中国科学技术大学计算机学院

《数字电路实验》报告

实验题目：FPGA 实验平台及 IP 核使用

学生姓名：张郑飞扬

学生学号：PB21071416

完成日期：2022.11.24

计算机实验教学中心制

2020 年 09 月

【实验题目】

FPGA 实验平台及 IP 核使用

【实验目的】

熟悉 FPGAOL 在线实验平台结构及使用

掌握 FPGA 开发各关键环节

学会使用 IP 核（知识产权核）

【实验环境】

VLAB 平台： vlab.ustc.edu.cn

FPGAOL 平台： fpgaol.ustc.edu.cn

Vivado

Logisim

【实验过程】

Step1.FPGAOL 实验平台介绍

阅读实验手册，了解 FPGAOL 实验平台功能特性。

Step2.外设工作原理介绍

阅读实验手册，了解了 FPGAOL 平台的各种外设工作原理。

时钟信号：与 FPGA 芯片的 E3 管脚直接相连，在开发板上电后，会持续的为 FPGA 芯片提供一个 100MHz 频率的时钟信号，不需要用户进行操作。

LED：FPGA 管脚为高电平时对应 LED 点亮，为低电平时则熄灭。拨动开关作为输入设备，当开关推到上方后，对应 FPGA 管脚输入高电平，当开关拉下来之后，对应 FPGA 管脚输入低电平；按键的

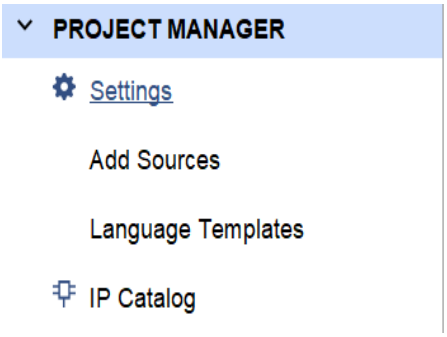
工作原理与拨动开关类似，区别在于：按键按下时向 FPGA 芯片输入高电平信号，松开时输入低电平信号，按键默认处于松开状态。还了解了平台对数码管和 LED 管脚的使用规则。

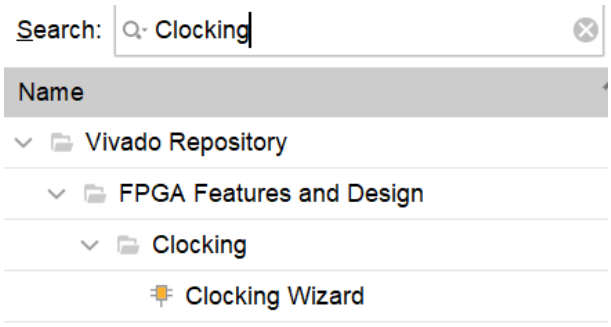
Step3. 使用时钟管理单元 IP 核

在 FPGA 开发中，有很多常用功能的模块是不需要自己开发的，用户可以复用第三方开发好的模块，这种模块被称为 IP 核。

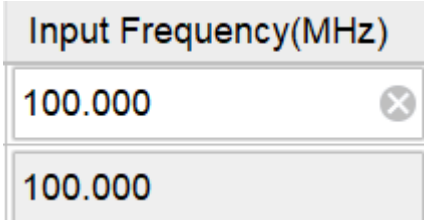
首先学习使用时钟管理单元 IP 核：

如果设计中确实需要不同频率的时钟信号应该通过时钟管理单元 IP 核生成。 首先，点击“IP catalog”，在对应窗口中选中“Clocking Wizard”并双击。





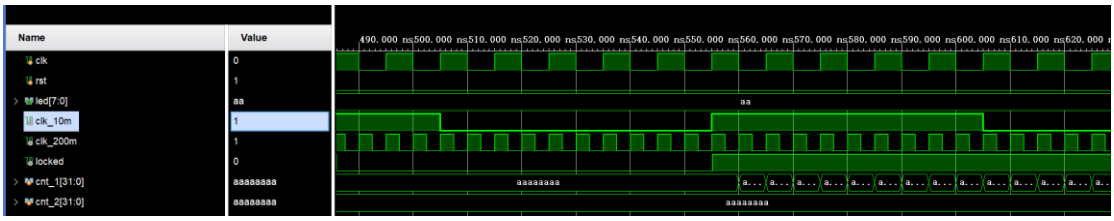
在弹出的窗口中， 对其进行设置， 输入时钟频率设置为 100MHz，输出时钟有两个，分别为 10MHz 和 200MHz。



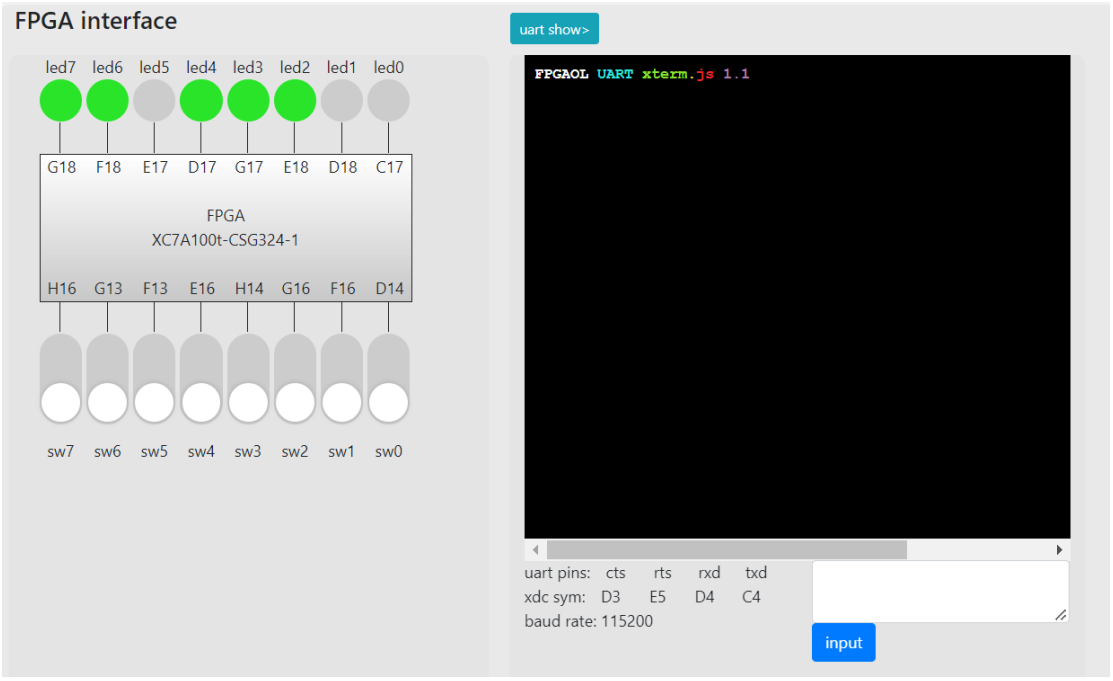
Output Clock	Port Name	Output Freq (MHz) Requested
clk_out1	clk_out1	10.000
clk_out2	clk_out2	200.000

设置完成后点击确认按钮，生成 IP 核。

新建源文件，录入实验手册所给的代码，首先进行波形仿真，结果如下图：



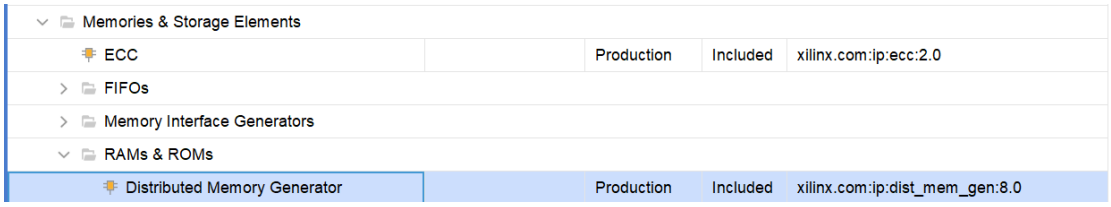
烧写到 FPGA 上，如图：



符合代码所呈现的规则，左边四个 led（led[7:4]）闪烁频率快，右边四个 led（led[3:0]）闪烁频率慢。

Step4. 使用片内存储单元

Vivado 中也提供了存储器相关的 IP 核，下面进行该部分的学习。以“Distributed Memory”为例进行学习。

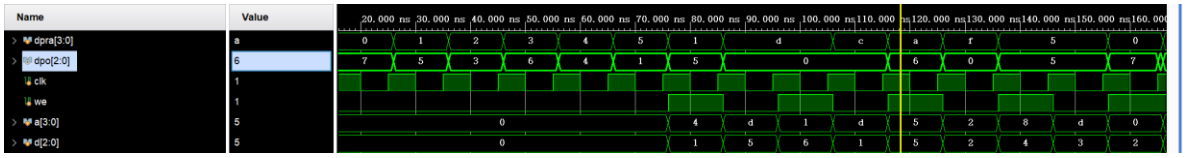


在存储器参数设置页面，用户可以对 IP 核名称、存储单元深度和位宽、存储器类型等进行设置。将存储器深度设为 16=2^4，数据位宽为 3 位，因此地址信号为 4 位，数据端口位宽为 3。 存储器类型选为简单双端口，因此包含了两套端口，其中 dpra、 dpo 构成了读端口， d、 a、 clk、 we 构成了写端口。

在“RST&Initialization”页面，通过后缀为 coe 的文件对存储器进行初始化，初始化文件内容为：

```
memory_initialization_radix=16;
memory_initialization_vector=7 5 3 6 4 1 2 0 3 1 6;
```

使用实验手册上的仿真文件得到波形如下：

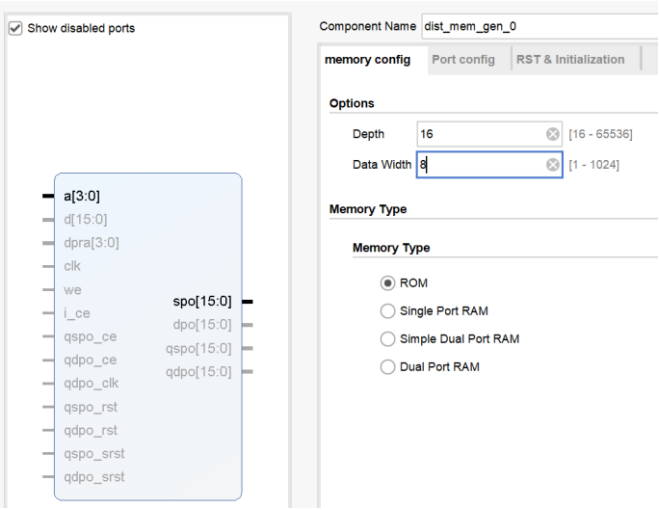


可以看到， 5 号地址的初始值为 1，与 coe 文件相一致，在 115ns 时钟的上升沿处，通过写端口将该地址改写成了 5，因此在 136ns 处读端口从该地读取到的是改写后的数值。

【实验练习】

题目 1

例化一个 16*8bit 的 ROM，并对其进行初始化如下：



设置深度为 16，位宽为 8，coe 文件如下：

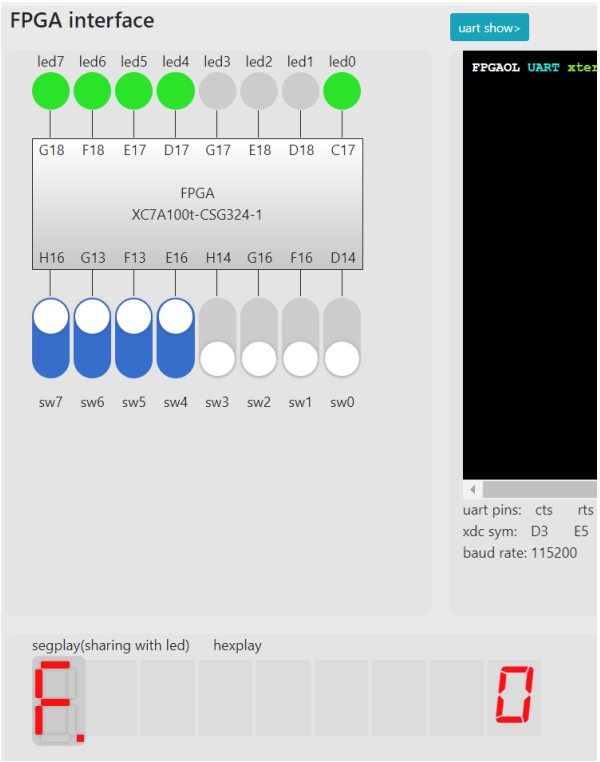
memory_initialization_radix=16;

memory_initialization_vector=bf 86 db cf e6 ed fd 87 ff ef f7 fc b9 de f9 f1;

设计文件如下：

```
23  module T1(
24  input clk,
25  input rst,
26  input [3:0] sw,
27  output reg [7:0]led
28      );
29  wire [7:0]spo;
30  always@(posedge clk or posedge rst)
31  begin
32    if(rst)
33     led <= 8'h00;
34    else
35     led <= {spo[7],spo[6],spo[5],spo[4],spo[3],spo[2],spo[1],spo[0]};
36  end
37
38  dist_mem_gen_1 dist_mem_gen_1(
39  .a  (sw),
40  .spo(spo)
41  );
42  endmodule
```

烧写到 FPGAOL 上效果如下：



用左边四个开关控制数码管输出，测试得到结果正确。上图输入 1111，即 16 进制 F，数码管显示 F。

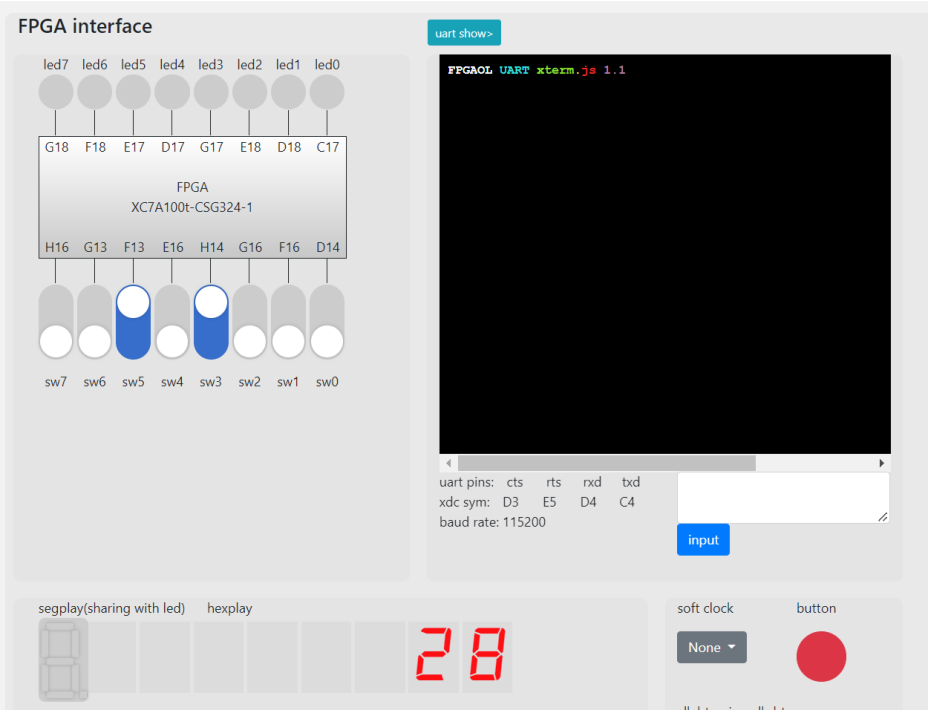
题目 2

首先我们需要对板载时钟进行分频，我们选择分频至 100hz，由于这个频率很低，不能用 IP 核来实现，所以我们用计数器进行时钟分频，实现方式在设计代码图中。

其次，为了实现时分复用，我们引入一个 ctrl 寄存器，他于时钟上升沿到来时在 01 两者中变化，并由此控制 an 的值，当时钟频率适合时，肉眼观察到的就是两个数码管都在亮。实现方式亦在代码图中。

```
23  module T3(
24      input clk,
25      input [7:0] sw,
26      output reg [2:0] an,
27      output reg [3:0] data);
28      reg  ctrl;
29      reg  [20:0] cnt;
30      wire clk_50hz;
31      always@(posedge clk)
32      begin
33          if(cnt >= 99_9999)
34              cnt <= 0;
35          else
36              cnt <= cnt + 1;
37      end
38      assign clk_50hz = (cnt == 1);
39      always@(posedge clk_50hz)
40      begin
41          ctrl <= ~ctrl;
42          if(ctrl == 1)
43          begin
44              an <= 3'b000;
45              data <= sw[3:0];
46          end
47          else
48          begin
49              an <= 3'b001;
50              data <= sw[7:4];
51          end
52      end
53  endmodule
```

烧写到 FPGAOL 上，功能成功实现，如图所示：



题目 3

和题目 2 类似使用计数器进行时钟分频，并使用时分复用方式来实现 4 个数码管的同时显示。

分模块展现该设计代码：

1. 分频模块

在该模块中，引入一个参量 mag，mag 是频率倍数，比如 mag 取到 2，输出的 clkout 周期就是 clk 的两倍。这样我们可以自由调整分频结果。

```
22
23  module fenpin(
24  input clk,
25  input [18:0] mag,
26  output reg clkout
27  );
28  reg [17:0] cnt;
29  initial
30  begin
31    cnt = 0;
32    clkout = 0;
33  end
34  always @ (posedge clk )
35  begin
36    if(cnt == mag/2 - 1)
37    begin
38      clkout <= ~clkout;
39      cnt<=0;
40    end
41    else
42      cnt <= cnt + 1;
43  end
44  endmodule
45
```

2. 时分复用实现模块

和题目 2 中的实现方式相似，只是此处因为需要四个数码管，所以用了 2 位的 sel 来实现。

```
module lab3(
input clk,
input [15:0]sw,
output reg [2:0]sel,
output reg [3:0]out
);
wire [18:0] mag;
wire clkd;
assign mag = 19'b111_1010_0001_0010_0000;//clkd周期为5ms
fenpin fenpin1(
.clk(clk),
.mag(mag),
.clkout(clkd));
reg [1:0] choose;
initial choose = 2'b00;//用来选数码管
always@(posedge clkd)
begin
  case(choose)
  2'b00: choose=2'b01;
  2'b01: choose=2'b10;
  2'b10: choose=2'b11;
  2'b11: choose=2'b00;
  default: choose=2'b00;
  endcase
end
```

3.

顶层文件

调用计时模块和数码管输出模块。

```
21
22
23 module lab3top(
24 input clk,
25 input rst,
26 output [2:0] sel,
27 output [3:0] out
28     );
29 wire [15:0] times;
30 lab3dn lab3dn(
31 .clk(clk),
32 .rst(rst),
33 .out(times));
34 lab3 lab3(
35 .clk(clk),
36 .sw(times),
37 .sel(sel),
38 .out(out));
39 endmodule
40
```

【总结与思考】

本次实验有一定难度，花费了不少时间，但好在是顺利做完了。