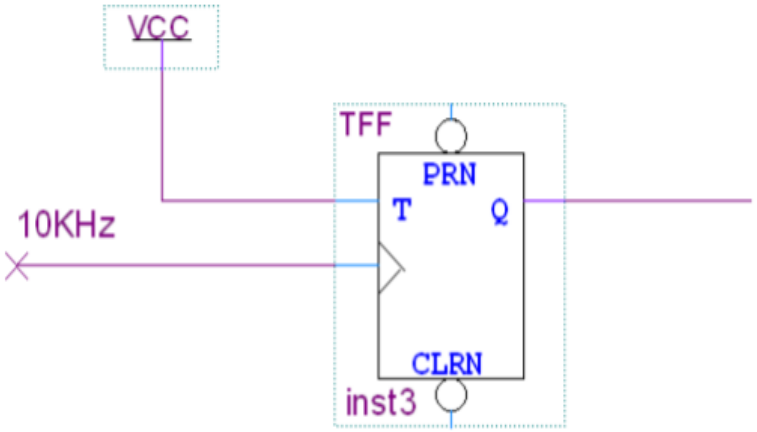
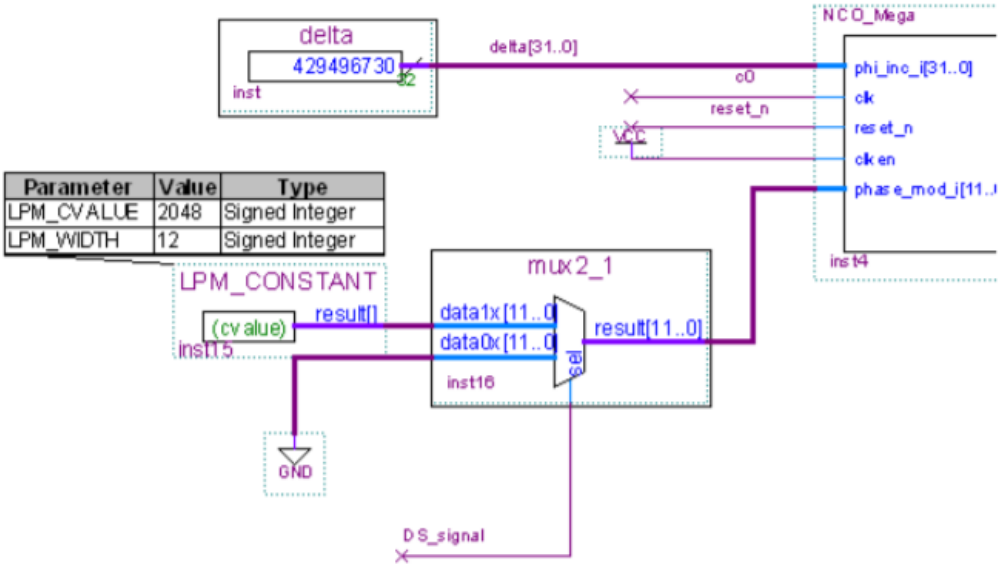


图 6. T 触发器模拟产生调制信息

6. 将 Gold 码与调制信息模二加（扩频）后作为 BPSK 的调制信号；
7. 在利用 IPcore 生成的 NCO 模块中，添加 phase_mod 输入端，采用图 4 所示的方式，利用扩频后的调制信号选择全 0 或全 1 数据作为相位调整参数。实

现框图如图 7 所示。相位调制端口参数设置如图 8 所示。



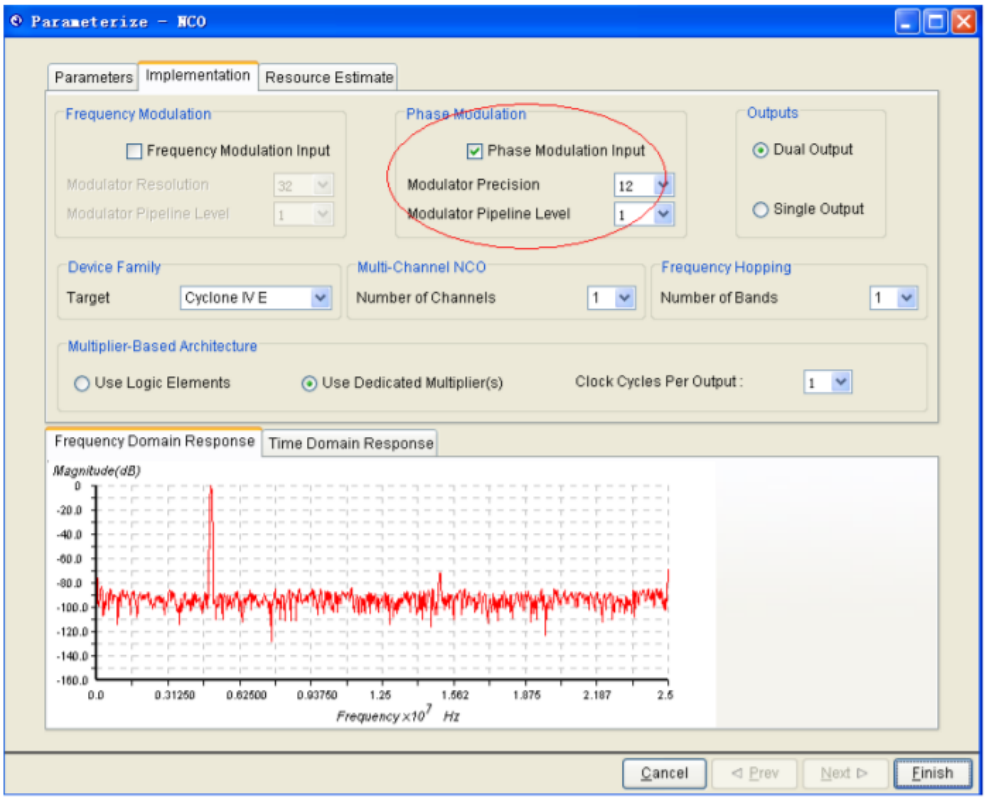


图 8. 相位调制端口参数设置

8. 在自己编写的相位累加器、sin 表、cos 表模块中，利用调制信号改变查找表的地址信号最高位，实现相位调整，可以同样实现 BPSK 信号调制，实现框图如图 8 所示。

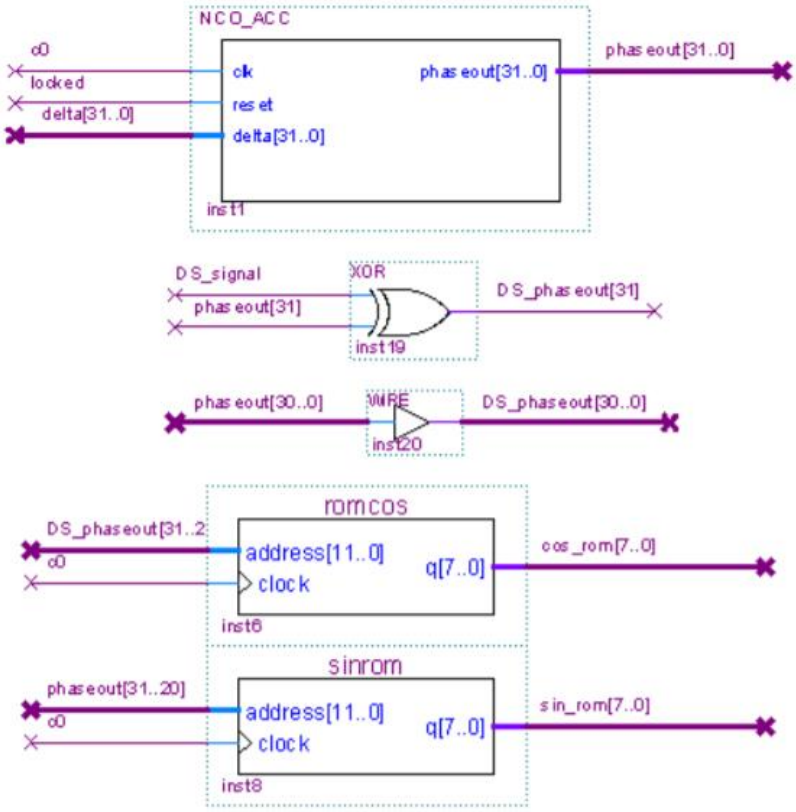


图 8. 相位选择法产生 BPSK 信号的第二种方法

9. 利用 signalTap 观测所有关键信号。分析 BPSK 的调制过程。

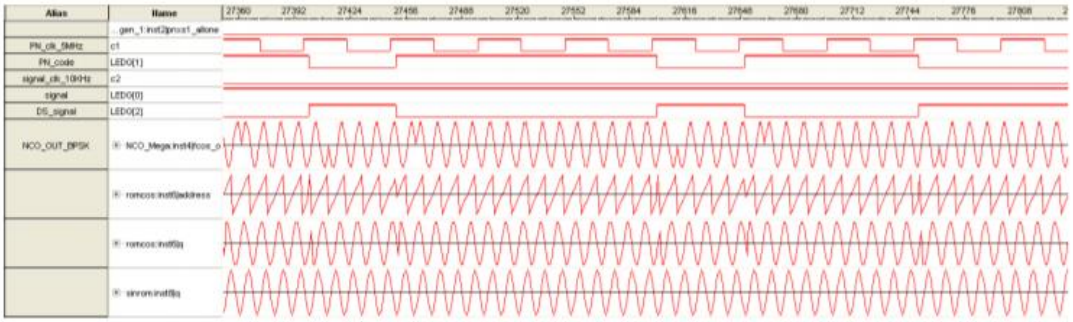
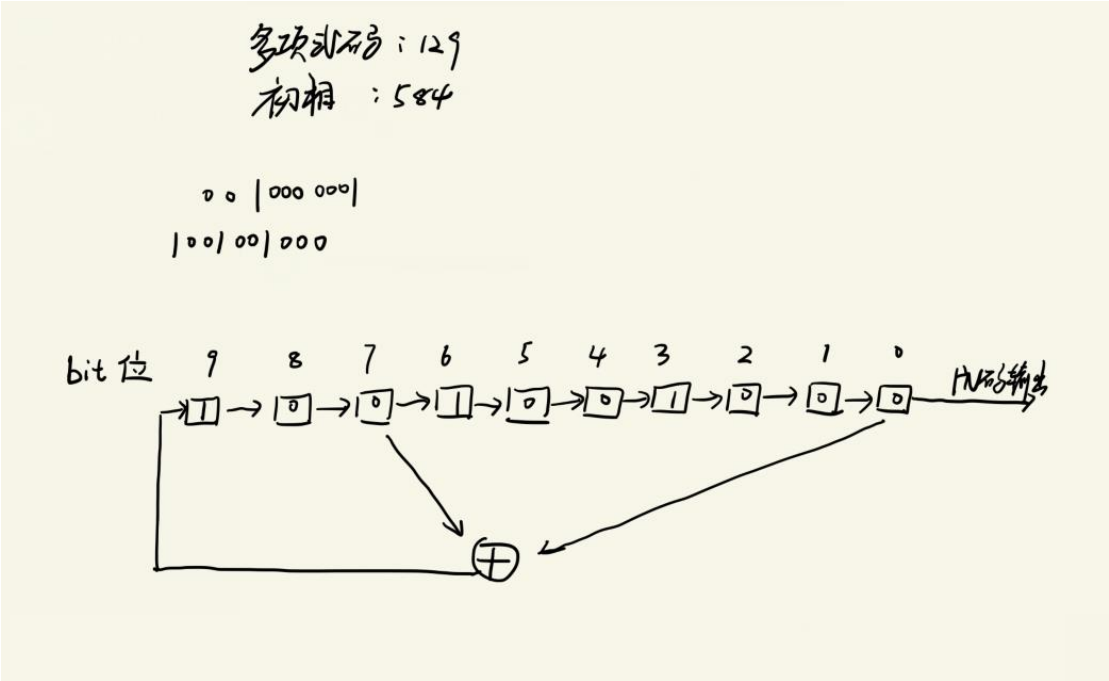


图 9. 信号观测结果

M 序列产生代码如下：

```
module PN_1023_2_gen  (sysclk,reset,pnclk,
                       pnxs2_allone,pnxs2_code,
                       pnclkpos);


   input sysclk,reset,pnclk;
   output pnxs2_allone,pnxs2_code;
   reg    pnxs2_allone,pnxs2_code;

   output pnclkpos;
   wire pnclkpos;

   reg [9:0]  gxs2regshift;

   wire polyvalgxs2;

   reg [9:0]  gxs2_counter;
   reg [1:0]  tempa,tempb;

   parameter  gxs2_poly = 129;
   parameter  gxs2_ip   = 584;
   parameter  jdxs_cnt  = 1023;


/*-------------------------------------code clk gen--------------------------*/
always @(posedge sysclk)
   begin
      if(reset)
         begin
            tempb<=2'd0;
         end
      else
         begin
            tempb<={ tempb[0],pnclk };
         end
   end
assign   pnclkpos=( tempb[0] & ~tempb[1] );


/*-----------------------------------------short(1023) code gen-----------------
assign polyvalgxs2=^(gxs2_poly & gxs2regshift);

always @(posedge sysclk)
   begin
      if(reset)
         begin
            gxs2regshift<=gxs2_ip;
            gxs2_counter<=0;
         end
      else
         begin
            if(pnclkpos)
               begin
                  if(gxs2_counter < jdxs_cnt-1)
                     begin
                        gxs2regshift<={polyvalgxs2,gxs2regshift[9:1]};
                        gxs2_counter<=gxs2_counter+18'd1;
                     end
                  else
                     begin
                        gxs2regshift<=gxs2_ip;
                        gxs2_counter<=0;
                     end
               end
            else
               begin
                  gxs2regshift<=gxs2regshift;
                  gxs2_counter<=gxs2_counter;
               end
         end
   end

always @(posedge sysclk)
   begin
      if(reset)
         begin
            pnxs2_code<=0;
            pnxs2_allone<=1;
         end
      else
         begin
            pnxs2_code<=gxs2regshift[0];
            pnxs2_allone<=(gxs2regshift==gxs2_ip);
         end
   end


endmodule
```

流程图如下：



四、回答问题

5. 试说明采用 IPcoe 和自编模块生成 BPSK 信号的这两种方式在信号时序上的区别。

在用自编模块生成 bpsk 信号时，信号通过自编模块后，需要再通过一步最高位反转才可以实现相位的改变，而在使用 IP 核生成 bpsk 信号时，通过 IP 核后，相位就可以直接实现反转而不需要经过最高位反转。

五、心得体会

理解了一个工程的建立的完整过程，首先需进行需求分析，然后按自顶向下的方法一步一步建立，一个工程看似巨大，但其实将它分解为一个一个的模块，并对每个模块进行理解后再搭建出每一个模块，最后就可以将模块连接为整体，从而得到完整的工程，增强了动手能力，理解了一些 verilog 语言的知识，为将来的工作打下了一些基础。学习伪随机码的产生原理，数字频率直接合成（DDS）的基本原理，综合应用了在本课程前期学习并设计的 PN 码产生

器模块、数字频率直接合成（DDS）模块，按照 BPSK 调制原理，设计了一个 BPSK 调制器的数字部分。