

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：\_\_\_宋玮\_\_\_\_\_

学生学号：\_PB20151793\_

完成日期：\_\_\_2021.12.5\_

计算机实验教学中心制

## 【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](http://vlab.ustc.edu.cn)

FPGAOL 平台: [fpgaol.ustc.edu.cn](http://fpgaol.ustc.edu.cn)

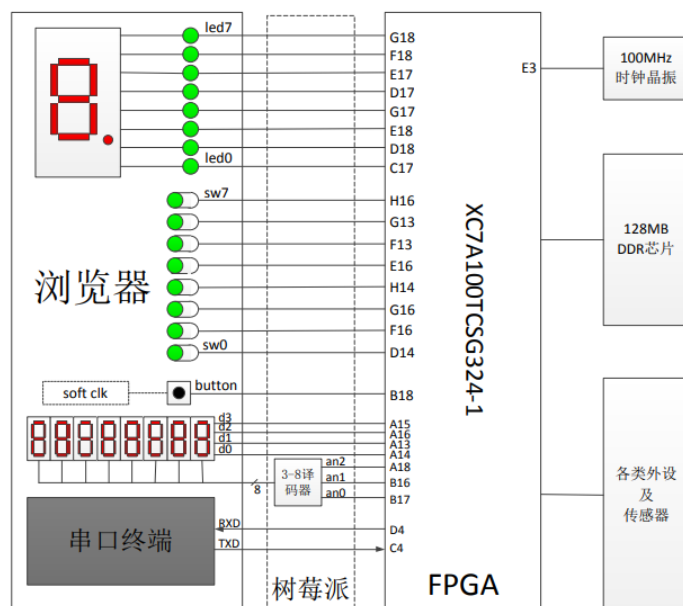
Vivado

Logisim

## 【实验过程】

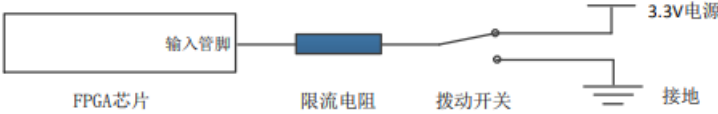
Step1. FPGAOL 实验平台介绍

具体管脚对应关系参考下图:

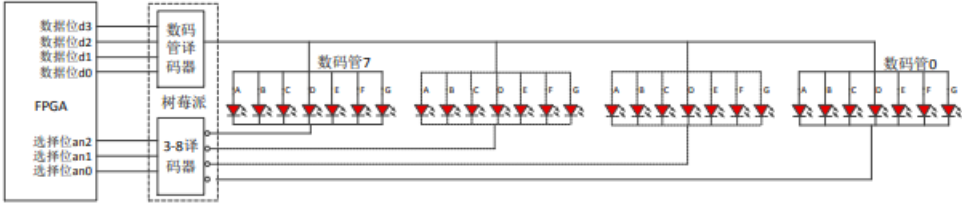


Step2. 外设工作原理介绍

FPGAOL 实验平台中 LED 的工作方式为：FPGA 管脚为高电平时对应 LED 点亮，为低电平时则熄灭。



平台包含一个 8 位的十六进制数码管，该数码管共用 4bit 位宽的数据位，同时由 3bit 的选择位经译码后生成 8 个选择信号，控制 8 个数码管的使能，数据位个选择位信号都是高电平有效。



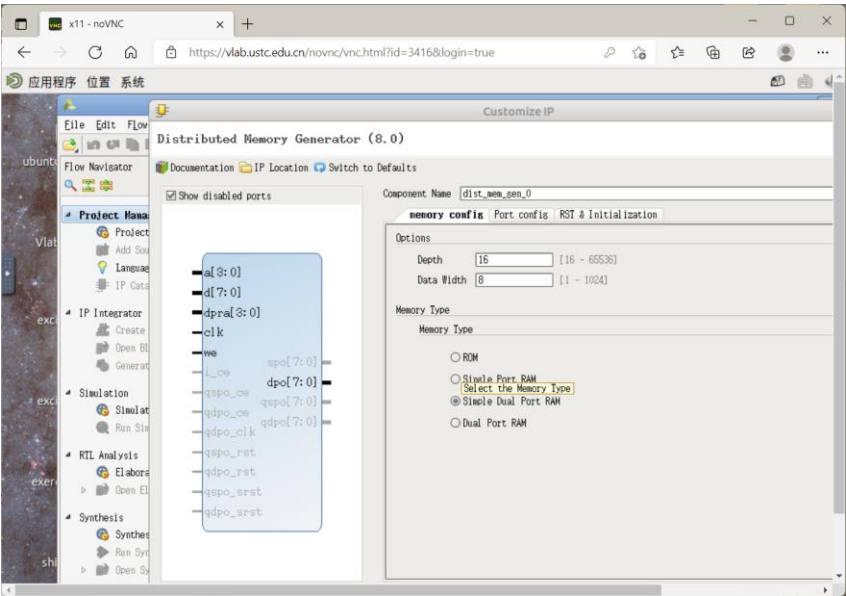
### Step3. 使用时钟管理单元 IP 核

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。此处我们先学习时钟管理单元 IP 核的使用。

FPGA 开发板的 FPGA 芯片 E3 管脚连接了一个 100MHz 频率的时钟晶振，可用作时序逻辑电路的时钟信号。如果我们需要一个其它频率的时钟信号，例如 10MHz，应该怎么办呢？一般的做法是通过计数器产生一个低频的脉冲信号，然后再将该脉冲信号控制其他逻辑的控制信号。

### Step4. 使用片内存储单元

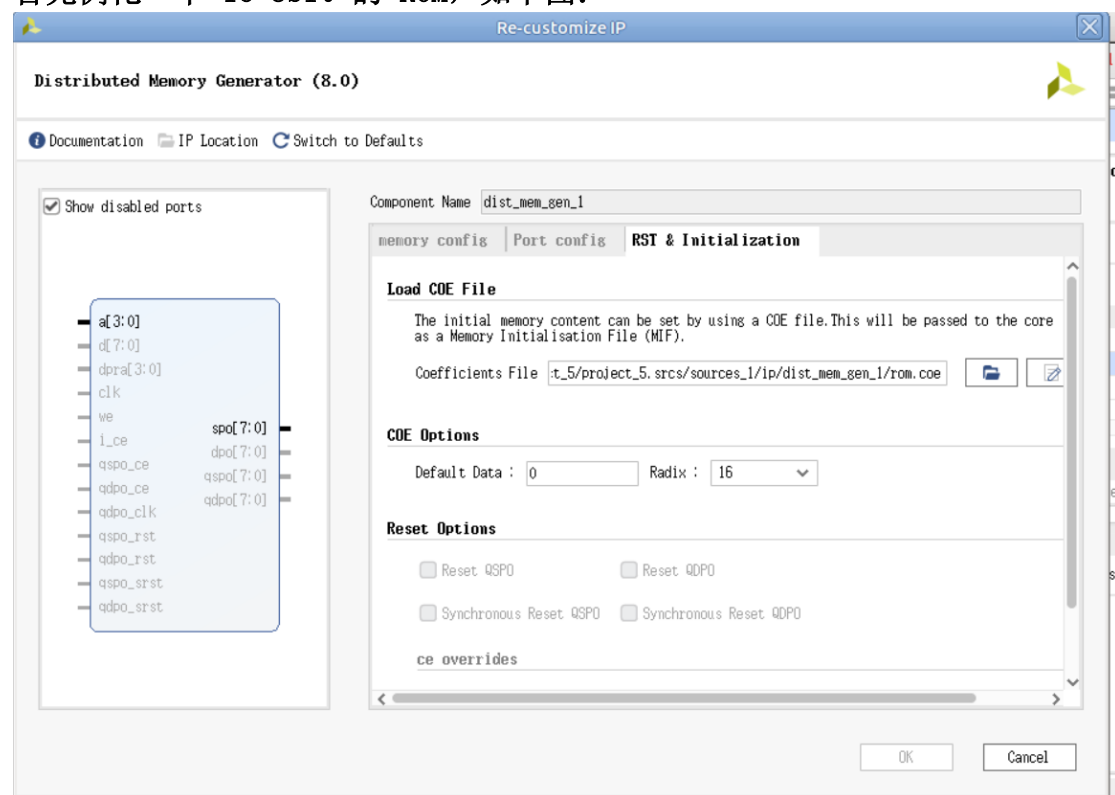
Vivado 中也提供了存储器相关的 IP 核，种类和接口类型比 Logisim 中更加丰富。IP 核目录中提供了两种存储器实现方式：“Distributed Memory”和“Block Memory”两种。



## 【实验练习】

题目 1. 例化一个 16\*8bit 的 ROM，并对其进行初始化，输入端口由 4 个开关控制，输出端口连接到七段数码管上（七段数码管与 LED 复用相同的一组管脚），控制数码管显示与开关相对应的十六进制数字，例如四个开关输入全为零时，数码管显示“0”，输入全为 1 时，数码管显示“F”。

首先例化一个 16\*8bit 的 ROM，如下图：



初始化文件内容为：

```
memory_initialization_radix=16;  
memory_initialization_vector=3F 06 5B 4F 66 6D 7D 07 7F 67 77 7C 39  
5E 79 71;
```

设计文件：

```
module lab7_1(  
    input  [3:0] a,  
    output [7:0] spo  
);
```



题目 2. 采用 8 个开关作为输入，两个十六进制数码管作为输出，采用时分复用的方式将开关的十六进制数值在两个数码管上显示出来，例如高四位全为 1，低四位全为 0 时，数码管显示“F0”。

**分析：**控制八个数码管的 3 位 ( $an0, an1, an2$ ) 每次均为确定的值，若要同时显示出两个数码管的数字，就要将时间分成两半，一半时间显示一个数码管，另一半时间显示另外一个数码管。并且保证人眼难以辨别。所谓：分时复用。

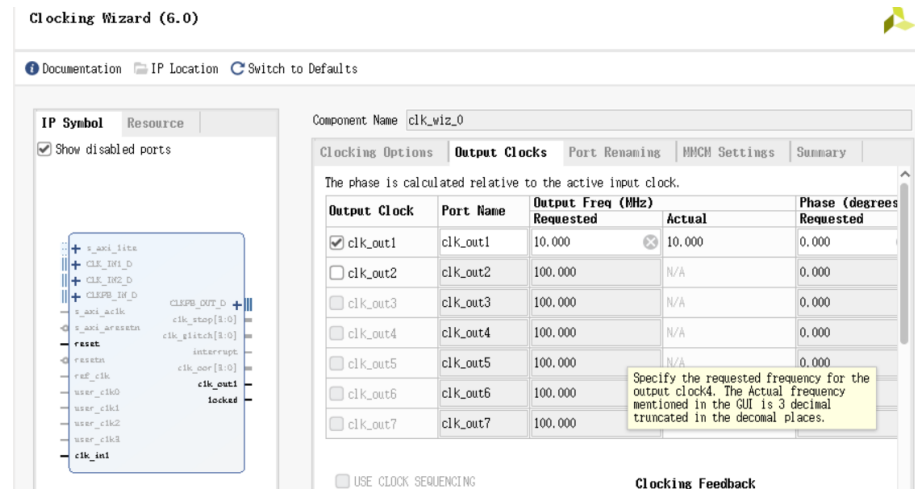
**设计文件：**

```
module lab7_2(
input clk,
input [7:0] sw,
output reg [3:0] d,
output reg [2:0] a=3'b001
);
wire clk1;
wire rs=1'b0;

always@(posedge clk1)
begin
    if(a==3'b001)
    begin
        d<={sw[3], sw[2], sw[1], sw[0]};
        a<=3'b000;
    end
    else begin
        d<={sw[7], sw[6], sw[5], sw[4]};
        a<=3'b001;
    end
end

clk_wiz_0 clk_wiz_0(
    .clk_out1(clk1),
    .reset(rs),
    .locked(),
    .clk_in1(clk)
);
endmodule
```

此处 *clk\_wiz\_0* ip 核输出的频率为 10mhz。



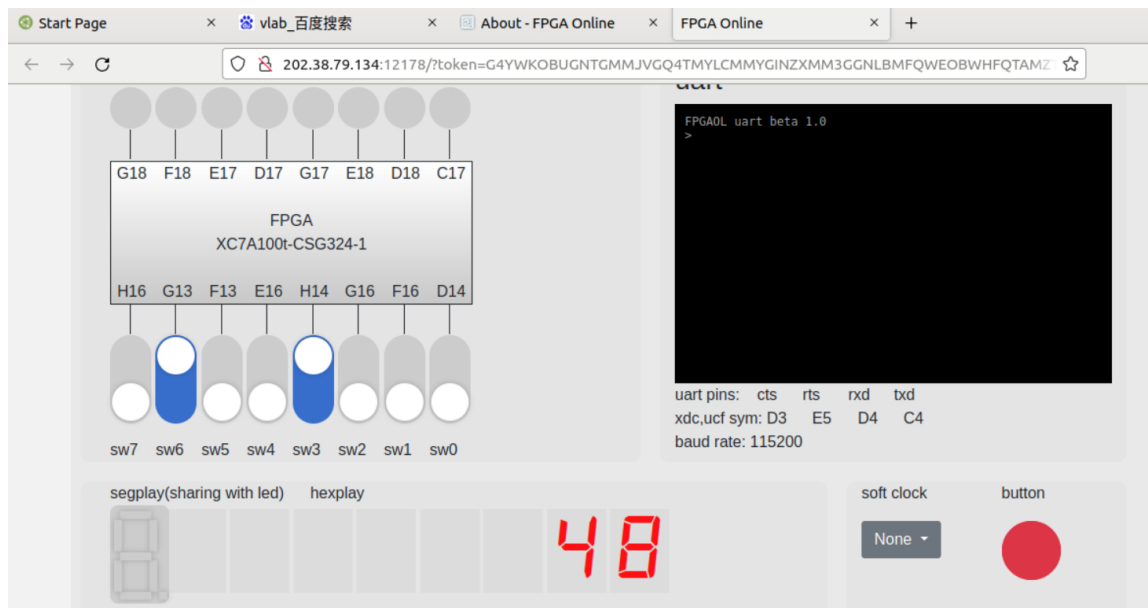
约束文件:

```
set_property -dict { PACKAGE_PIN E3      IOSTANDARD LVCMOS33 } [get_ports {clk}];
```

```
set_property -dict { PACKAGE_PIN A15     IOSTANDARD LVCMOS33 } [get_ports {d[3]}];
set_property -dict { PACKAGE_PIN A16     IOSTANDARD LVCMOS33 } [get_ports {d[2]}];
set_property -dict { PACKAGE_PIN A13     IOSTANDARD LVCMOS33 } [get_ports {d[1]}];
set_property -dict { PACKAGE_PIN A14     IOSTANDARD LVCMOS33 } [get_ports {d[0]}];
set_property -dict { PACKAGE_PIN A18     IOSTANDARD LVCMOS33 } [get_ports {a[2]}];
set_property -dict { PACKAGE_PIN B16     IOSTANDARD LVCMOS33 } [get_ports {a[1]}];
set_property -dict { PACKAGE_PIN B17     IOSTANDARD LVCMOS33 } [get_ports {a[0]}];
```

```
set_property -dict { PACKAGE_PIN H16     IOSTANDARD LVCMOS33 } [get_ports {sw[7]}];
set_property -dict { PACKAGE_PIN G13     IOSTANDARD LVCMOS33 } [get_ports {sw[6]}];
set_property -dict { PACKAGE_PIN F13     IOSTANDARD LVCMOS33 } [get_ports {sw[5]}];
set_property -dict { PACKAGE_PIN E16     IOSTANDARD LVCMOS33 } [get_ports {sw[4]}];
set_property -dict { PACKAGE_PIN H14     IOSTANDARD LVCMOS33 } [get_ports {sw[3]}];
set_property -dict { PACKAGE_PIN G16     IOSTANDARD LVCMOS33 } [get_ports {sw[2]}];
set_property -dict { PACKAGE_PIN F16     IOSTANDARD LVCMOS33 } [get_ports {sw[1]}];
set_property -dict { PACKAGE_PIN D14     IOSTANDARD LVCMOS33 } [get_ports {sw[0]}];
```

实验结果图:



题目 3. 利用本实验中的时钟管理单元或周期脉冲技术，设计一个精度为 0.1 秒的计时器，用 4 位数码管显示出来，数码管从高到低，分别表示分钟、秒钟十位、秒钟个位、十分之一秒，该计时器具有复位功能（可采用按键或开关作为复位信号），复位时计数值为 1234，即 1 分 23.4 秒

设计文件：

```
module lab7_3(
    input clk1,
    input rs,
    output reg [3:0] d, reg [2:0] a
);
    wire rst=1'b0;
    wire clk;
    reg [3:0] m;
    reg [3:0] s10;
    reg [3:0] s1;
    reg [3:0] s_1;
    reg [21:0] t;
```



```

initial
begin
    s_1 <= 4'b0000;
    s1 <= 4'b0000;
    s10 <= 4'b0000;
    m <= 4'b0000;
end

always@(posedge clk)
begin
    if (t >= 1000000)
        t <= 0;
    else
        t <= t + 1;
    end

always@(posedge clk)
begin
    if (a == 3)
        a <= 0;
    else
        a <= a + 1;
    end

always @(posedge clk)
begin
    case(a)
        2:
            begin
                d <= {m[3],m[2],m[1],m[0]};
            end
        1:
            begin
                d <= {s10[3],s10[2],s10[1],s10[0]};
            end
        0:
            begin
                d <= {s1[3],s1[2],s1[1],s1[0]};
            end
        3:
            begin
                d <= {s_1[3],s_1[2],s_1[1],s_1[0]};
            end
    endcase
end

```

```

        end
    endcase
end

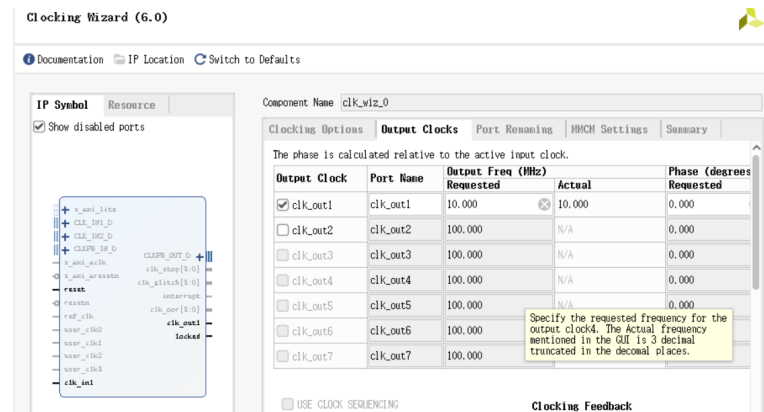
always @(posedge clk)
begin
    if(rs)
    begin
        s_1 <= 4'b0100;
        s1 <= 4'b0011;
        s10 <= 4'b0010;
        m<= 4'b0001;
    end
    else
    begin
        if(t == 0)
        begin
            s_1 <= s_1 + 1;
            if(s_1 == 4'b1001)
            begin
                s_1 <= 4'b0000;
                s1 <= s1 + 1;
                if(s1 == 4'b1001)
                begin
                    s1 <= 4'b0000;
                    s10 <= s10 + 1;
                    if(s10 == 4'b0101)
                    begin
                        s10 <= 4'b0000;
                        m = m + 1;
                    end
                end
            end
        end
    end
end

clk_wiz_0 clk_wiz_0(
.clk_out1(clk),
.reset(rst),
.locked(),
.clk_in1(clk1)
);

```

endmodule

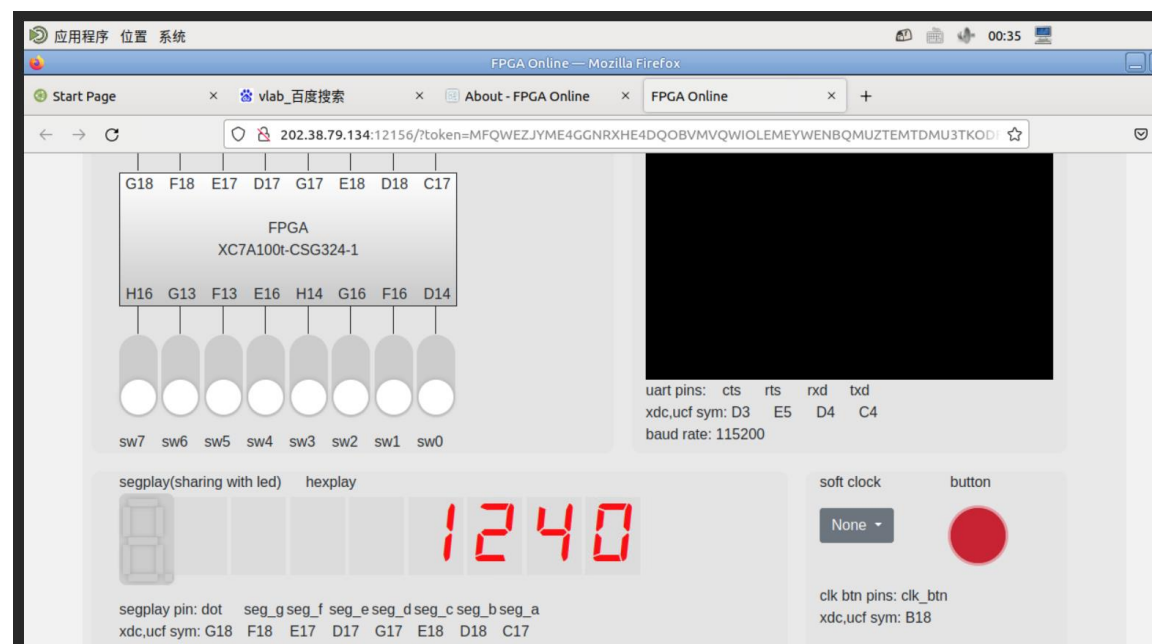
此处 *clk\_wiz\_0* ip 核输出的频率为 10mhz。



约束文件:

```
set_property -dict { PACKAGE_PIN E3      IOSTANDARD LVCOS33 } [get_ports {clk1}];
set_property -dict { PACKAGE_PIN B18     IOSTANDARD LVCOS33 } [get_ports {rs}];
```

```
set_property -dict { PACKAGE_PIN A15     IOSTANDARD LVCOS33 } [get_ports {d[3]}];
set_property -dict { PACKAGE_PIN A16     IOSTANDARD LVCOS33 } [get_ports {d[2]}];
set_property -dict { PACKAGE_PIN A13     IOSTANDARD LVCOS33 } [get_ports {d[1]}];
set_property -dict { PACKAGE_PIN A14     IOSTANDARD LVCOS33 } [get_ports {d[0]}];
set_property -dict { PACKAGE_PIN A18     IOSTANDARD LVCOS33 } [get_ports {a[2]}];
set_property -dict { PACKAGE_PIN B16     IOSTANDARD LVCOS33 } [get_ports {a[1]}];
set_property -dict { PACKAGE_PIN B17     IOSTANDARD LVCOS33 } [get_ports {a[0]}];
```



### 【总结与思考】

本次实验体验了 ip 核的使用，包括 rom 和时钟核。由于本次实验任务中要同时显示多个代码管，因此要降低时钟频率。遇到了挺多问题，而且本次实验任务量比较大，花费了不好时间。实验写的 verilog 代码也明显增多了，逻辑也更复杂。不过最后也顺利完成了，无特殊建议。