

1. LED 流水灯实验例程

1.1 MES22GP 开发板简介

MES22GP 开发板共有 11 个翠绿 LED 灯，其中 1 个是电源指示灯（POWER）； 2 个是 FPGA 的运行状态指示灯：INIT 和 DONE；有 8 个用户 LED 灯（LED1~8），FPGA 输出高电平时对应的 LED 灯亮灯（详情请查看“MES22GP 开发板硬件使用手册”）。

1.2 实验目的

控制 8 个 LED 灯按顺序依次点亮和熄灭。

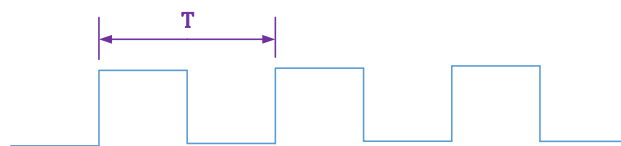
1.3 实验原理

通常的时，分，秒的计时进位大家应该不陌生；

1 小时=60 分钟=3600 秒，当时针转动 1 小时，秒针跳动 3600 次；



在数字电路中的时钟信号也是有固定的节奏的，这种节奏的开始到结束的时间，我们通常称之为周期（T）。



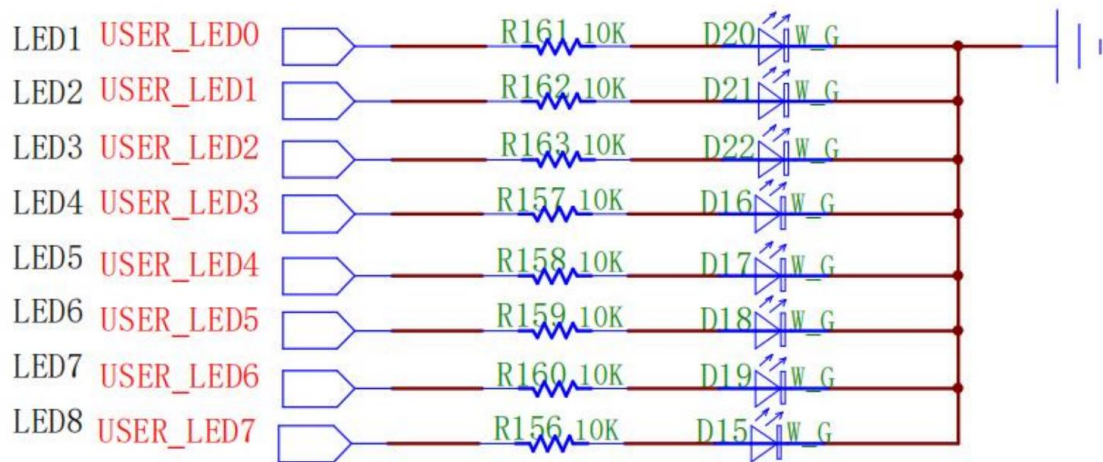
在数字系统中通常关注到时钟的频率，那频率与周期的关系如下：

$$f = \frac{1}{T};$$

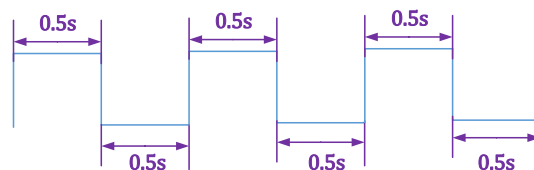
MES22GP 板卡上单端时钟有一个 50MHz 和一个 27MHz 的晶振提供时钟给到 PGL22G；

实验分析：

控制 LED 亮灭需要控制 IO 输出的高低电平即可（高电平点亮，低电平熄灭），原理图如下：



控制 LED 依次 0.5s 亮, 0.5s 灭, 需要控制 IO 依次输出 0.5s 高电平, 0.5s 低电平周期变化, 如下图波形:



若使用 50MHz 外部输入时钟, 时钟周期为 20ns (在 verilog 设计中的计数器的计时原理基本上是一致的, 确认输入时钟周期和目标计时时间后可得到计数器的计数值到达多少后可得到计时宽度);

$$0.5s = 25000000 \times 20ns = 25000000 \times T_{50MHz};$$

IO 输出状态只有两种: 1 或 0; 我们可以使用一个计数器, 计数满 25000000 个时钟周期时变化不同 LED 点亮。

1.4 实验源码设计

1.4.1 文件头设计

在 module 之前添加文件头, 文件头中包含信息有: 公司, 作者, 时间, 设计名, 工程名, 模块名, 目标器件, EDA 工具 (版本), 模块描述, 版本描述 (修改描述) 等信息; 以及仿真时间单位定义;

```

1  `timescale 1ns / 1ps
2  //////////////////////////////////////
3  //Company       : MEYESEMI Technology CO., Ltd
4  //Internet site  : www.meyesemi.com
5  // WeChat public account : MEYESEMI
6  //Engineer      : Nill
7  //
8  //Create Date   : 2022-12-27 14:34
9  //Design Name   : led_test
10 //Module Name   : led_test
11 //Project Name  : led_light
12 //Target Devices : Pango
13 //Tool Versions :
14 //Description   :
15 //
16 //Dependencies  :
17 //
18 //Revision      : Revision 0.01 - File Created
19 //Additional Comments :
20 //
21 //////////////////////////////////////
22 `define UD #1

```

`timescale 1ns / 1ps 表示仿真精度是 1ns，显示精度是 1ps；

`define UD #1 定义 UD 表示#1；#1 仅仿真有效，表示延时一个仿真精度，结合上一条语句表示延时 1ns；

1.4.2 设计 module

1.4.2.1 创建 module，确定输入输出信号

```

module led_light(
    input      clk,      // input clock, the frequency is 50MHz
    input      rstn,     // input reset signal, active at low
    output [7:0] led      // output LED control signal , lighting at high
);
endmodule

```

此段代码是标准的 module 创建的模型，module 创建时需要确认输入输出信号并定义好位宽，之后在对 module 进行具体的逻辑设计；管脚与管脚之间用 “,” 隔开，最后一个管脚不用间隔符号；

创建 module 时需要定义输入输出信号；本实验输入时钟和复位即可，输出是控制 LED 的亮灭，MES22GP 板卡上共有 8 个 LED，故而输出 8bit 位宽的信号；

1.4.2.2 设计一个计数器；

单个状态计数 25_000_000，即 $24_999_999 = 25' b1_0111_1101_0111_1000_0011_1111$ ；

所以计数器的位宽为 25 位即可, 此处请结合数字电路中的同步计数器的工作原理分析;

```
1    reg [24:0] led_light_cnt;
2
3    // time counter
4    always @(posedge clk) // 触发条件: posedge 为上升沿, negedge 为下降沿
5    begin
6        if(!rstn)
7            led_light_cnt <= `UD 25'd0;
8        else if(led_light_cnt == 25'd24_999_999)
9            led_light_cnt <= `UD 25'd0;
10       else
11           led_light_cnt <= `UD led_light_cnt + 25'd1;
12    end
```

当计数器计数到 25'd24_999_999 时, 计数过程包含了从 0~26'd2499_9999 的时钟周期, 故而总时长为 $25'd25_000_000 \times T_{clk}$; 硬件输入时钟为 50MHz, 所以此计数器的计数周期是 0.5s;

1.4.2.3 led 显示状态控制

在指定的时间刻度上对 LED 的状态进行变更, 以达到控制 LED 依次亮灭的目的;

led_light_cnt 的计时周期为 0.5s, 故在 led_light_cnt 上取一个点来变更 LED 的显示状态即可完成每隔 0.5s LED 显示发生变化; 由于 LED 亮和灭只有两个状态, 在赋值处理上将寄存器进行移位操作;

```
1    reg [7:0] led_status;
2
3    always @(posedge clk)
4    begin
5        if(!rstn)
6            led_status <= `UD 8'b0000_0001;
7        else if(led_light_cnt == 25'd2499_9999)
8            led_status <= `UD { led_status[6:0], led_status[7] };
9    end
10
11    assign led = led_status;
```

1.4.3完整的 Module（不含注释）

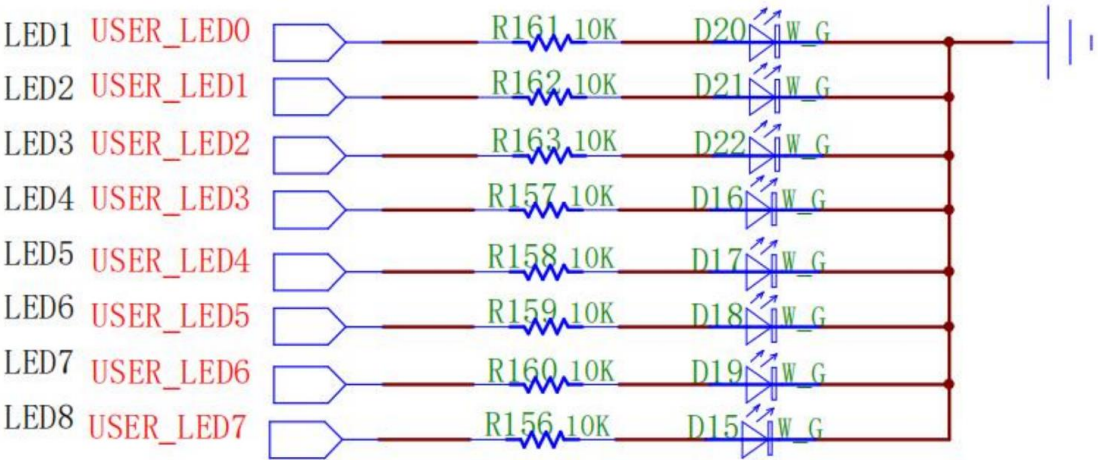
```

1  `timescale 1ns / 1ps
2
3  `define UD #1
4  module led_light(
5      input          clk,
6      input          rstn,
7      output [7:0]   led
8  );
9
10 //=====
11 //reg and wire
12 reg [24:0] led_light_cnt;
13 reg [ 7:0] led_status;
14 // time counter
15 always @(posedge clk)
16 begin
17     if(!rstn)
18         led_light_cnt <= `UD 25'd0;
19     else if(led_light_cnt == 25'd2499_9999)
20         led_light_cnt <= `UD 25'd0;
21     else
22         led_light_cnt <= `UD led_light_cnt + 25'd1;
23 end
24
25 // led status change
26 always @(posedge clk)
27 begin
28     if(!rstn)
29         led_status <= `UD 8'b0000_0001;
30     else if(led_light_cnt == 25'd2499_9999)
31         led_status <= `UD { led_status[6:0], led_status[7] };
32 end
33 assign led = led_status;
34 endmodule

```

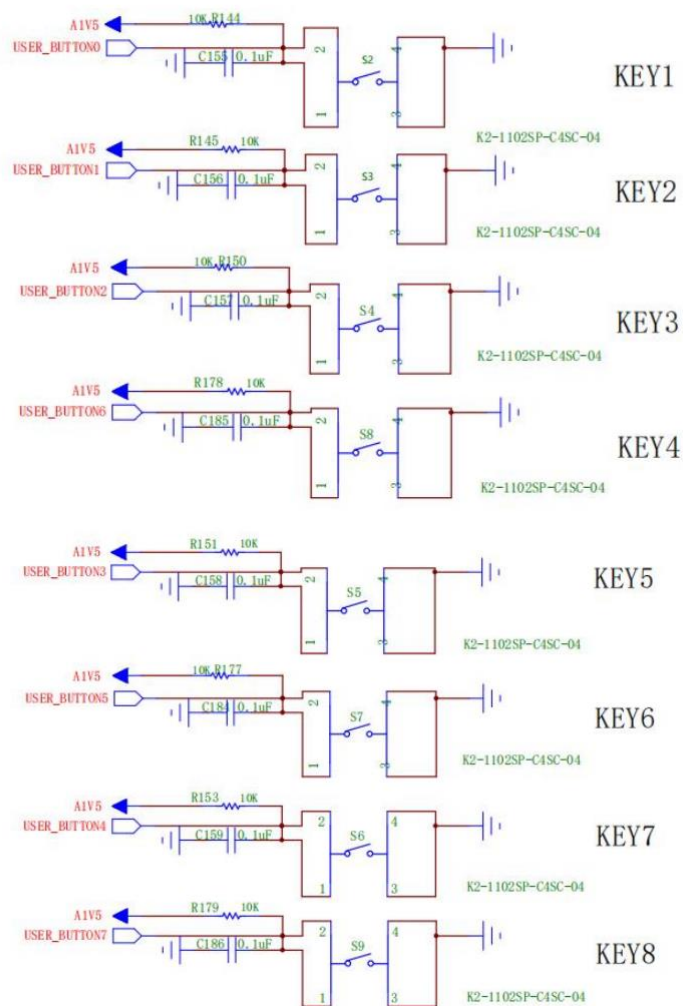
1.4.4硬件管脚分配

MES22GP 的 LED 和 CLK 与 FPGA 的 IO 连接部分的原理图如下，详情可查看硬件使用手册或原理图：



信号	PGL22G Pin
LED1	E5
LED2	E6
LED3	F7
LED4	F8
LED5	E8
LED6	D8
LED7	C8
LED8	A8

复位设计是低电平有效，MES22GP 开发板提供了 8 个用户按键（K1~K8），按键低电平有效，但按键按下时，IO 上的输入电压为低；当没有按下按键时，IO 上的输入电压为高电平；选择任一个用户按键作为复位输入即可。



1.5 实验步骤

1.5.1 PSD 软件相关

(1) PSD 软件安装包获取:

软件版本: PDS_2022.1

(注: 本文档中所有实验均使用 PDS_2022.1 版本实现)

软件安装包位置: 5_Software\PDS2022.1;

(2) PSD 软件 license 申请:

方法一: 通过紫光同创官网申请 license, 官网链接: <https://www.pangomicro.com/>;

方法二: 虚拟网卡设置 MAC, 参考文档: 7_doc\虚拟网卡设置 MAC;

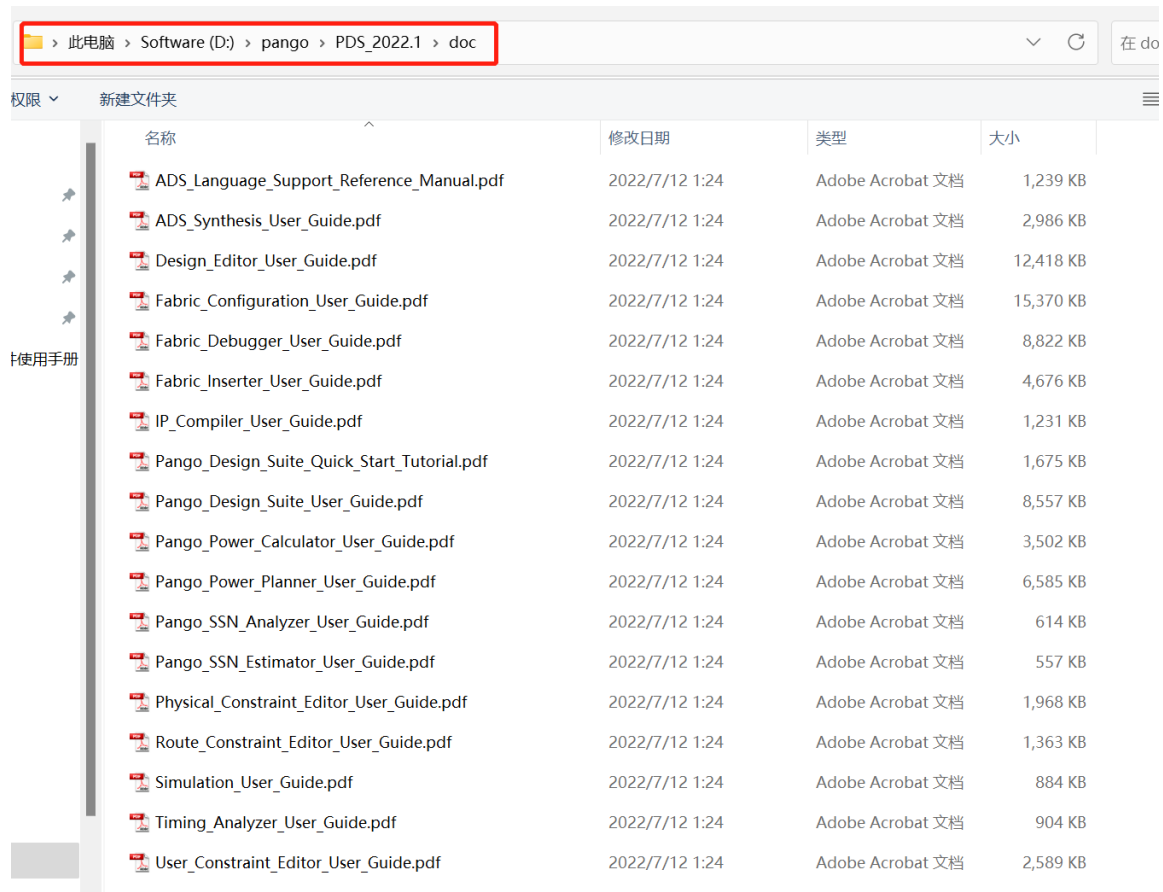
(3) PSD 软件安装流程:

PSD 软件安装步骤参考文档: 1_Demo_document\工具使用篇;

(4) PDS 软件使用:

a. 快速使用手册可查看文档: 1_Demo_document\工具使用篇\ 02_PDS 快速使用手册

b. PDS 各工具使用可查看: 安装路径下 doc 文件夹, 示意图如下:



1.6 下载与 Flash 固化

下载与 flash 固化可参考文档: 1_Demo_document\工具使用篇\04_FPGA&CPLD 的下载与固化

1.7 实验现象

8 颗 LED 灯按照设定的顺序和时间依次点亮和熄灭。