

# 通信系统大型实验指导书

浙江工业大学信息工程学院通信工程系

2005. 6

# 实 验 内 容

## 第一部分：简单的调试程序

实验一、键盘输入操作实验

实验二、拨码开关操作实验

实验三、A/D测试实验

实验四、D/A测试实验

## 第二部分：应用系统设计实验

实验一、全数字HDB3编解码系统

实验二、全数字2FSK调制解调系统

实验三、全数字2DPSK调制解调系统

# 第一部分：简单的调试程序

为了更好地利用实验板上资源，同时也为了能更好地做好第二部分的系列大型实验，以下设计了一些小程序，以便熟悉板上资源和相关接口。

## 实验一 键盘输入操作实验

- 1、实验目的：熟悉硬件环境。
- 2、实验要求：掌握键盘操作并编写简单的硬件描述语言。
- 3、实验设备：微机1台、示波器1台或万用表一只、稳压电源1台、实验箱一套。
- 4、实验内容：

通信系统大型实验板上有4个按键，分别为K1、K2、K3、K4，相对应的管脚为PIN46、PIN47、PIN48、PIN49，按下按键K1、K2、K3、K4（定义为key）用示波器或万用表观测被控制的输出端口（定义为tpin）变化，应有电平跳变，简单的FPGA测试程序key\_test.v如下：

```
// key_test
module key_test(
    key,
    tpin );    //输入输出端口列表

input  [3:0]key;
output [7:0]tpin;    //输入输出端口说明

assign tpin[3:0]=key;
assign tpin[7:4]=key;    //赋值语句
endmodule
```

程序中assign语句将按键key直接赋值给tpin管脚，按下K1、K2、K3、K4后，对应的tpin管脚从高电平跳到低电平，可以测试按键与输入/输出管脚的变化。管脚定义如下：

"key0"	:	INPUT_PIN	=	46	;
"key1"	:	INPUT_PIN	=	47	;
"key2"	:	INPUT_PIN	=	48	;
"key3"	:	INPUT_PIN	=	49	;
"tpin0"	:	OUTPUT_PIN	=	8	;
"tpin1"	:	OUTPUT_PIN	=	9	;
"tpin2"	:	OUTPUT_PIN	=	10	;
"tpin3"	:	OUTPUT_PIN	=	11	;
"tpin4"	:	OUTPUT_PIN	=	12	;
"tpin5"	:	OUTPUT_PIN	=	13	;
"tpin6"	:	OUTPUT_PIN	=	14	;
"tpin7"	:	OUTPUT_PIN	=	17	;

以上管脚可通过MAXPLUS软件的菜单Assign—Pin/Location/Chip来配置，芯片配置可通过MAXPLUS软件的菜单Assign—Device，选择ACEX1K—EP1K30TC144-1即可。

从管脚配置可以清楚的看出按键k1对应实验板上的PIN8、PIN12，按键k2对应实验板上的PIN9、PIN13，按键k3对应实验板上的PIN10、PIN14，按键k4对应实验板上的PIN11、PIN17。

按键开关可用作系统的复位操作。

## 实验二 拨码开关操作实验

- 1、实验目的：熟悉硬件环境。
- 2、实验要求：掌握拨码开关操作并编写简单的硬件描述语言。
- 3、实验设备：微机1台、示波器1台或万用表一只、稳压电源1台、实验箱一套。
- 4、实验内容：

通信系统大型实验板上有2组拨码开关，分别为S1、S2，相对应的管脚为114、116、117、118、119、120、121、122、130、131、132、133、135、136、137、138，拨S1或S2的ON或OFF，观测被控制的输出端口（定义为pa11和pa12）变化，对应ON输出端口为高电平，对应OFF输出端口为低电平，简单的FPGA测试程序dial\_test.v如下：

```
// dial_test
module dial_test(
    S1,
    S2,
    Pa11,
    Pa12    );    //输入输出端口列表

input    [7:0]S1;
input    [7:0]S2;
output   [7:0]pa11;
output   [7:0]pa12;    //输入输出端口说明

assign   pa11=S1;
assign   pa12=S2;    //赋值语句

endmodule
```

程序中assign语句将按键拨码开关S1和S2直接赋值给pa11和pa12两组输出管脚，管脚定义如下：

"S10"	:	INPUT_PIN	=	114	;
"S11"	:	INPUT_PIN	=	116	;
"S12"	:	INPUT_PIN	=	117	;
"S13"	:	INPUT_PIN	=	118	;
"S14"	:	INPUT_PIN	=	119	;
"S15"	:	INPUT_PIN	=	120	;
"S16"	:	INPUT_PIN	=	121	;
"S17"	:	INPUT_PIN	=	122	;
"S20"	:	INPUT_PIN	=	130	;

```

"S21"      : INPUT_PIN  = 131    ;
"S22"      : INPUT_PIN  = 132    ;
"S23"      : INPUT_PIN  = 133    ;
"S24"      : INPUT_PIN  = 134    ;
"S25"      : INPUT_PIN  = 136    ;
"S26"      : INPUT_PIN  = 137    ;
"S27"      : INPUT_PIN  = 138    ;
"pal10"    : OUTPUT_PIN = 19     ;
"pal11"    : OUTPUT_PIN = 20     ;
"pal12"    : OUTPUT_PIN = 21     ;
"pal13"    : OUTPUT_PIN = 22     ;
"pal14"    : OUTPUT_PIN = 23     ;
"pal15"    : OUTPUT_PIN = 26     ;
"pal16"    : OUTPUT_PIN = 27     ;
"pal17"    : OUTPUT_PIN = 28     ;
"pal20"    : OUTPUT_PIN = 29     ;
"pal21"    : OUTPUT_PIN = 30     ;
"pal22"    : OUTPUT_PIN = 31     ;
"pal23"    : OUTPUT_PIN = 32     ;
"pal24"    : OUTPUT_PIN = 33     ;
"pal25"    : OUTPUT_PIN = 37     ;
"pal26"    : OUTPUT_PIN = 38     ;
"pal27"    : OUTPUT_PIN = 39     ;

```

以上管脚可通过MAXPLUS软件的菜单Assign——Pin/Location/Chip来配置，芯片配置可通过MAXPLUS软件的菜单Assign—Device, 选择ACEX1K—EP1K30TC144-1即可。

从管脚配置可以清楚的看出拨码开关S1组的8个开关分别对应实验板上的Pal1组8个输出管脚，拨码开关S2组的8个开关分别对应实验板上的Pal2组8个输出管脚，用示波器或万用表可观察到相应的高低电平变化。

拨码开关可以用作系统的置位操作。

## 实验三 A/D测试实验

- 1、实验目的：熟悉硬件环境。
- 2、实验要求：掌握A/D采样操作并用硬件描述语言编写采样程序。
- 3、实验设备：微机1台、数字存储示波器1台、函数发生器一台、稳压电源1台、实验箱一套。
- 4、实验内容：

A/D模拟数字转换模块采用的是MX7821芯片，该芯片为双列直插20脚封装，管脚定义如图1所示。具体可以参考MX7821数据手册。MX7821读数据时序如图2所示。注意：MX7821最高工作频率为1MHz，也就是读信号RD周期不要小于1us。通信系统大型实验板上的MX7821没有外围芯片，有一个INPUT输入接口，输出8位数据线直接与FPGA相连。一个简单的数据采集程序MX7821.v如下：

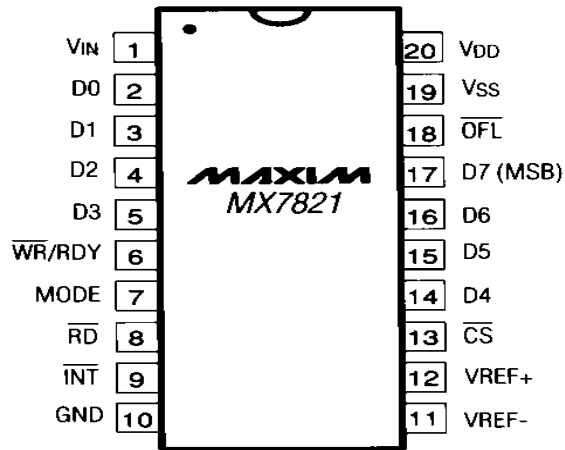


图1 MX7821芯片管脚封装图

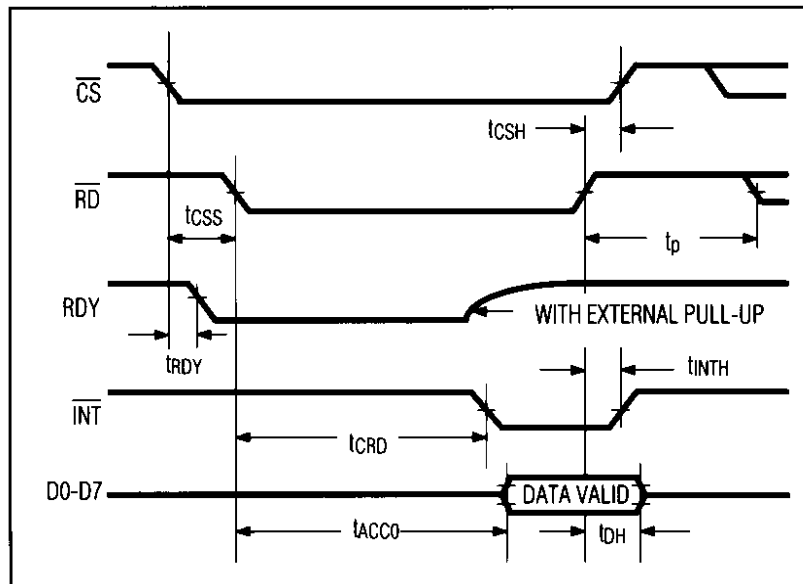


图2 MX7821读数据时序

```
//MX7821
module MX7821( clk,
               din,
               dout,
               rd);           //输入输出端口列表

input clk;
input [7:0]din;
output [7:0]dout;
output rd;                  //输入输出端口定义

reg rd;
reg [6:0]cot;
reg [7:0]dout;
wire [5:0]count;           //定义数据类型
```

```

assign  count=cot[6:1];    //赋值语句

always @(posedge clk)
    cot=cot+1;             //计数操作

always @(posedge clk)
    if (count==0 || count==1 || count==2 || count==3 || count==4 || count==5 ||
        count==6 || count==7 )
    begin
        rd=1;              //数据采样
    end
    else
    begin
        rd=0;
    end

always @(posedge clk)
    if (count>=60)
        dout=din;          //数据输出到FPGA芯片

endmodule

```

该程序将MX7821的8位输出数据读入FPGA芯片，以便FPGA采集到数据后可以作出相应的处理。

实验时，用信号发生器产生一个低频正弦波，将此低频正弦波输入到实验板上的INPUT，将FPGA的dout管脚定义到发光管的管脚上，则可观察到发光二极管作规律性变化。管脚定义如下：

"clk"	:	INPUT_PIN	=	55	;
"din0"	:	INPUT_PIN	=	95	;
"din1"	:	INPUT_PIN	=	96	;
"din2"	:	INPUT_PIN	=	97	;
"din3"	:	INPUT_PIN	=	98	;
"din4"	:	INPUT_PIN	=	99	;
"din5"	:	INPUT_PIN	=	100	;
"din6"	:	INPUT_PIN	=	101	;
"din7"	:	INPUT_PIN	=	102	;
"dout0"	:	OUTPUT_PIN	=	87	;
"dout1"	:	OUTPUT_PIN	=	86	;
"dout2"	:	OUTPUT_PIN	=	83	;
"dout3"	:	OUTPUT_PIN	=	82	;
"dout4"	:	OUTPUT_PIN	=	81	;
"dout5"	:	OUTPUT_PIN	=	80	;
"dout6"	:	OUTPUT_PIN	=	79	;
"dout7"	:	OUTPUT_PIN	=	78	;

```
"rd" : OUTPUT_PIN = 110 ;
```

以上管脚可通过MAXPLUS软件的菜单Assign——Pin/Location/Chip来配置,芯片可通过MAXPLUS软件的菜单Assign—Device,选择ACEX1K—EP1K30TC144-1配置。

A/D芯片用来采集模拟信号,采集后的数据直接输入到FPGA进行信号处理。

一个A/D采样的实例由FPGA的wave\_top.gdf文件实现。实验时,用信号发生器产生一个10KHz左右的正弦波、三角波或方波,幅度为1V左右,将此低频信号输入到实验板上的INPUT,将FPGA的输出管脚定义到D/A芯片管脚。打开顶层文件wave\_top.gdf,编译后进行下载,管脚配置如下:

```
"clk" : INPUT_PIN = 55 ;
"din0" : INPUT_PIN = 95 ;
"din1" : INPUT_PIN = 96 ;
"din2" : INPUT_PIN = 97 ;
"din3" : INPUT_PIN = 98 ;
"din4" : INPUT_PIN = 99 ;
"din5" : INPUT_PIN = 100 ;
"din6" : INPUT_PIN = 101 ;
"din7" : INPUT_PIN = 102 ;
"dout0" : OUTPUT_PIN = 68 ;
"dout1" : OUTPUT_PIN = 67 ;
"dout2" : OUTPUT_PIN = 65 ;
"dout3" : OUTPUT_PIN = 64 ;
"dout4" : OUTPUT_PIN = 63 ;
"dout5" : OUTPUT_PIN = 62 ;
"dout6" : OUTPUT_PIN = 60 ;
"dout7" : OUTPUT_PIN = 59 ;
"rd" : OUTPUT_PIN = 110 ;
```

用数字存储示波器观察输入端的INPUT和输出端的OUTPUT,可发现输出信号随输入信号的波形变化而变化。必须注意:根据A/D时序图,采样时钟必须是输入信号频率的4倍以上,否则将不能正确采集信号。

## 实验四 D/A测试实验

- 1、实验目的:熟悉硬件环境。
- 2、实验要求:掌握D/A操作并用硬件描述语言编写D/A转换程序。
- 3、实验设备:微机1台、数字存储示波器1台、稳压电源1台、实验箱一套。
- 4、实验内容:

D/A 数字模拟转换模块采用的是 DAC0832 芯片,该芯片为双列直插 20 脚封装,最高转换速率为 1MHz,管脚定义如 3 所示。在实验板中,D/A 模块输入直接与 FPGA 相连,输出接口为 OUTPUT,由于 DAC0832 芯片为电流输出,后面跟两级运算放大器将电流输出转换成电压输出,具体电路参见 DAC0832 数据手册,运放采用 TL081 芯片。

DAC0832 可直接进行数据输出完成 DA 转换,无需其他控制信号。



## Dual-In-Line and Small-Outline Packages

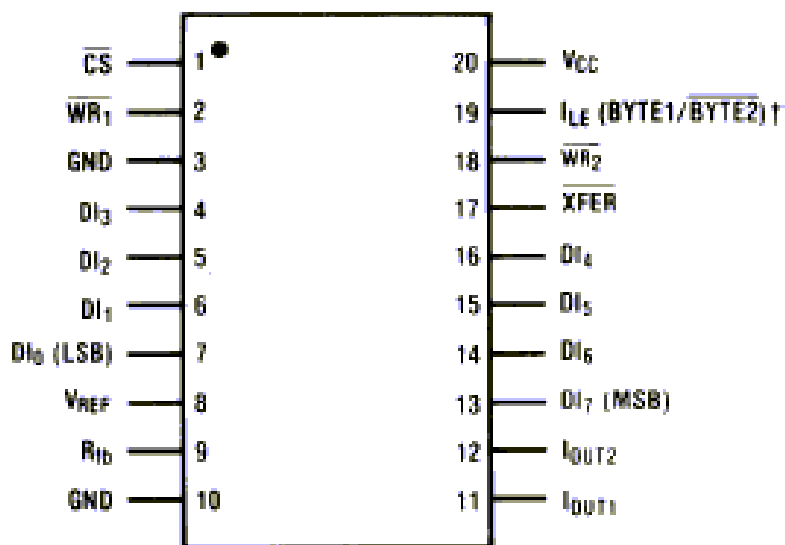


图 3 DAC0832 管脚封装图

令  $1\text{LSB} = \frac{|V_{\text{REF}}|}{128}$ , 输入输出转换表如表 1, 本实验箱采用 +5V 参考电压。

表 1 DAC0832 输入输出转换表

输入 码								理 想 输 出	
MSB				LSB				$+V_{\text{REF}}$	+5V
1	1	1	1	1	1	1	1	$V_{\text{REF}} - 1\text{LSB}$	$\approx +5\text{V}$
1	1	0	0	0	0	0	0	$V_{\text{REF}}/2$	2.5V
1	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	$-1\text{LSB}$	$\approx 0$
0	0	1	1	1	1	1	1	$- V_{\text{REF}} /2 - 1\text{LSB}$	$\approx -2.5\text{V}$
0	0	0	0	0	0	0	0	$- V_{\text{REF}} $	-5V

用D/A模块结合FPGA产生的一个简单的正弦波程序如下:

1) 在MAXPLUS软件的文本输入窗口编辑\*.mif文件, 该文件定义了正弦函数的一个周期取值, 以下是一个例子, 文件名为dac.mif, 一共有1024个取值, 每个数据用8bit数字信号存储。

```
// dac.mif
DEPTH = 1024;    //一个周期取1024个数据
WIDTH = 8;       // 每个数据8位数据宽度
ADDRESS_RADIX = DEC; //地址按10进制表示
DATA_RADIX = DEC;    //数据按10进制表示
```

CONTENT

BEGIN

0: //地址从0开始，以下是1024个数值

128 128 129 130 131 131 132 133 134 135 135 136 137 138 138 139  
140 141 142 142 143 144 145 146 146 147 148 149 149 150 151 152  
152 153 154 155 156 156 157 158 159 159 160 161 162 162 163 164  
165 165 166 167 168 168 169 170 171 171 172 173 174 174 175 176  
176 177 178 179 179 180 181 182 182 183 184 184 185 186 186 187  
188 189 189 190 191 191 192 193 193 194 195 195 196 197 197 198  
199 199 200 201 201 202 202 203 204 204 205 206 206 207 207 208  
209 209 210 211 211 212 212 213 213 214 215 215 216 216 217 217  
218 219 219 220 220 221 221 222 222 223 223 224 224 225 225 226  
226 227 227 228 228 229 229 230 230 231 231 232 232 233 233 233  
234 234 235 235 236 236 236 237 237 238 238 238 239 239 240 240  
240 241 241 241 242 242 243 243 243 244 244 244 245 245 245 245  
246 246 246 247 247 247 247 248 248 248 249 249 249 249 250 250  
250 250 250 251 251 251 251 251 252 252 252 252 252 253 253 253  
253 253 253 253 254 254 254 254 254 254 254 255 255 255 255  
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255  
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255  
255 255 255 255 255 254 254 254 254 254 254 254 254 253 253 253  
253 253 253 253 252 252 252 252 252 251 251 251 251 251 250 250  
250 250 250 249 249 249 249 248 248 248 247 247 247 247 246 246  
246 245 245 245 245 244 244 244 243 243 243 242 242 241 241 241  
240 240 240 239 239 238 238 238 237 237 236 236 236 235 235 234  
234 233 233 233 232 232 231 231 230 230 229 229 228 228 227 227  
226 226 225 225 224 224 223 223 222 222 221 221 220 220 219 219  
218 217 217 216 216 215 215 214 213 213 212 212 211 211 210 209  
209 208 207 207 206 206 205 204 204 203 202 202 201 201 200 199  
199 198 197 197 196 195 195 194 193 193 192 191 191 190 189 189  
188 187 186 186 185 184 184 183 182 182 181 180 179 179 178 177  
176 176 175 174 174 173 172 171 171 170 169 168 168 167 166 165  
165 164 163 162 162 161 160 159 159 158 157 156 156 155 154 153  
152 152 151 150 149 149 148 147 146 146 145 144 143 142 142 141  
140 139 138 138 137 136 135 135 134 133 132 131 131 130 129 128  
128 127 126 125 124 124 123 122 121 120 120 119 118 117 117 116  
115 114 113 113 112 111 110 109 109 108 107 106 106 105 104 103  
103 102 101 100 99 99 98 97 96 96 95 94 93 93 92 91  
90 90 89 88 87 87 86 85 84 84 83 82 81 81 80 79  
79 78 77 76 76 75 74 73 73 72 71 71 70 69 69 68  
67 66 66 65 64 64 63 62 62 61 60 60 59 58 58 57  
56 56 55 54 54 53 53 52 51 51 50 49 49 48 48 47  
46 46 45 44 44 43 43 42 42 41 40 40 39 39 38 38  
37 36 36 35 35 34 34 33 33 32 32 31 31 30 30 29

```

29 28 28 27 27 26 26 25 25 24 24 23 23 22 22 22
21 21 20 20 19 19 19 18 18 17 17 17 16 16 15 15
15 14 14 14 13 13 12 12 12 11 11 11 10 10 10 10
9 9 9 8 8 8 8 7 7 7 6 6 6 6 5 5
5 5 5 4 4 4 4 3 3 3 3 3 2 2 2
2 2 2 2 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 2 2 2
2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5
5 5 5 6 6 6 6 7 7 7 8 8 8 8 9 9
9 10 10 10 10 11 11 11 12 12 12 13 13 14 14 14
15 15 15 16 16 17 17 17 18 18 19 19 19 20 20 21
21 22 22 22 23 23 24 24 25 25 26 26 27 27 28 28
29 29 30 30 31 31 32 32 33 33 34 34 35 35 36 36
37 38 38 39 39 40 40 41 42 42 43 43 44 44 45 46
46 47 48 48 49 49 50 51 51 52 53 53 54 54 55 56
56 57 58 58 59 60 60 61 62 62 63 64 64 65 66 66
67 68 69 69 70 71 71 72 73 73 74 75 76 76 77 78
79 79 80 81 81 82 83 84 84 85 86 87 87 88 89 90
90 91 92 93 93 94 95 96 96 97 98 99 99 100 101 102
103 103 104 105 106 106 107 108 109 109 110 111 112 113 113 114
115 116 117 117 118 119 120 120 121 122 123 124 124 125 126 127 ;
END;

```

2) 在MAXPLUS软件的文本输入窗口编辑ROM文件来调用dac.mif文件，下面定义了一个romdac.v文件。

```

//romdac.v文件
module romdac (
    address,
    inclock,
    q);          //输入输出端口列表

    input  [9:0]  address;
    input      inclock;
    output [7:0]  q;      //输入输出端口说明

    wire [7:0] sub_wire0;
    wire [7:0] q = sub_wire0[7:0];    //定义数据类型

    lpm_rom lpm_rom_component (
        .address (address),
        .inclock (inclock),
        .q (sub_wire0));    //调用maxplus库文件lpm_rom，产生一个新的元件

```

```

defparam                                //参数重定义
    lpm_rom_component.lpm_width = 8,      //8位数据宽度
    lpm_rom_component.lpm_widthad = 10,   //1024个数据
    lpm_rom_component.lpm_address_control = "REGISTERED",
    lpm_rom_component.lpm_outdata = "UNREGISTERED",
    lpm_rom_component.lpm_file = "dac.mif"; //调用dac.mif文件

endmodule

```

3) 编写一个顶层文件，实现正弦波的功能，例子见dac0832.v文件

```

//dac0832.v
module dac0832(
    clk ,
    dat );                                //输入输出端口列表

input    clk        ;
output    [7:0]dat    ; //输入输出端口说明

reg    [14:0]count ;
reg    [7:0]dat    ;
reg    [7:0]datt    ; //定义数据类型

always @(posedge clk)
    count=count+1 ; //计数器，给地址赋值

always @(posedge clk)
    dat=datt ; //赋值

romdac
U1(
    .address(count[14:5]) ,
    .inclock(clk) ,
    .q(datt) ); //调用romdac.v文件

endmodule

```

4) 管脚定义可通过MAXPLUS软件的菜单Assign——Pin/Location/Chip来配置，芯片可通过MAXPLUS软件的菜单Assign—Device, 选择ACEX1K—EP1K30TC144-1配置。管脚定义如下：

```

"clk"           : INPUT_PIN  = 55      ;
"dat0"          : INPUT_PIN  = 68      ;
"dat1"          : INPUT_PIN  = 67      ;
"dat2"          : INPUT_PIN  = 65      ;
"dat3"          : INPUT_PIN  = 64      ;
"dat4"          : INPUT_PIN  = 63      ;

```

```

"dat5"          : INPUT_PIN  = 62      ;
"dat6"          : INPUT_PIN  = 60      ;
"dat7"          : INPUT_PIN  = 59      ;

```

5) 顶层文件编译好后就可以下载。实验时，将实验箱接上电源，下载后用示波器观察D/A的OUTPUT输出脚，可观察到稳定的正弦波形。

如果想产生其它函数的输出波形，可修改dac.mif文件中的数据，分别可产生方波、矩形波、三角波等各种波形，因此D/A芯片与FPGA结合可制作函数信号发生器。

## 第二部分：应用系统设计实验

应用系统设计实验是为了培养学生通过所学知识开拓和获取新知识，独立分析并解决问题，进一步培养学生与他人合作共事、组织协同工作的能力。为了使学生获得动手经验，通过开设以下三组系统大型实验使学生掌握数字通信系统设计的理论和实践知识。学会运用各方面综合知识，设计并实现一个通信系统。

### 实验一、全数字HDB3编解码系统

#### 一. 实验目的：

熟悉HDB3编解码的原理；

掌握MAX+plu II 开发软件的运用，在该软件下熟练的运用多种输入方式完成各种电路设计的要求；

初步掌握VHDL语言，能够运用该语言编写简单的程序，完成设计要求；

熟悉对PLD的下载和仿真，学会观察测试结果的正确性；

学会运用各方面知识，设计并实现一个系统。

#### 二. 实验要求：

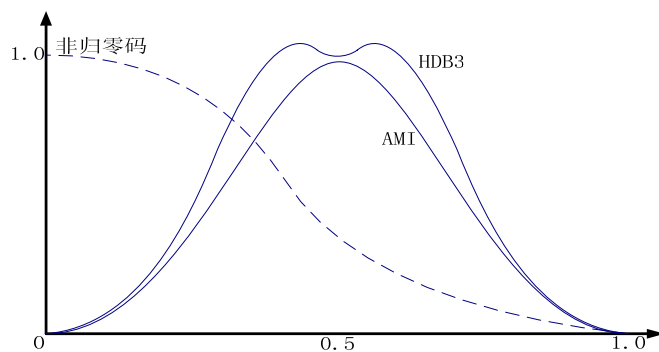
使用MAX+plu II 软件，用M序列发生器作为信号源设计全数字HDB3编解码器，速率可自定，并在实验箱上调试出编码和解码波形，比较信号源和解调后的信号波形。

#### 三. 实验设备：

MAX+plus II 开发系统、微机1台、示波器1台、稳压电源1台、万用表1只、实验箱1套

#### 四. 实验原理：

HDB<sub>3</sub>(High Density Bipolar三阶高密度双极性)码是一种AMI码的改进型，又称四连“0”取代码，功率谱如图（1）所示：



图（1）二进制数字基带信号的功率密度谱

在AMI码中，如果连续的较长的一段序列为“0”码，则在接收端会因为长时间无交替变化波形的控制而失去同步信号。

HDB3码除具有AMI码功率谱中无直流分量，可进行差错检错等优点外，还克服了AMI码当信息中出现连“0”码时定时提取困难的缺点。此外，HDB3码还具有频谱能量主要集中在基波频率以下，占用频带较窄等特点。

### 1、HDB3编码原理

在消息的二进制代码序列中：

- （1）当连“0”码的个数不大于3时，HDB3编码规律与AMI码相同，即“1”码变为“+1”、“-1”交替脉冲；
- （2）当代码序列中出现4个连“0”码或超过4个连“0”码时，把连“0”段按4个“0”分节，即“0000”，并使第4个“0”码变为“1”码，用V脉冲表示，这样可以消除长连“0”现象。为了便于识别V脉冲，使V脉冲极性与前一个“1”脉冲极性相同，这样就破坏了AMI码极性交替的规律，所以V脉冲为破坏脉冲，把V脉冲和前3个连“0”称为破坏节“000V”；
- （3）为了使脉冲序列仍不含直流分量，则必须使相邻的破坏点V脉冲极性交替；
- （4）为了保证（2）（3）两条件成立，必须使相邻的破坏点之间有奇数个“1”码。如果原序列中破坏点之间的“1”码为偶数个，则必须补为奇数，即将破坏节中的第一个“0”码变为“1”，用B脉冲表示，这时的破坏节变为“B00V”形式。B脉冲极性与前一“1”脉冲极性相反，而B脉冲极性与V脉冲极性相同，如：

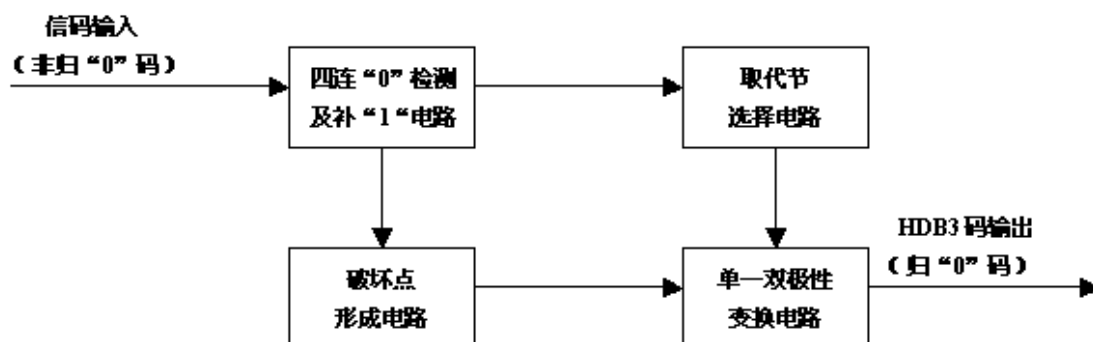
代码序列：	1000	0	1000	0	1	1	000	0	1	1
AMI码：	-1000	0	+1000	0	-1	+1	000	0	-1	+1
HDB3码：	-1000	-V	+1000	+V	-1	+1	-B00	-V	+1	-1

表(1)是HDB3编码取代规则：

前一个“V”码的极性	前一个取代码后“1”的个数	
	奇 数	偶 数
V+	000V-	B-00V-
V-	000V+	B+00V+

### 2、编码模块

编码模块的原理方框图如下图(2)。



图(2) 编码模块的原理方框图

#### (1) 连“0”检测及补“1”电路

单极性信码进入本电路，首先检测有无四连“0”码，没有四连“0”时，信码不改变地通过；有四连“0”时，在第四个“0”码出现时，将一个“1”码放入信号中，取代第四个“0”码，补入“1”码称为V码。

#### (2) 破坏点形成电路

将补放的“1”码变成破坏点。方法是在取代节内第二位处再插入一个“1”码，使单一双极性变换电路多翻转一次，后续的V码就会与前面相邻的“1”码极性相同，破坏了交替反转的规律，形成了破坏点。

#### (3) 取代节选择及补B码电路

电路计算两个V脉冲之间的“1”码个数，若为奇数，则用“000V”作为取代节；若为偶数，则将“000V”码流中的第一个“0”改为“1”，即此时用“B00V”作为取代节。

#### (4) 单一双极性变换电路

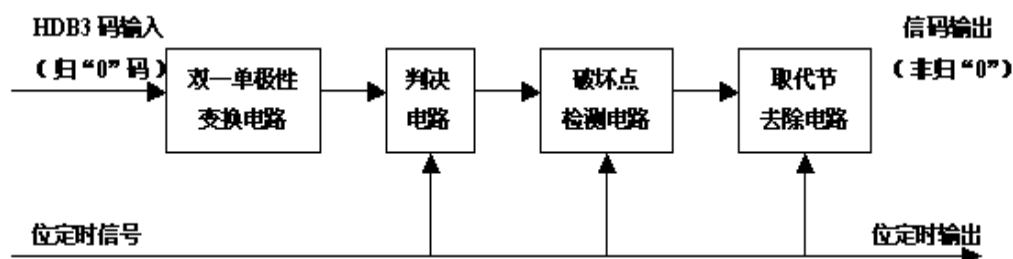
电路中的除2电路对加B码、插入码、V码的码序计数，它的输出控制已经加入取代节的信号码流，使其按交替翻转规律分成两路，再将此两路合成双极性信号。

### 3、HDB3译码原理

虽然HDB3码的编码规则比较复杂，但译码却比较简单。从HDB3编码原理看出，每一个破坏符号V总是与前一非0符号同极性（包括B在内），译码的原理是检测数据中是否连续出现两个+1或-1，如果有就表明是破坏节，将这两个+1或-1都变成“0”，并将两位二进制数据用一位二进制数据表示。

### 4、译码模块

译码电路完成恢复定时并再生码的功能，原理框图如图(3)所示，各部分功能如下：



图(3) 解码部分的原理方框图

### （1） 双—单极性变换电路

从传输线来的H D B 3 码加入本电路，经A/D转换后形成单极性归零的脉冲。

### （2） 判决电路

选用合适的判决电平以去除信码经信道传输之后引入的干扰信号。信码经判决电路之后成为半占空的两路信号，相加后成为一路单极性归“0”信码，送到定时恢复电路和码再生电路。

### （3） 破坏点检测电路

将判决后的数据逐次通过4个移位寄存器，再用一个寄存器来存储前一个+1或-1码，将每一个刚接收到的数据与这个寄存器比较，如果内容相同就表示是破坏节，即V码。

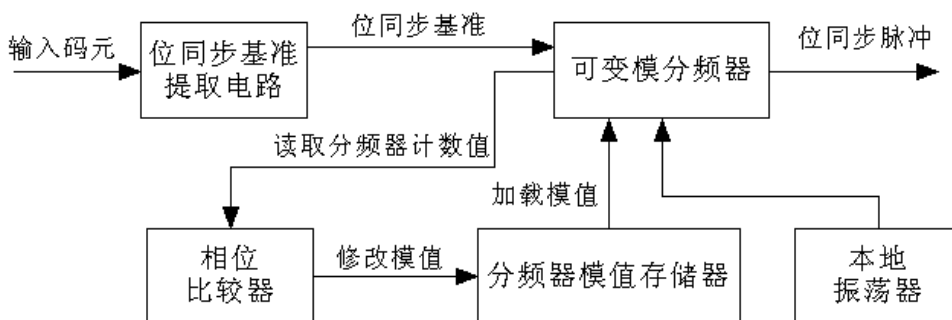
### （4） 取代节去除电路

在V码出现的时刻将信码流中的V码及它前面的第三位码置为“0”，去掉取代节之后，再将信号整形即可恢复原来的信码。破坏点检测与取代节去除电路一起完成信码再生功能。

## 5、位同步设计

利用FPGA实现位同步， 数字锁相环DPLL的原理框图如图（4）所示。

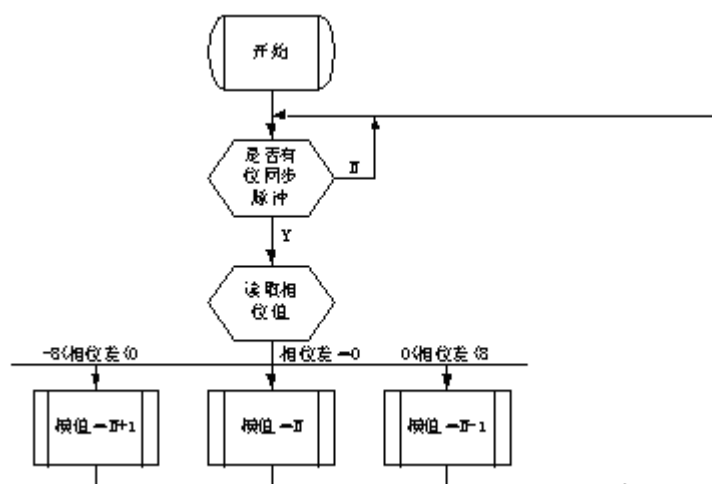
数字锁相环电路主要是由鉴相器、序列滤波器、可控分频器和时钟源组成，完成对输入定时信号提取、数字滤波和定时综合。其中可控分频器模值设计为 $N+1$ 、 $N$ 、 $N-1$ 三种；序列滤波器（相当于图中的分频器模值存储器）受鉴相器（相当于图中的相位比较器）的输出控制。鉴相器采用的是微分型导前一滞后型鉴相器。时钟源使用系统时钟。



图(4) 同步锁相环框图

锁相环的算法如图（5）所示。由图可见，当提取位同步基准脉冲后，锁相环读取表示位同步脉冲可变模分频器的相位计数值，如果相位差在0到8（可以自己设定，与锁相环参数有关）之间，则加大或减小分频器模值存储器；如果相位差为零，则保持原来的分频器模值大小。这样，不断调整的结果，可以使本地振荡器的相位与从线路码提取的位同步基准脉冲同相。





图（5） 位同步锁相环算法

### （1）位同步基准提取

位同步基准提取模块的作用为检测二进制波形的0→1的跳变，并输出一个窄的同步基准脉冲。

### （2）鉴相器

设码元速率为 $f_b$ ，本地振荡器频率为 $Nf_b$ ，即可变模分频器的初始分频值为 $N$ 。当位同步基准到来时，如果读取的分频器值为 $N/2$ ，则判为同步状态；大于 $N/2$ 时判为滞后状态；小于 $N/2$ 时判为超前状态。

### （3）可变模分频器

分频器设计长度可自己设定。

## 五. 实验内容：

1. 根据HDB3编码规则和译码原理框图，在MAX+plus II 环境下设计编解码电路及其他外围电路模块。
2. 观察仿真波形图，列出6阶M序列的63位信码，参照HDB3编码规则，将其编成HDB3码，再与调制后的编码输出对照，看是否符合编码规则。最后观测调制、解调总框图，看总体结果。
3. 硬件下载：在仿真波形正确后，通过MAX+plus II 软件的下载功能将电路模块下载到FPGA实验板上，通过示波器观察经D/A转换后的调制输出。这样可以更清楚直接的验证HDB3编解码电路设计的正确性。
4. 总结实验结果，完成实验报告。

## 六. 注意事项：

1. 在电路设计中注意时钟频率，如果下载的时候编解码由两块FPGA板子一同下载，则要注意时钟频率的一致性。
2. 下载时注意电源的输入电压和极性，以免烧坏实验箱。

## 实验二、全数字2FSK调制解调系统

### 一. 实验目的

掌握2FSK调制解调的原理及其实现方法；

掌握MAX+plus II 开发软件的运用，在该软件下熟练的运用多种输入方式完成各种电路设计的要求；

初步掌握VHDL语言，能够运用该语言编写简单的程序，完成设计要求；

熟悉对PLD的下载和仿真，学会观察测试结果的正确性；

学会运用各方面知识，设计并实现一个系统。

### 二. 实验要求

使用MAX+plus II 软件，用M序列发生器作为信号源设计全数字2FSK调制解调系统，速率可自定，并在实验箱上调试出调制和解调波形，并将信号源和解调结果进行比较。

### 三. 实验设备

MAX+plus II 开发系统、微机1台、示波器1台、稳压电源1台、万用表1只、实验箱1套

### 四. 实验原理：

FSK (Frequency Shift Keying) ---移频键控，或称数字频率调制。是数字通信中使用较早的一种调制方式。数字频率调制的基本原理是利用载波的频率变化来传递数字信息。在数字通信系统中，这种频率变化不是连续而是离散的。

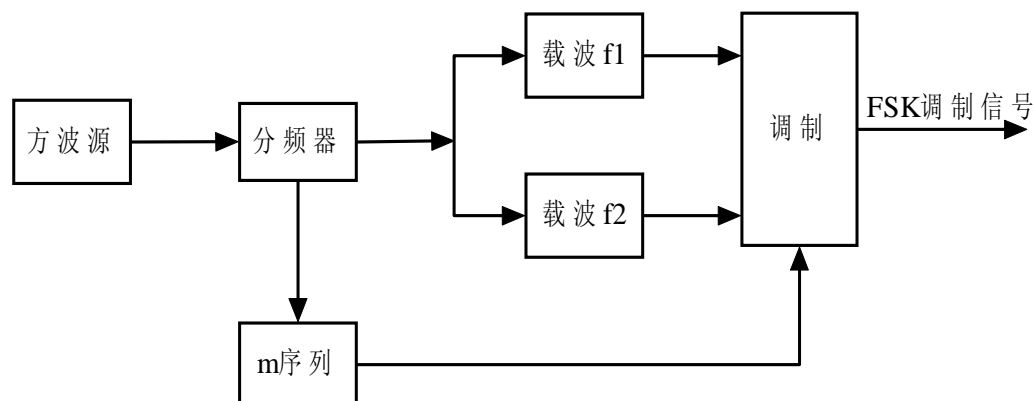
FSK 广泛应用于低速数据传输设备中，根据国际电报和电话咨询委员会 (ITU-T) 的建议，传输速率为 1200 波特以下的设备一般采用 FSK 方式传输数据。

FSK 具有调制方法简单易于实现，解调不需要恢复本地载波、可以异步传输、抗噪声和抗衰落性能较强等特点。由于这些原因，FSK 是在模拟电话网上用来传输数据的低速、低成本异步调制解调的一种主要调制方式。

在一个 FSK 系统中，发端把基带信号的变化转换成对应的载频变化，而在收端则完成与发端相反的转换，将载波频率的变化变为基带信号的变化。

#### 1、FSK调制原理

FSK 调制总体框图：



(一) 方波源

由FPGA内部晶振构成，频率为32.768MHz，为了使后续的A/D和D/A能在1MHz正常工作，方波源必须进行分频，降到1MHz以下。

## （二）分频器

由不同进制的计数器构成不同模的分频器。由于载波的频率要高于m序列信号源的频率，因此产生m序列的分频器模要比产生载波的分频器模大，且载波1和载波2的频率不同，它们所用的分频器的模也不同。

## （三）m序列

由伪随机信号m序列构成，可根据其生成多项式选择不同阶数的伪随机序列。

## （四）载波

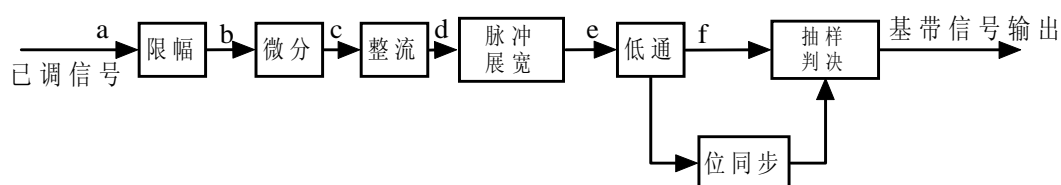
载波f1和载波f2通过D/A映射表来设计，一周取8点或16点或32点都可以，自己决定。

## （五）调制

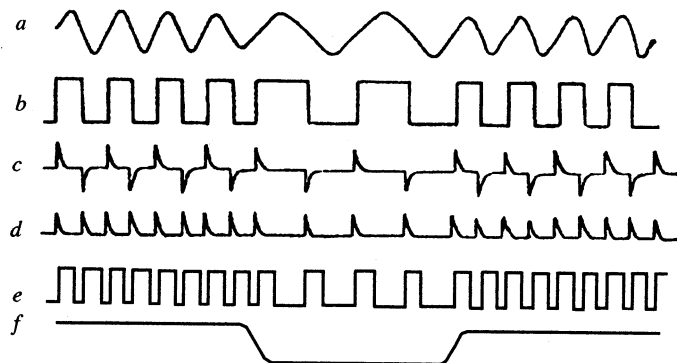
采用二选一数据选择器即可完成。

## 2、FSK 解调原理

FSK 解调总体框图如下：



各部分的对应的波形原理为：



经抽样判决后，得到规整的信码。

## 3、位同步提取

与HDB3位同步原理相同，略。

## 五. 实验内容：

1. 根据2FSK调制解调原理框图，在MAX+plus II环境下设计调制解调电路及其他外围电路模块。

2. 观察仿真波形图，列出m序列信码，将它作为2FSK的信号源，观察调制解调后的信码是否一致。

3. 硬件下载：在仿真波形正确后，通过MAX+plus II 软件的下载功能将电路模块下载到FPGA实验板上，通过示波器观察经D/A转换后的调制输出并对照解调后的信码与输入信码，验证设计结果的正确性。

4. 总结实验结果，完成实验报告。

## 六. 注意事项：

1. 在电路设计中注意时钟频率，如果下载的时候编解码由两块FPGA板子一同下载，则要注意时钟频率的一致性。

2. 下载时注意电源的输入电压和极性，以免烧坏实验箱。

## 实验三、全数字2DPSK调制解调系统

### 一. 实验目的：

掌握2DPSK调制与解调的基本原理和实现方法；

掌握MAX+plus II 开发软件的运用，在该软件下熟练的运用多种输入方式完成各种电路设计的要求；

初步掌握HDL语言，能够运用硬件描述语言编写简单的程序，完成设计要求；

熟悉对PLD的下载和仿真，学会观察测试结果的正确性；

学会运用各方面知识，设计并实现一个系统。

### 二. 实验要求：

使用MAX+plus II 软件，用M序列发生器作为信号源设计全数字2DPSK调制解调系统，速率可自定，并在实验箱上调试出调制和解调波形，并将信号源和解调结果进行比较。

### 三. 实验设备

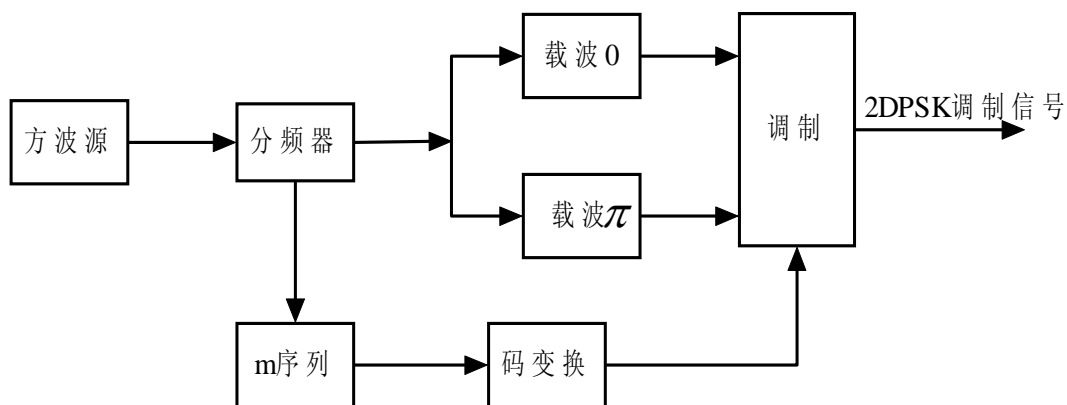
MAX+plus II 开发系统、微机1台、示波器1台、稳压电源1台、万用表1只、实验箱1套。

### 四. 实验原理：

2DPSK全称是二进制相对（差分）移相键控，其基本原理是利用前后相邻码元的相对载波相位值去表示数字信息。即用相位偏移 $\Delta\Phi$ 取不同的值（如0或 $\pi$ ）去表示数字信息的“1”或“0”，其解调采用差分相干解调法。它的特点是可以避免倒 $\pi$ 现象的发生。

#### 1、2DPSK 调制原理

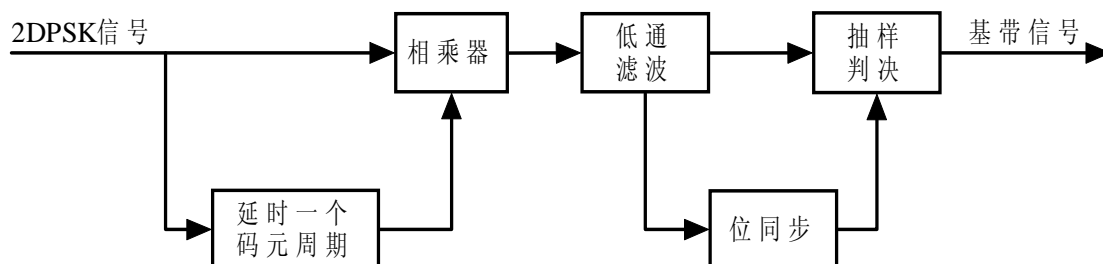
2DPSK 调制原理框图如下：



其框图结构与 2FSK 不同的地方只是多了码变换器，另外，两个载波的频率相同，只是相位相差  $\pi$ 。码变换器实现差分编码功能。

## 2、2DPSK 解调原理

采用差分相干解调法，它直接比较前后码元的相位差，故又称为相位比较法解调，其原理框图如下。由于此时的解调同时完成码变换的作用，故无需码反变换器。



### (一) 延时电路

注意必须延时一个码元周期，而不是一个载波周期。

### (二) 相乘器

由于输入为 8bit 数据，延时一个码元后的数据也是一个 8bit 的数据，相乘后为 16bit 数据，必须进行数据优化。

### (三) 低通滤波器

设计一个低通滤波器，滤除 2 倍频分量信号，让有用信号能顺利通过。

## 3、位同步提取

与 HDB3 位同步原理相同，略。

## 五. 实验内容：

1. 根据 2DPSK 调制解调原理框图，在 MAX+plus II 环境下设计调制解调电路及其他外围电路模块。
2. 观察仿真波形图，列出 M 序列信码，将它作为 2DPSK 的信号源，观察调制解调后的信码是否一致。

3. 硬件下载：在仿真波形正确后，通过MAX+plus II 软件的下载功能将电路模块下载到FPGA实验板上，通过示波器观察经D/A转换后的调制输出并比较调制前后和解调后的信码是否一致，验证设计结果的正确性。

4. 总结实验结果，完成实验报告。

## 六. 注意事项：

1. 在电路设计中注意时钟频率，如果下载的时候编解码由两块FPGA板子一同下载，则要注意时钟频率的一致性。

2. 下载时注意电源的输入电压和极性，以免烧坏实验箱。

## 参考文献

- 【1】樊昌信, 詹道庸, 徐炳祥, 吴成柯. 通信原理(第五版). 国防工业出版社
- 【2】隗永安. 现代通信原理. 西南交通大学出版社
- 【3】罗伟雄, 韩力, 原东昌, 丁志杰. 通信原理与电路. 北京理工大学出版社
- 【4】赵雅兴. FPGA原理、设计与应用. 天津大学出版社
- 【5】袁俊泉, 孙敏琪, 曹瑞编著. Verilog HDL数字系统设计及其应用. 西安电子科技大学出版社
- 【6】侯伯亨, 顾新. VHDL硬件描述语言与数字逻辑电路设计. 西安电子科技大学出版社
- 【7】通信工程专业实验讲义. 浙江工业大学通信实验室
- 【8】杨静. 采用在系统可编程逻辑器件实现HDB3编解码. 北京联合大学学报. 总第31期