

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：林宸昊
学生学号：PB20000034
完成日期：11.25

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在线实验平台的结构和使用
- 掌握FPGA开发各关键环节
- 学会使用IP核

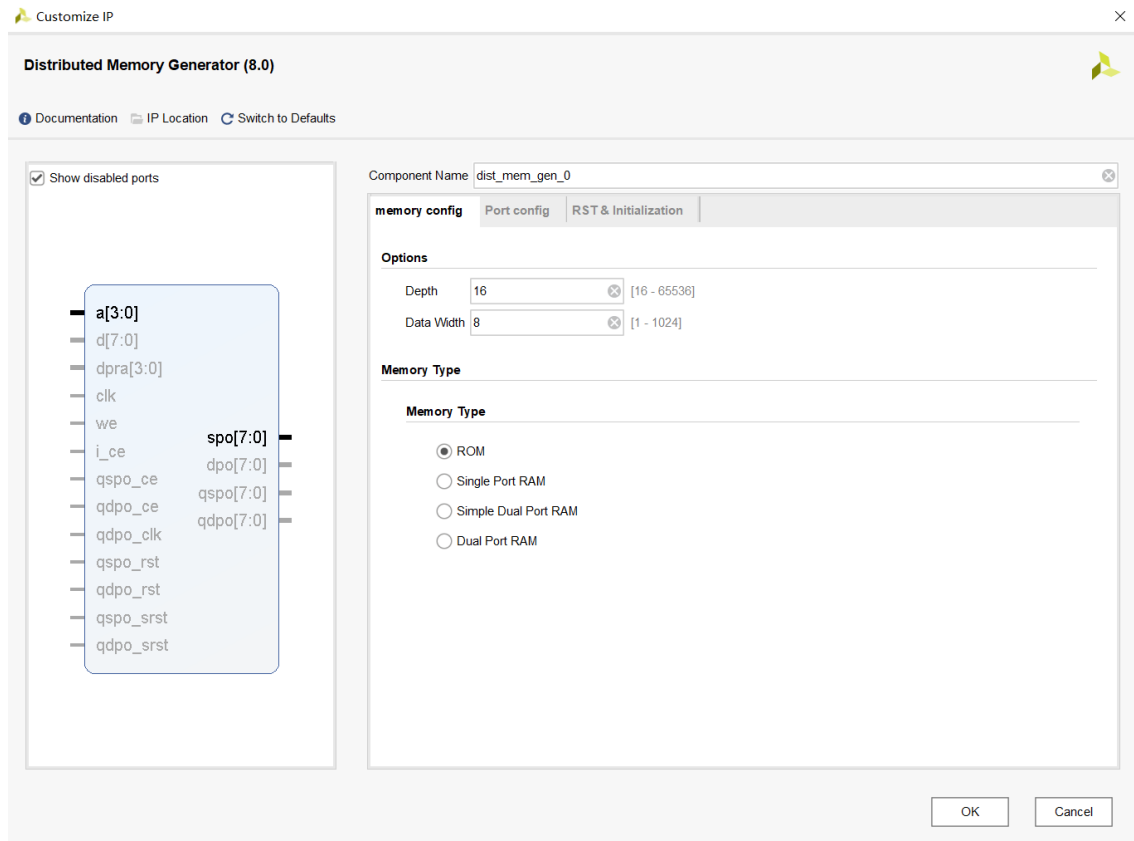
【实验环境】

- FPGAOL平台: fpgaol.ustc.edu.cn
- Vivado
- Logisim

【实验练习】

题目1

- 例化ROM，并将其初始化
 - 设置数据位宽及深度



- 设置初始化值（根据数码管的不同显示计算相应16进制表示）

COE File Editor - default.coe

×

Key	Value
memory_initialization_radix	16
memory_initialization_vector	3F 06 5B 4F 66 6D 7D 07 7F 6F F7 FF B9

Σ

Σ

Σ

Validate

Save

Save As...

Close

```
memory_initialization_radix=16;
memory_initialization_vector=3F 06 5B 4F 66 6D 7D 07 7F 6F F7 FF B9 BF F9 F1;
```

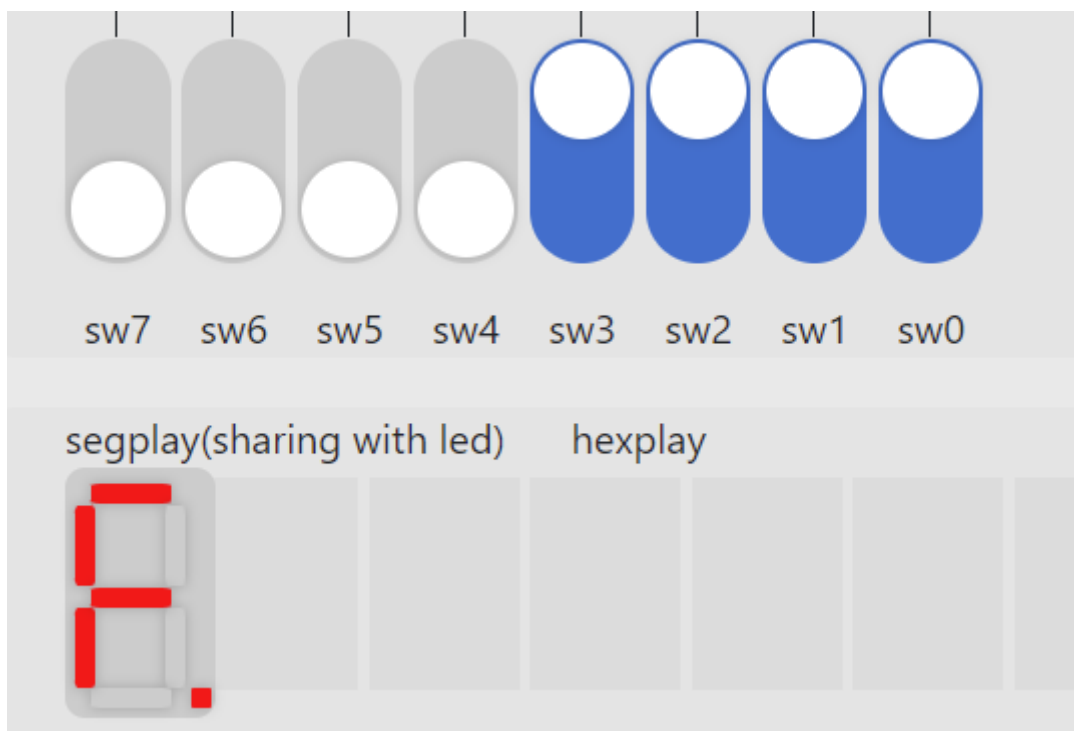
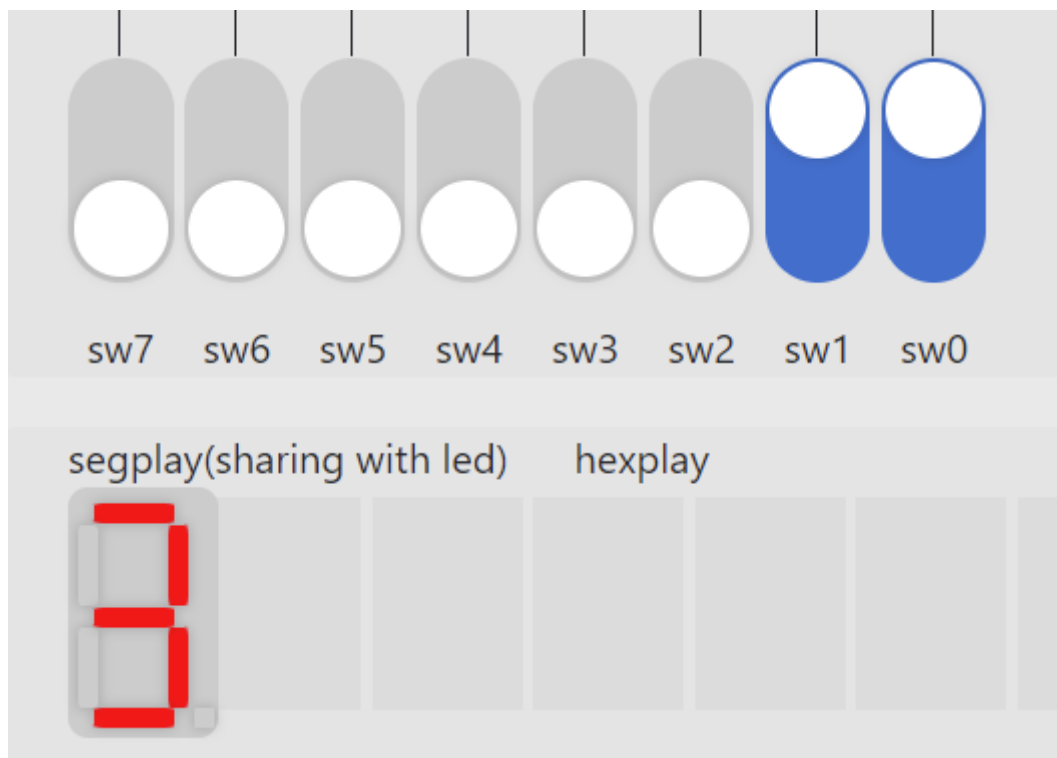
引用例化

```
module EXP1(
    input [3:0] sw,
    output [7:0] led);
    dist_mem_gen_0 rom1(.a(sw),.spo(led));
endmodule
```

设置管脚

```
6 : ## Clock signal
7 : #set_property -dict { PACKAGE_PIN E3      IOSTANDARD LVCMOS33 } [get_ports { CLK100MHZ }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
8 : #create_clock -add -name sys_clk_pin -period 10.00 -waveform { 0 5} [get_ports { CLK100MHZ }];
9 :
10 :
11 : ## FPGA0L LED (single-digit-SEGPLAY)
12 :
13 : set_property -dict { PACKAGE_PIN C17      IOSTANDARD LVCMOS33 } [get_ports { led[0] }];
14 : set_property -dict { PACKAGE_PIN D18      IOSTANDARD LVCMOS33 } [get_ports { led[1] }];
15 : set_property -dict { PACKAGE_PIN E18      IOSTANDARD LVCMOS33 } [get_ports { led[2] }];
16 : set_property -dict { PACKAGE_PIN G17      IOSTANDARD LVCMOS33 } [get_ports { led[3] }];
17 : set_property -dict { PACKAGE_PIN D17      IOSTANDARD LVCMOS33 } [get_ports { led[4] }];
18 : set_property -dict { PACKAGE_PIN E17      IOSTANDARD LVCMOS33 } [get_ports { led[5] }];
19 : set_property -dict { PACKAGE_PIN F18      IOSTANDARD LVCMOS33 } [get_ports { led[6] }];
20 : set_property -dict { PACKAGE_PIN G18      IOSTANDARD LVCMOS33 } [get_ports { led[7] }];
21 :
22 :
23 : ## FPGA0L SWITCH
24 :
25 : set_property -dict { PACKAGE_PIN D14      IOSTANDARD LVCMOS33 } [get_ports { sw[0] }];
26 : set_property -dict { PACKAGE_PIN F16      IOSTANDARD LVCMOS33 } [get_ports { sw[1] }];
27 : set_property -dict { PACKAGE_PIN G16      IOSTANDARD LVCMOS33 } [get_ports { sw[2] }];
28 : set_property -dict { PACKAGE_PIN H14      IOSTANDARD LVCMOS33 } [get_ports { sw[3] }];
29 : set_property -dict { PACKAGE_PIN E16      IOSTANDARD LVCMOS33 } [get_ports { sw[4] }];
30 : set_property -dict { PACKAGE_PIN F13      IOSTANDARD LVCMOS33 } [get_ports { sw[5] }];
31 : set_property -dict { PACKAGE_PIN G13      IOSTANDARD LVCMOS33 } [get_ports { sw[6] }];
32 : set_property -dict { PACKAGE_PIN H16      IOSTANDARD LVCMOS33 } [get_ports { sw[7] }];
```

烧写到平台



题目2

- 设计文件
 - 由于同一时刻平台只能给一个数码管赋值，我们需要借用平台的高时钟频率经过适当降频使在视觉上同时出现两个数码管。

```
module EXP2(
    input clk,
    input [7:0] sw,
    output reg [2:0] ans, //作为数码管的选择输入
    output reg [3:0] d    //作为数码管的数据输入
);
    reg [9:0] count;      //作为降频用的计数器
    wire slowed_pulse;    //降频后的信号
```

```

always @ (posedge clk)
    count <= count + 1;
assign slowed_pulse = (count >= 9'h1FF); //设定降频信号
always @ (posedge clk)
    begin
        if(slowed_pulse)
            begin
                ans <= 3'b000;
                d <= sw[3:0]; //低四位的输出
            end
        else
            begin
                ans <= 3'b001;
                d <= sw[7:4]; //高四位的输出
            end
        end
    end
endmodule

```

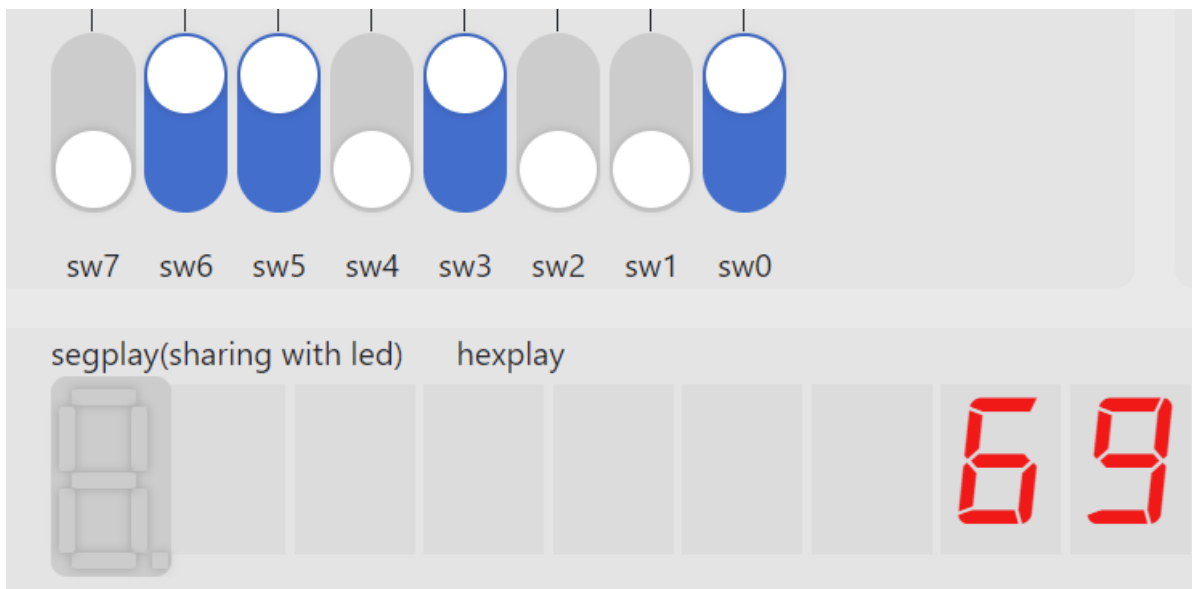
- 约束文件(时钟的管脚截图略去)

```

35  ## FPGA0L HEXPLAY
36
37  set_property -dict { PACKAGE_PIN A14  IOSTANDARD LVCMOS33 } [get_ports { d[0] }];
38  set_property -dict { PACKAGE_PIN A13  IOSTANDARD LVCMOS33 } [get_ports { d[1] }];
39  set_property -dict { PACKAGE_PIN A16  IOSTANDARD LVCMOS33 } [get_ports { d[2] }];
40  set_property -dict { PACKAGE_PIN A15  IOSTANDARD LVCMOS33 } [get_ports { d[3] }];
41  set_property -dict { PACKAGE_PIN B17  IOSTANDARD LVCMOS33 } [get_ports { ans[0] }];
42  set_property -dict { PACKAGE_PIN B16  IOSTANDARD LVCMOS33 } [get_ports { ans[1] }];
43  set_property -dict { PACKAGE_PIN A18  IOSTANDARD LVCMOS33 } [get_ports { ans[2] }];

```

- 烧写到平台



题目3

- 设计文件
 - 我们知道fpga平台所提供的时钟频率是固定100MHz，则若需以0.1s为周期，即10Hz，则需要延时24'd1000_0000个原有时钟周期。

```

module EXP3(
    input clk,

```

```

input rst,
output reg [3:0] d,
output reg [2:0] ans
);
parameter MAX = 24'd1000_0000; //作为一个周期结束时计数器的数值
reg [23:0] count;                //用于计数，24位足够
reg flag;                        //用于判断是否到达一次0.1s
reg [3:0] cnt1;                  //储存毫秒位
reg [3:0] cnt2;                  //储存秒个位
reg [3:0] cnt3;                  //储存秒十位
reg [3:0] cnt4;                  //储存分钟位
reg [9:0] times;                 //用于降频输出
always @ (posedge clk or posedge rst)
begin
    if(rst)
    begin
        count <= 0;
        flag <= 0;
    end
    else if (count == MAX)
    begin
        count <= 0;
        flag <= 1;           //到达一个周期，flag置1，否则清0
    end
    else
    begin
        count <= count + 1'b1;
        flag <= 0;
    end
end

always @ (posedge flag or posedge rst)
begin
    if(rst)
    begin
        cnt1 <= 4;
        cnt2 <= 3;
        cnt3 <= 2;
        cnt4 <= 1;
    end
    else
    begin
        if(cnt1 < 9) //逢10进1
            cnt1 <= cnt1 + 1'b1;
        else
        begin
            if(cnt2 < 9) //逢10进1
                cnt2 <= cnt2 + 1'b1;
            else
            begin
                if(cnt3 < 5) //逢6进1
                    cnt3 <= cnt3 + 1'b1;
                else
                begin
                    if(cnt4 < 9) //逢10进1
                        cnt4 <= cnt4 + 1'b1;
                    else
                        cnt4 <= 0;
                end
            end
        end
    end
end

```

```

        cnt3 <= 0;
    end
    cnt2 <= 0;
end
cnt1 <= 0;
end
end
end
//由于需要对四个数码管进行输出，故单一的SLOWER_PULSE不好控制，我们可以直接将降频计数器
//分段并对应输出
always @(posedge clk)
begin
    times <= times +1;
    if(times == 10'd256)
    begin
        d <= cnt1;
        ans <= 0;
    end
    else if(times == 10'd512)
    begin
        d <= cnt2;
        ans <= 1;
    end
    else if(times == 10'd768)
    begin
        d <= cnt3;
        ans <= 2;
    end
    else if(times == 10'd1024)
    begin
        d <= cnt4;
        ans <= 3;
    end
end
end
endmodule

```

- 约束文件同上
- 烧写到平台



【总结与思考】

- 实验收获
 - 学会使用常用的IP核，如clocking wizard，RAM，ROM；
 - 了解FPGA平台上数码管的对应管脚如何控制数码管的输出；
 - 学会通过降频使得同时出现多个数码管图像。
- 实验难易及任务量

- 较难。主要体现在对实验过程的理解，要求能够对新的工具（IP核）熟练操作并使用，以及能熟练的进行降频操作来获得目标输出。这就要求细读实验过程并最好能将所有代码按照要求进行仿真以求加深理解，这样可以极大程度的方便后续实验。同时，多重进位的计数器也是一个难点。
- 任务量较大，体现在时间方面就是代码的规范以及烧写过程所耗费的大量时间。但其实只要能提前理解时分复用的含义，总体来说并不难。
- 实验建议
 - 也许可以出一些计算题，计算通过某种降频后得到的频率是多少；或者也可以要求用两种及以上的降频方式来达到预期效果，毕竟从老师代码及我个人所写来看，降频方式有多种且效果也有所不同。