

## 2020 年新工科联盟-Xilinx 暑期学校团队项目设计文档

作品名称	基于 FPGA 的数字调制器设计
板卡型号	xc7s15ftgb196-1
所在班级	东南大学电子学院 A 班
成员姓名、学号、学校	王欣茹 06017407 东南大学 电子科学与工程 季雨佳 06017405 东南大学 电子科学与工程
Github 链接	<a href="https://github.com/SEU-wxr/xilinx20200801.git">https://github.com/SEU-wxr/xilinx20200801.git</a>

第一部分  
设计概述 /Design Introduction

在当今高度信息化的社会，信息和通信已经成为现代社会的“命脉”。信息作为一种资源，只有通过广泛地传播与交流，才能产生利用价值，促进社会成员之间的合作，推动社会生产力的发展，创造出巨大的经济效益。而通信作为传输信息的手段或方式，与传感技术、计算机技术相互融合，已经成为 21 世纪国际社会和世界经济发展的强大推动力。可以预见，未来的通信对人们的生活方式和社会发展将会产生更加重大和意义深远的影响。

通常，按照信道中传输的是模拟信号还是数字信号，相应地把通信系统分为模拟通信系统和数字通信系统。与模拟通信相比，数字通信具有很多优良的特性：

- ① 数字信号便于存储、处理、抗干扰能力强；
- ② 数字信号便于交换和传输；
- ③ 可靠性高，传输过程中的差错可以设法控制；
- ④ 数字信号易于加密且保密性强；
- ⑤ 通用性和灵活性好。

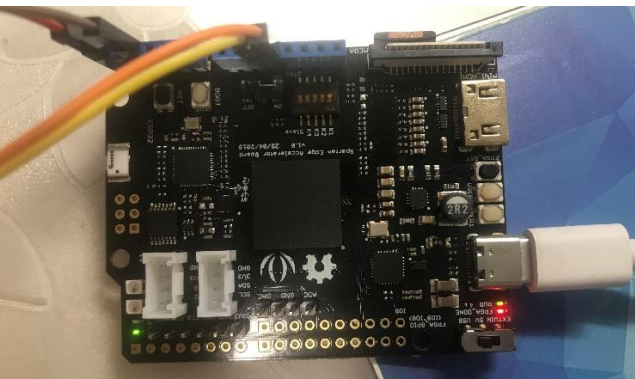
它的主要缺点是设备复杂并且需要较大的传输带宽。近年来，随着大规模集成电路的出现，数字系统的设备复杂程度和技术难度大大降低，同时高效的数据压缩技术以及光纤等大容量传输介质的使用正逐步使带宽问题得到了解决。因此，数字传输方式日益受到欢迎。

数字信号的传输方式分为基带传输和带通传输。然而，实际中的大多数信道因具有带通特性而不能直接传送基带信号，这是因为数字基带信号往往具有丰富的低频分量。为了使数字信号在带通信道中传输，必须用数字基带信号对载波进行调制，以使信号与信道的特性匹配。

二进制数字调制所用调制信号由“0”和“1”代表的数字信号脉冲序列组成。因此，数字调制信号也成为键控信号。二进制振幅调制、频率调制、相位调制和差分相位调制分别称为振幅键控（2ASK）、频移键控（2FSK）、相移键控（2PSK）、差分相移键控（2DPSK）。数字调制产生模拟信号，其载波参量的离散状态是与数字数据相对应的，这种信号适宜于在带通型的模拟信道上传输。

成员分工：

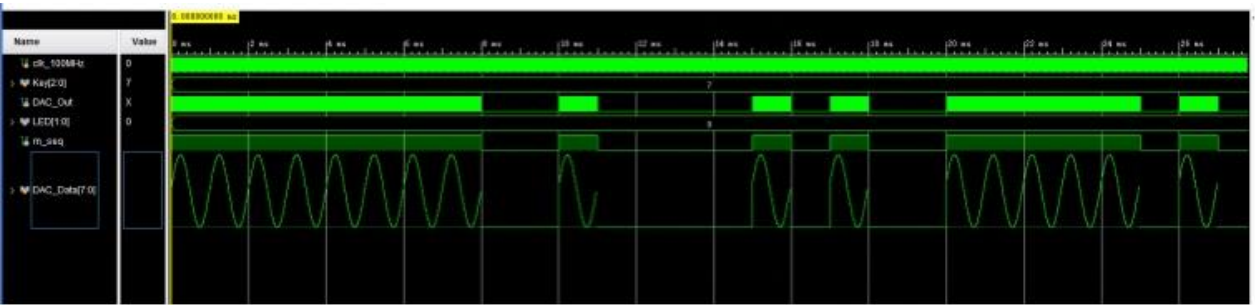
组员	贡献	完成部分
王欣茹	70%	载波发生模块、调制模块、按键模块、DAC 输出模块、仿真
季雨佳	30%	随机序列发生模块、测试示波器



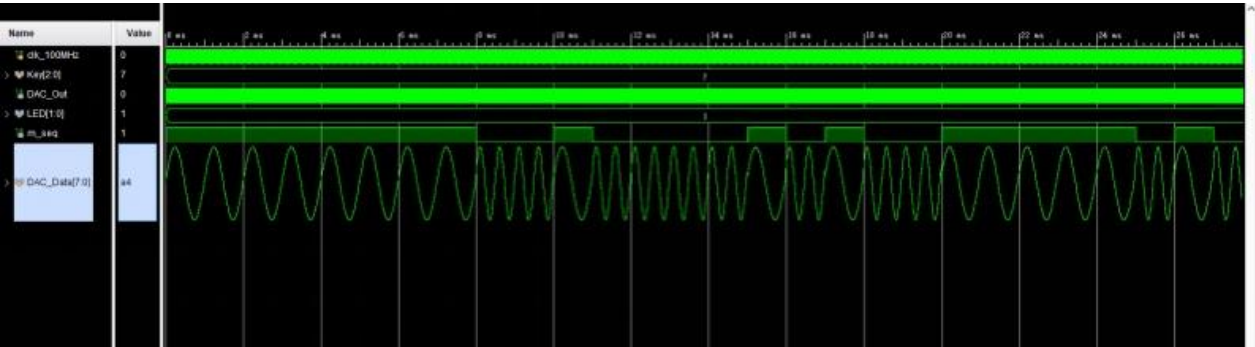
（硬件连接图）

连接示波器测试时，无法输出稳定波形，功能仿真如下：

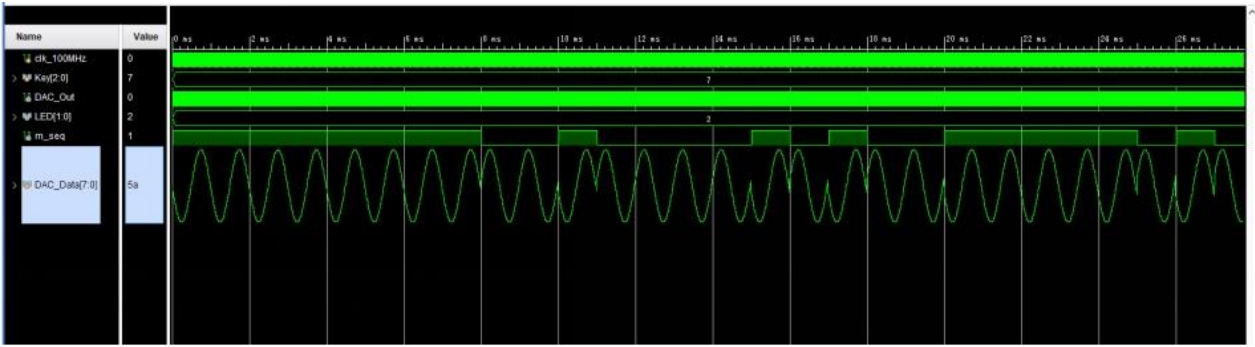
1、2ASK



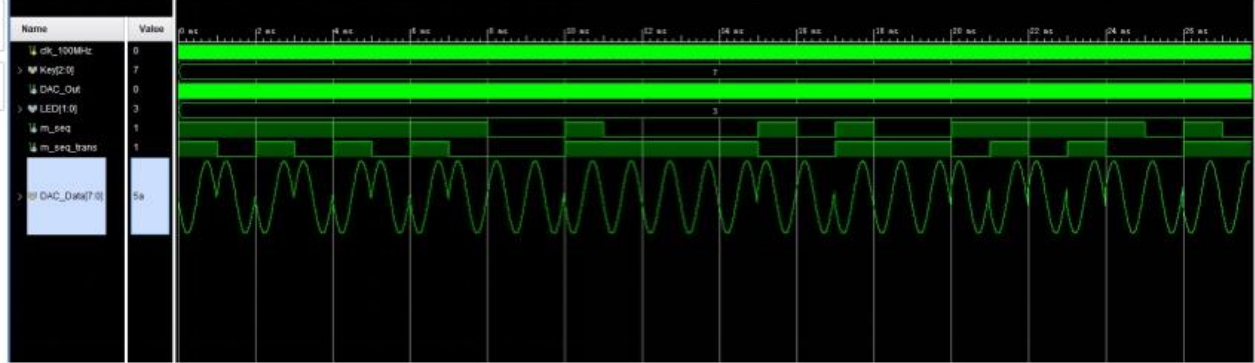
2、2FSK



3、2PSK



4、2DPSK

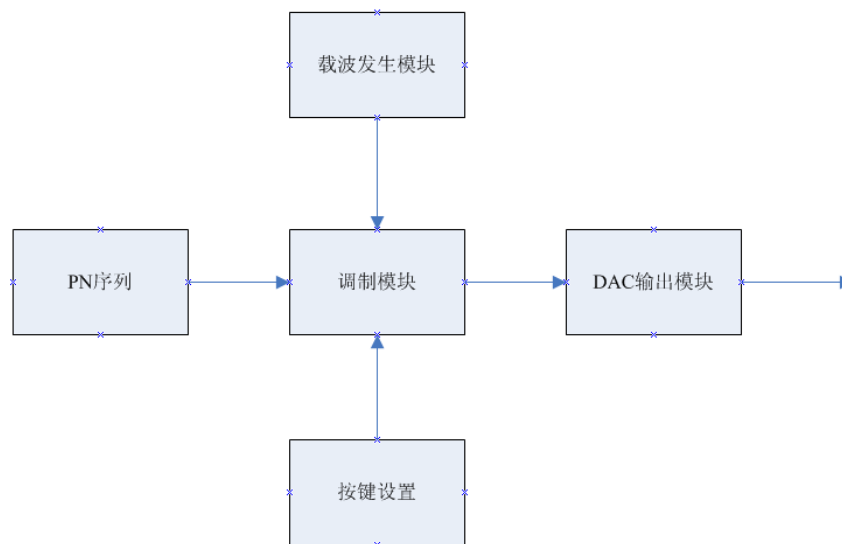


## 第二部分

### 系统组成及功能说明 /System Construction & Function Description

**功能：**通过按键控制，对伪随机序列进行 2ASK、2FSK、2PSK、2DPSK 调制，调制信号通过 DAC 输出到示波器。

**框图：**



**系统：**

#### 1、载波发生模块

将所需标准正弦信号存在 ROM 中，当需要载波时，FPGA 利用地址译码器从 ROM 中读取信号。通过调节地址译码器的参数，来实现频率、相位的调整；通过改变增益系数，来实现幅值调整的目的。

#### 2、调制模块

##### (1) 2ASK

振幅键控是利用载波的幅度变化来传递数字信息，而其频率和初始相位保持不变。在 2ASK 中，载波的幅度只有两种变化状态，分别对应二进制信息的“0”和“1”。即只需要一路正弦载波，通过输出 DAC 时的通-断键控，就能改变正弦波形的输出幅度，实现调制。

```

case (Mode)
  //2ASK
  0:
    begin
      if (m_seq==0) DAC_Data=0;
      else DAC_Data=DAC_Data_Sin;
    end

    default DAC_Data=DAC_Data_Sin;
endcase
  
```

### (2) 2FSK

频移键控是利用载波的频率变化来传递数字信息。在 2FSK 中，载波的频率随二进制基带信号在 f1 和 f2 两个频率点间变化，分别对应二进制信息的“0”和“1”。2FSK 信号的产生方法主要有两种，本组设计的方法是：在二进制基带矩形脉冲序列的控制下对两个不同的分频模式进行选通，从而通过分频器产生两个不同频率的载波，使其在每一个码元 Ts 期间输出 f1 和 f2 两个载波之一。其中 f1 频率为 2.5kHz，f2 频率为 5kHz。

```
case (Mode)
    //2FSK
    1:
        begin
            if (m_seq==0)
                begin
                    FWORD= 100000000/(2000*256);           //f2
                    PWORD=0;
                end
            else
                begin
                    FWORD= 100000000/(1000*256);           //f1
                    PWORD=0;
                end
            end
        end
    end
```

### (3) 2PSK

相移键控是利用载波的相位变化来传递数字信息，而振幅和频率保持不变。在 2PSK 中，通常用初始相位 0 和  $\pi$  分别表示二进制“1”和“0”。2PSK 信号的调制原理与 2ASK 非常相似，只是对基带矩形脉冲序列 s(t) 的要求不同，在 2ASK 中，s(t) 是单极性的，而在 2PSK 中 s(t) 是双极性的。本组设计的方法是，在二进制基带矩形脉冲序列的控制下对地址译码器中的相位参数进行选通，从而使正弦波在输出时相位发生 180° 改变。

```
case (Mode)
    //2PSK
    2:
        begin
            if (m_seq==0)
                begin
                    FWORD= 100000000/(1000*256);
                    PWORD=0;
                end
            else
                begin
                    FWORD= 100000000/(1000*256);
                    PWORD=(180*NWORD)/360;
                end
            end
        end
    end
```

#### (4) 2DPSK

2DPSK 是利用前后相邻码元的载波相对相位变化传递数字信息, 所以又称相对相移键控。与 2PSK 的绝对相移相比, 2DPSK 在解调过程中, 载波恢复的相位没有 0、 $\pi$  的模糊性, 不会导致解调过程中出现“反向工作”现象, 所以有更多的实际应用。本组只设计了 2DPSK 的解调方法: 先对二进制数字基带信号进行差分编码, 即把表示数字信息序列的绝对码变换成相对码, 然后再根据相对码进行绝对调相, 从而产生 2DPSK 信号。

```
case (Mode)
    //2DPSK
    3:
        begin
            if (m_seq_trans==0)
                begin
                    FWORD= 100000000/(1000*256);
                    PWORD=0;
                end
            else
                begin
                    FWORD= 100000000/(1000*256);
                    PWORD=(180*NWORD)/360;
                end
            end
        end
    end
```

#### 3、PN 序列

PN 序列一种伪噪声序列, 这类序列具有类似随机噪声的一些统计特性, 但和真正的随机信号不同, 它可以重复产生和处理, 故称作伪随机噪声序列。其中最基本常用的一种是最长线性反馈移位寄存器序列, 也称作 m 序列, 通常由反馈移位寄存器产生。

产生伪随机序列方法是: 带线性反馈逻辑的移位寄存器设定各级寄存器的初始状态后, 在时钟触发下, 每次移位后各级寄存器状态会发生变化。其中一级寄存器 (通常为末级) 的输出, 随着移位时钟节拍的推移会产生一个序列, 即移位寄存器序列。

输入频率为 1KHz 的脉冲信号: clk\_1kHz

输出基带信号: m\_seq

初始化位移寄存器: [7:0] shift\_reg= 8'b11111111;

由本原多项式得到 parameter POLY = 8'b10001110;

always@(posedge clk\_1kHz)

```
begin
    shift_reg[7] <= (shift_reg[0] & POLY[7]) ^
        (shift_reg[1] & POLY[6]) ^
        (shift_reg[2] & POLY[5]) ^
        (shift_reg[3] & POLY[4]) ^
        (shift_reg[4] & POLY[3]) ^
        (shift_reg[5] & POLY[2]) ^
        (shift_reg[6] & POLY[1]) ^
        (shift_reg[7] & POLY[0]); //移位寄存器状态以此发生改变
```

```
        shift_reg [6:0] <= shift_reg[7:1];  
    end  
    assign m_seq = shift_reg[0]; //生成 m 序列
```

#### 4、码型转换

将表示数字信息序列的绝对码变换成相对码（差分码）。差分波形是用相邻码元的电平的跳变和不变来表示消息代码，而与码元本身的电位或极性无关。用差分波形传送代码可以消除设备初始状态的影响。设计方法是：将随机序列与它前一级自身异或。

```
reg dpsk_code_reg=0;  
always@(negedge clk_1kHz)    dpsk_code_reg<=dpsk_code_reg^ clk_in;  
assign clk_now=dpsk_code_reg;
```

#### 5、按键设置

通过摩尔型有限状态机实现：

没有按键按下（或两个以上按键同时按下）时，模式为 2ASK 调制，LED 灯灭；

只有 Key1 按下时，模式为 2FSK 调制，LED 绿灯亮；

只有 Key2 按下时，模式为 2PSK 调制，LED 红灯亮；

只有 Key3 按下时，模式为 2DPSK 调制，LED 红、绿灯同时亮；

#### 6、DAC 输出模块

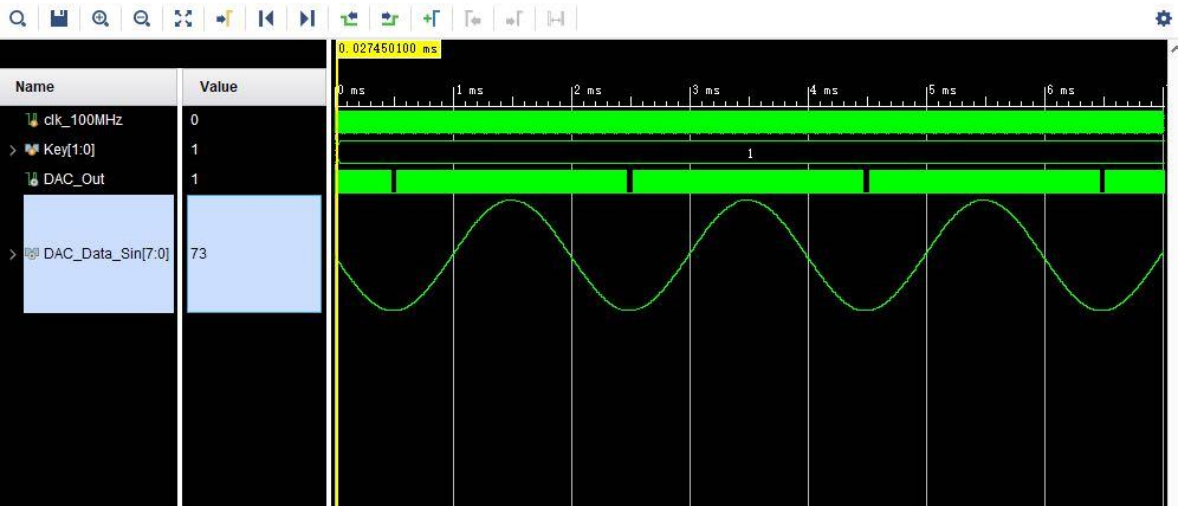
FPGA 利用地址译码器从 ROM 中读取信号，然后经过 DAC 芯片转换，得到输出。

第三部分

完成情况及性能参数 /Final Design & Performance Parameters

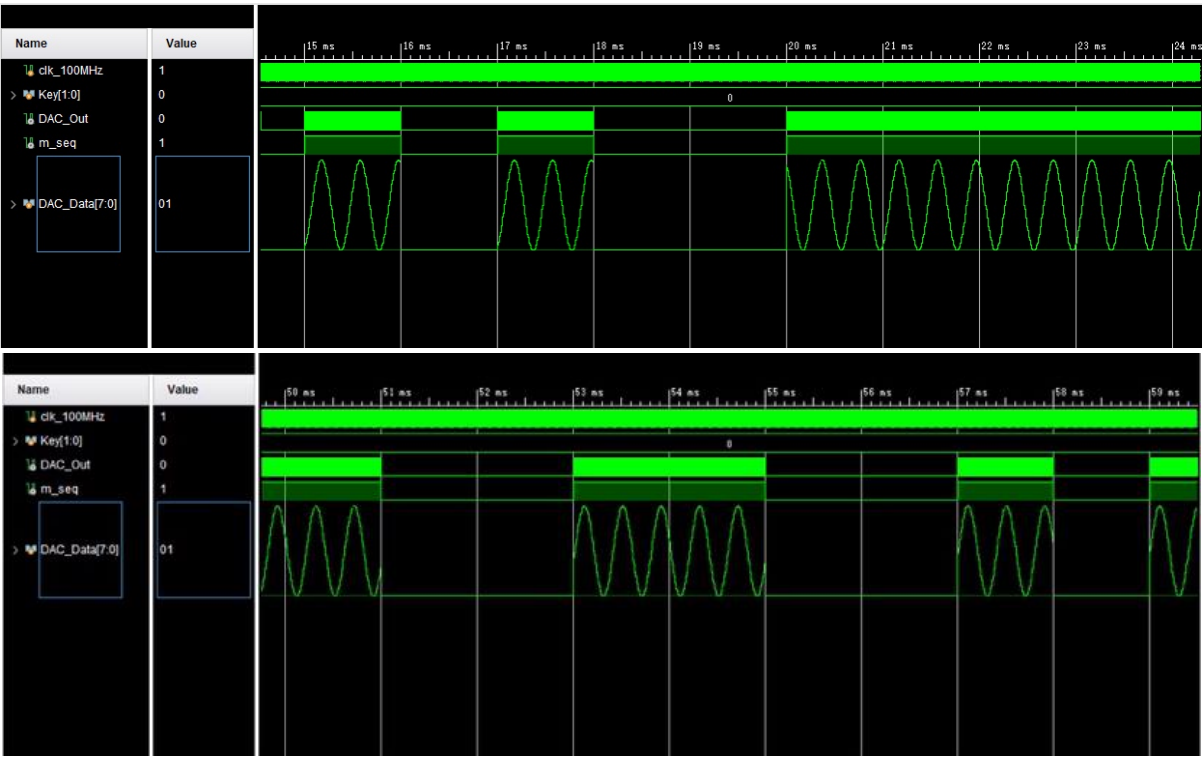
已实现功能的仿真：

1、载波发生模块



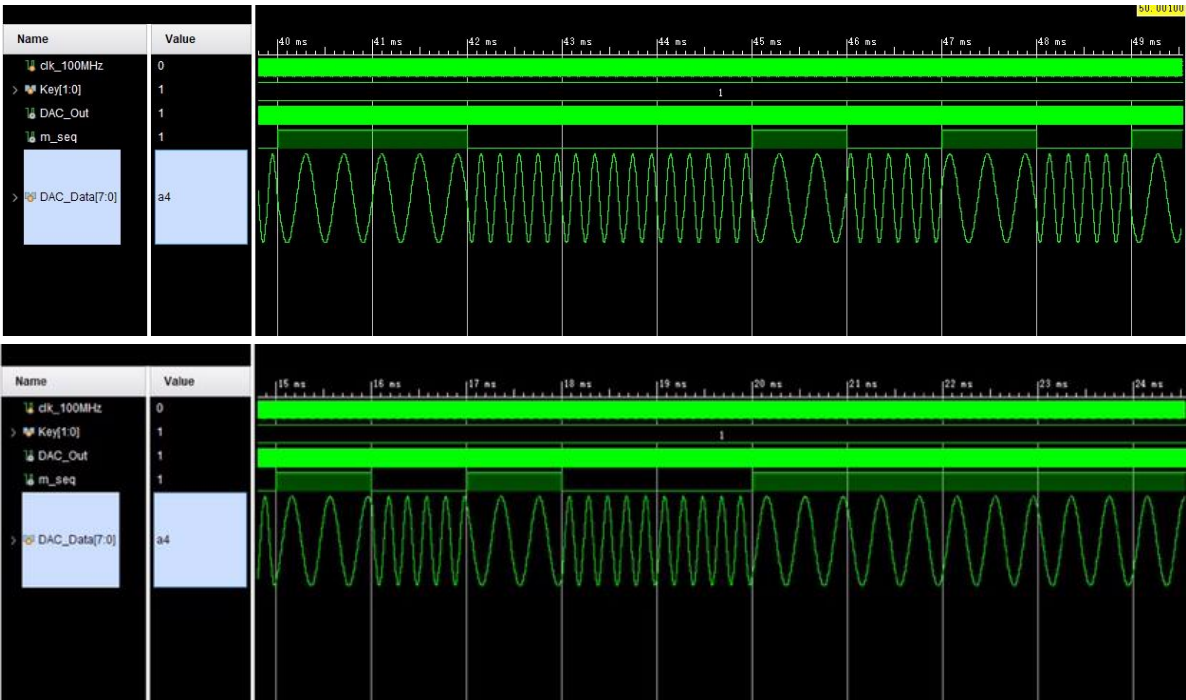
2、调制模块

(1) 2ASK

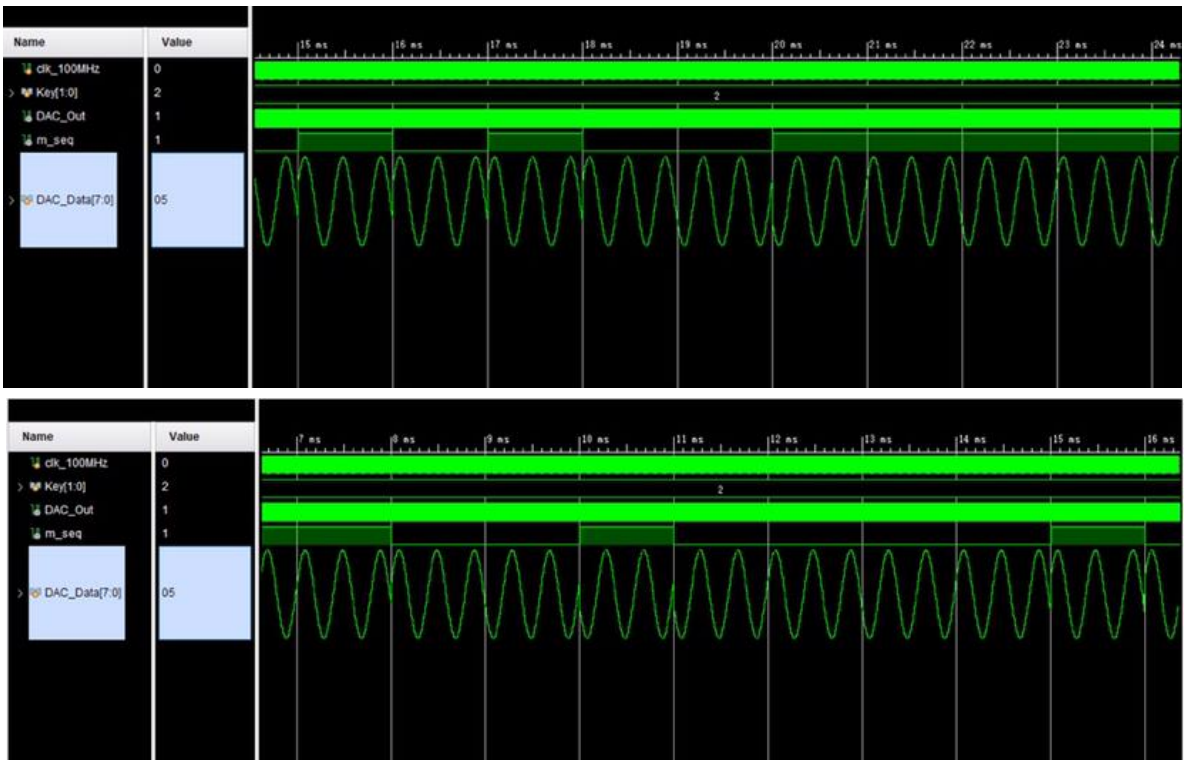




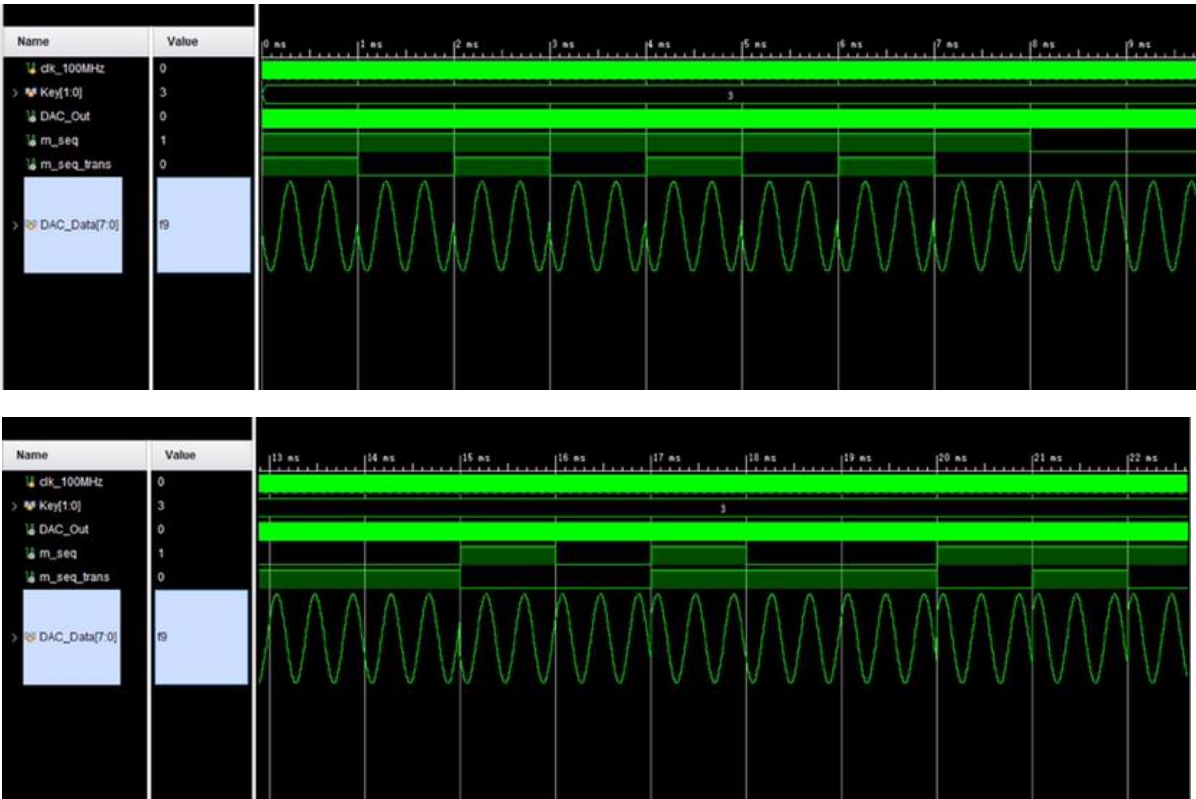
(2) 2FSK



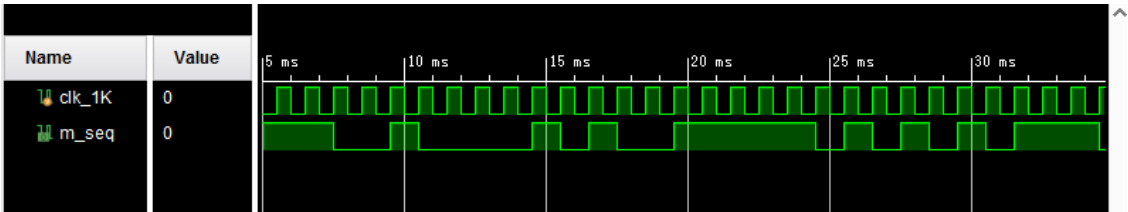
(3) 2PSK



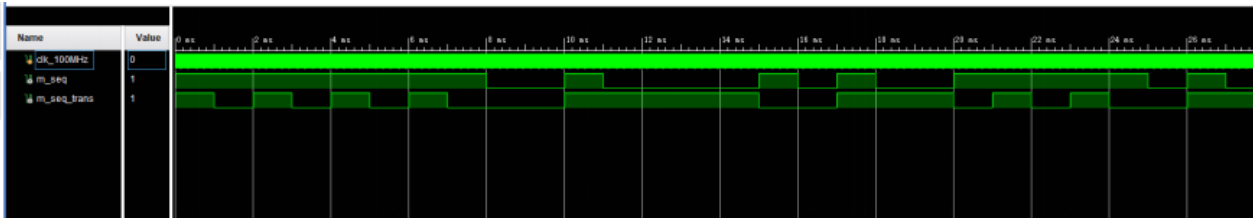
(4) 2DPSK



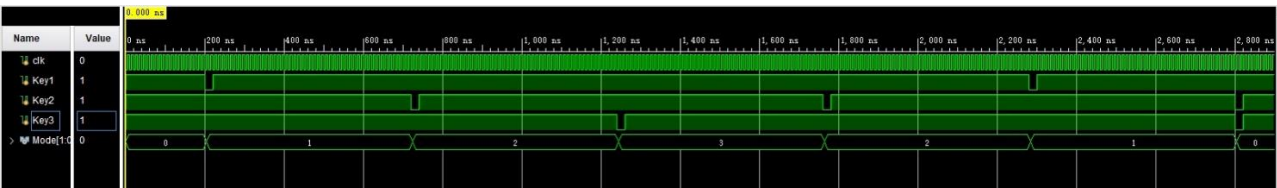
3、PN 序列



4、码型转换



5、按键设置



## 第四部分

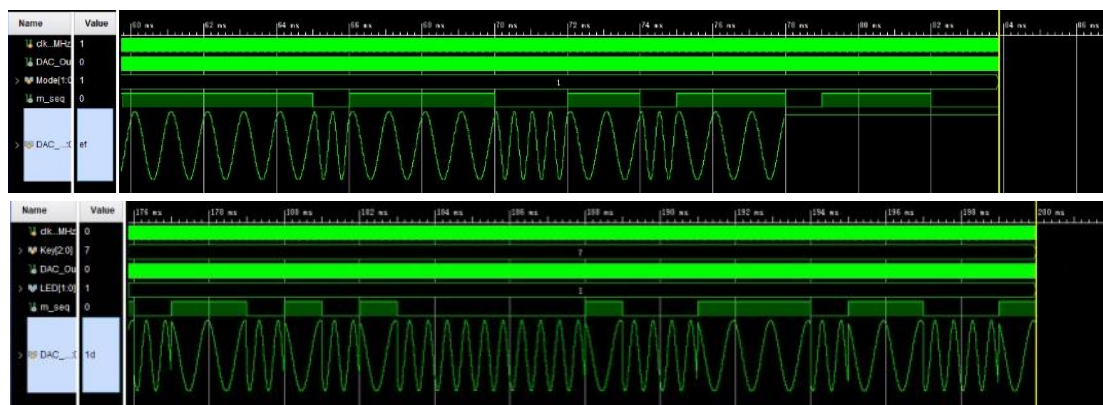
### 总结 /Conclusions

王欣茹 06017407:

还记得第一次课后的作业是写下对 FPGA 的理解,当时我对 FPGA 还不太熟悉,只是在大学数电实验和电子系统设计研讨课中用到过当中某些简单的模块,所以我参照着在赵谦博士课上记的笔记和网络上搜查的资料完成了作业。而在密集的五天实验集训和 5 天项目开发后,FPGA 对于我来说不再是百科词条上生硬的定义,也不是书上难懂的逻辑簇结构,它真正地变成了我手上能够实现各种功能的工具。编译程序也不再是简单地照搬指导书上的案例,而是像搭积木一样,建立在语法知识的基础上,从每个小模块起越搭越高。虽然我还没能掌握这块开发板上的所有芯片使用方法,但我相信在今后的学习过程中,也一定会用到 xilinx 暑期学校学习到的经验方法,继续研究 FPGA。

项目开发初立项时,我就一直想做一个和通信信号处理相关的项目,这主要是受到了上学期学习的两门通信原理课程的启发。我希望能够通过这个项目加深对所学知识的理解。这个目的也确实达到了,在学习编写 4 种调制方式的模块时,除了查阅实验指导手册,我的手边还会放一本《通信原理》随时翻查。遗憾地是,对于这个项目最终结果的呈现形式,我们在立项时并没有完全想清楚,是通过 DAC 输出到示波器,还是通过 uart 模块将数据传输到上位机。这也直接导致项目中途耽误了很多时间,结果也没能以实物形式呈现。

在编写各个模块的过程中,我也遇到过很多问题。其中最棘手的便是在仿真时,2FSK 调制信号的输出总是会在 80ms 以后呈现一条直线,无法上板测试。根据我的观察,虽然设计的基带信号、载波频率都是整数,但是随着仿真时间增加,输出波形会有相位的偏移,这种偏移对其他三种调制方式的输出几乎没有影响,但到达一定程度后就会导致 2FSK 调制信号的输出失常。于是我针对这个问题,对原代码进行了一些修改,从用序列控制地址译码器的频率变量,修改成用两个译码器生成两个独立的频率不同的正弦波,改后的波形随着仿真时间的加长相位依然有偏移,但不会再出现变成一条直线的情况了。除此之外,我也遇到了一些小问题,例如转码模块中,输出 wire 型需要



通过一个寄存器中介进行与输入进行异或运算。在 debug 这些小错误的过程里,我对之前很多实验中没有遇到的、被我忽视的 VHDL 语法现象进行了查漏补缺,有很多收获。

通过 xilinx 暑期学校的课程和项目,我不仅学习了指导书上的知识,还收获了自主完成项目的学习技能,例如从 github 上下载资料、通过 xilinx 官网 debug 等等,我相信这些在我以后的学习实践过程中将发挥重要的作用。

季雨佳 06017405:

在这两周的时间里，我了解到了 fpga 的开发流程，掌握了 vivado 的使用方法，了解了更多查找的渠道以帮助项目的开发。

前期相关课程的设置，老师们讲解的知识点，一方面让我们对已有的知识进行了巩固，另一方面拓宽了自己的知识面，对 fpga 的设计有了更深入的了解，下午的实验也让我们学以致用，在实践中检验自己的知识点。后期的项目开发也让我们能更加灵活地运用 vivado 以及 esp32 开发板的相关知识，虽然疫情期间不能回到学校，有很多不方便的地方，但这次实训意义非凡，锻炼了我们解决问题以及团队协作的能力。

非常感谢 xilinx 和东南大学的老师们给我们提供这个平台，通过这次实训，我们不仅学到了知识，也锻炼了专业技能，受益匪浅。